IMS
Version 15.1.0

Commands, Volume 3: IMS Component and z/OS Commands

(January 30, 2018 edition)

SC27-6782-00

IMS
Version 15.1.0

Commands, Volume 3: IMS Component and z/OS Commands

(January 30, 2018 edition)

SC27-6782-00

Note
Before you use this information and the product it supports, read the information in "Notices" on page 661.

January 30, 2018 edition.

This edition applies to IMS 15 (program number 5635-A06), IMS Database Value Unit Edition, V15.01.00 (program number 5655-DS5), IMS Transaction Manager Value Unit Edition, V15.01.00 (program number 5655-TM4), and to all subsequent releases and modifications until otherwise indicated in new editions.

© Copyright IBM Corporation 1974, 2017.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

About this information vii
Prerequisite knowledge vii
How new and changed information is identified . . vii
How to read syntax diagrams viii
Accessibility features for IMS 15 ix
How to send your comments x

Part 1. Base Primitive Environment commands 1

Chapter 1. Introduction to the BPE commands 3

Chapter 2. BPE DISPLAY VERSION command 7

Chapter 3. BPE TRACETABLE commands 9
BPE DISPLAY TRACETABLE command 9
BPE UPDATE TRACETABLE command 15

Chapter 4. BPE USEREXIT commands 23
BPE DISPLAY USEREXIT command 23
BPE REFRESH USEREXIT command 30

Part 2. Common Service Layer commands 37

Chapter 5. CSL SHUTDOWN command 39

Chapter 6. STOP csljob commands . . 41

Part 3. Common Queue Server command 43

Chapter 7. STOP cqsjob command. . . 45

Part 4. DBRC commands 47

Chapter 8. Introduction to the DBRC commands 49
DBRC batch commands 49
DBRC online commands 50
DBRC command syntax 50
DBRC time stamps 52

Chapter 9. BACKUP.RECON command 61

Chapter 10. CHANGE commands . . . 63
CHANGE.ADS command 63
CHANGE.BKOUT command 64
CHANGE.CA command 66
CHANGE.CAGRP command 69
CHANGE.DB command 72
CHANGE.DBDS command 84
CHANGE.DBDSGRP command 93
CHANGE.IC command 96
CHANGE.PART command 99
CHANGE.PRILOG OLDS command. 107
CHANGE.PRILOG RLDS command 109
CHANGE.PRILOG SLDS command 113
CHANGE.RECON command 118
CHANGE.SECLOG OLDS command 132
CHANGE.SECLOG RLDS command 134
CHANGE.SECLOG SLDS command. 138
CHANGE.SUBSYS command 143
CHANGE.UIC command 145

Chapter 11. CLEANUP.RECON command 147

Chapter 12. DELETE commands . . . 161
DELETE.ADS command. 161
DELETE.ALLOC command. 162
DELETE.BKOUT command. 163
DELETE.CA command 163
DELETE.CAGRP command. 164
DELETE.DB command 165
DELETE.DBDS command 166
DELETE.DBDSGRP command. 167
DELETE.IC command 167
DELETE.LOG command (for OLDS). 168
DELETE.LOG command (for RLDS and SLDS) . . 169
DELETE.PART command 172
DELETE.RECOV command. 173
DELETE.REORG command. 174
DELETE.SUBSYS command 175
DELETE.UIC command 176

Chapter 13. GENJCL commands . . . 177
GENJCL.ARCHIVE command 177
GENJCL.CA command 181
GENJCL.CLOSE command 186
GENJCL.IC command 189
GENJCL.OIC command 206
GENJCL.RECOV command. 211
GENJCL.USER command 218
Skeletal JCL for DBRC 222
Generating JCL and user-defined output . . . 223
Using IBM-supplied skeletal JCL 224

© Copyright IBM Corp. 1974, 2017

Skeletal JCL data set members 224
Skeletal JCL syntax 225
Symbolic keywords recognized by DBRC . . . 249
IBM-supplied skeletal JCL execution members 258

Chapter 14. INIT commands 283
INIT.ADS command 283
INIT.CA command 285
INIT.CAGRP command 287
INIT.DB command 289
INIT.DBDS command 294
INIT.DBDSGRP command 301
INIT.IC command 303
INIT.PART command 305
INIT.RECON command 311

Chapter 15. LIST commands 321
LIST.BKOUT command 321
LIST.CAGRP command 322
LIST.DB command 324
LIST.DBDS command. 326
LIST.DBDSGRP command 329
LIST.HISTORY command 331
LIST.LOG command (for a PRILOG family) . . . 334
LIST.LOG command (for a category of records) . . 335
LIST.RECON command 338
LIST.SUBSYS command 342

Chapter 16. NOTIFY commands . . . 345
NOTIFY.ALLOC command 345
NOTIFY.BKOUT command 347
NOTIFY.CA command 348
NOTIFY.IC command 352
NOTIFY.PRILOG command (for OLDS) 356
NOTIFY.PRILOG command (for RLDS) 359
NOTIFY.PRILOG command (for SLDS) 363
NOTIFY.RECOV command 366
NOTIFY.REORG command 370
NOTIFY.SECLOG command (for OLDS) 374
NOTIFY.SECLOG command (for RLDS) 377
NOTIFY.SECLOG command (for SLDS) 381
NOTIFY.SUBSYS command. 384
NOTIFY.UIC command 386

Chapter 17. REPAIR.RECON command 389

Chapter 18. Sample listings from the RECON data set 393
Sample listing of LIST.DB command output . . . 393
Sample listing of LIST.HISTORY command output 393
Sample listing of a RECON data set 406
Fields in a RECON listing, by record type 420
RECON record fields 420
Log record fields 425
LOGALL record fields 425
Online log record fields 425
SSYS record fields 425
BACKOUT record fields 425
CAGRP record fields 425
CA record fields 425
Data group record fields. 425
DB (IMS) record fields 426
DB (HALDB) record fields 426
DB (PART) record fields 426
DB (Fast Path) record fields 426
DBDS (non-Fast Path) record fields 426
DBDS (Fast Path) record fields 426
ALLOC record fields 426
IMAGE record fields 426
REORG record fields 426
RECOV record fields 426

Part 5. Fast Database Recovery commands 427

Chapter 19. MODIFY fdbrproc commands 429
F fdbrproc,DUMP 429
F fdbrproc,RECOVER 429
F fdbrproc,STATUS 429
F fdbrproc,STOP 431
F fdbrproc,TERM 431

Chapter 20. START fdbrproc command 433

Part 6. IMS Connect commands 435

Chapter 21. Equivalent IMS Connect WTOR, z/OS, and type-2 commands . 437

Chapter 22. IMS Connect type-2 commands 443

Chapter 23. IMS Connect WTOR commands 445
CLOSEHWS command 445
OPENDS command 447
OPENIP command 448
OPENPORT command 449
RECORDER command 450
REFRESH CONVERTER command 451
REFRESH RACFUID command 453
SETOAUTO command 454
SETPWMC command 455
SETRACF command 457
SETRRS command. 458
SETUIDC command 459
STARTDS command 460
STARTIA command 461
STARTIP command 462
STARTMSC command 464
STARTOD command 465
STARTPT command 466
STARTRMT command 467
STOPCLNT command 469

STOPDS command 470
STOPIA command 471
STOPIP command 473
STOPLINK command 474
STOPMSC command 476
STOPOD command 477
STOPPORT command 478
STOPRMT command 480
STOPSCLN command 481
VIEWDS command 483
VIEWHWS command 488
VIEWIA command 512
VIEWIP command 515
VIEWMSC command 516
VIEWPORT command 521
VIEWRMT command 526
VIEWUOR command 530

Chapter 24. IMS Connect z/OS commands 533
IMS Connect z/OS command syntax 533
IMS Connect DELETE CLIENT command 534
IMS Connect DELETE LINK command 535
IMS Connect DELETE SENDCLNT command . . 537
IMS Connect QUERY ALIAS command 539
IMS Connect QUERY DATASTORE command . . 540
IMS Connect QUERY IMSPLEX command 545
IMS Connect QUERY MEMBER command . . . 547
IMS Connect QUERY MSC command 571
IMS Connect QUERY PORT command 576
IMS Connect QUERY RMTIMSCON command . . 581
IMS Connect QUERY UOR command 585
IMS Connect SHUTDOWN MEMBER command 587
IMS Connect UPDATE ALIAS command 588
IMS Connect UPDATE CONVERTER command 589
IMS Connect UPDATE DATASTORE command . . 591
IMS Connect UPDATE IMSPLEX command . . . 592
IMS Connect UPDATE MEMBER command . . . 594
IMS Connect UPDATE MSC command 597
IMS Connect UPDATE ODBM command 599
IMS Connect UPDATE PORT command 601
IMS Connect UPDATE RACFUID command . . . 602
IMS Connect UPDATE RMTIMSCON command 603

Part 7. IMS control region commands 607

Chapter 25. MODIFY imsproc commands 609
F jobname,DUMP 609
F jobname,DUMPxxxx 609
F jobname,FORCExxxx 610
F jobname,RECONNECT 611
F jobname,RECONNSTR 612
F jobname,STOP 613
F jobname,STOPxxxx 613

Part 8. IRLM commands 615

Chapter 26. MODIFY irlmproc commands 617
F irlmproc,ABEND 617
F irlmproc,DIAG,HANG 618
F irlmproc,PURGE,imsname 619
F irlmproc,SET 620
F irlmproc,STATUS 623

Chapter 27. START irlmproc command 629

Chapter 28. STOP irlmproc command 633

Chapter 29. TRACE CT command . . . 635

Part 9. Open Database Access commands 639

Chapter 30. CANCEL ODBA command 641

Chapter 31. FORCE ODBA command 643

Part 10. Repository Server commands 645

Chapter 32. MODIFY reposervername commands 647
F reposervername,ADMIN 647
F reposervername,AUDIT 651
F reposervername,SECURITY 652
F reposervername,SHUTDOWN 653

Chapter 33. P reposervername command 657

Part 11. Appendixes 659

Notices 661
Trademarks 663
Terms and conditions for product documentation 663
IBM Online Privacy Statement 664

Bibliography 665

Index X-1

About this information

These topics provide command syntax and usage information for various types of IMS™ commands, such as: Base Primitive Environment (BPE); Common Service Layer (CSL); Database Recovery Control (DBRC) facility; IMS Transport Manager Subsystem (TMS); Repository Server, IMS Connect; and the z/OS® commands for IMS. Information about IMS type-1 and type-2 commands is in *IMS Version 15 Commands, Volume 1: IMS Commands A-M* and *IMS Version 15 Commands, Volume 2: IMS Commands N-V*.

This information is available in IBM® Knowledge Center.

Prerequisite knowledge

Before using this information, you should have knowledge of either IMS Database Manager (DB) or IMS Transaction Manager (TM). You should also understand basic z/OS and IMS concepts, your installation's IMS system, and have general knowledge of the tasks involved in project planning.

Recommendation: Before using this information, you should be familiar with the following resources:

- *IMS Version 15 Operations and Automation*
- *z/OS JES2 Commands*
- *z/OS JES3 Commands*
- *z/OS MVS™ System Commands*

You can learn more about z/OS by visiting the "z/OS basic skills" topics in IBM Knowledge Center.

You can gain an understanding of basic IMS concepts by reading *An Introduction to IMS*, an IBM Press publication.

IBM offers a wide variety of classroom and self-study courses to help you learn IMS. For a complete list of courses available, go to the IBM Skills Gateway and search for IMS.

How new and changed information is identified

New and changed information in most IMS library PDF publications is denoted by a character (revision marker) in the left margin. The first edition (-00) of *Release Planning*, as well as the *Program Directory* and *Licensed Program Specifications*, do not include revision markers.

Revision markers follow these general conventions:

- Only technical changes are marked; style and grammatical changes are not marked.
- If part of an element, such as a paragraph, syntax diagram, list item, task step, or figure is changed, the entire element is marked with revision markers, even though only part of the element might have changed.
- If a topic is changed by more than 50%, the entire topic is marked with revision markers (so it might seem to be a new topic, even though it is not).

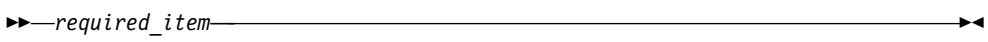
© Copyright IBM Corp. 1974, 2017

Revision markers do not necessarily indicate all the changes made to the information because deleted text and graphics cannot be marked with revision markers.

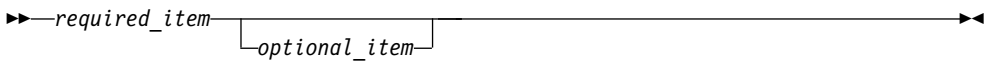
How to read syntax diagrams

The following rules apply to the syntax diagrams that are used in this information:

- Read the syntax diagrams from left to right, from top to bottom, following the path of the line. The following conventions are used:
  - The >>--- symbol indicates the beginning of a syntax diagram.
  - The ---> symbol indicates that the syntax diagram is continued on the next line.
  - The >--- symbol indicates that a syntax diagram is continued from the previous line.
  - The --->< symbol indicates the end of a syntax diagram.
- Required items appear on the horizontal line (the main path).



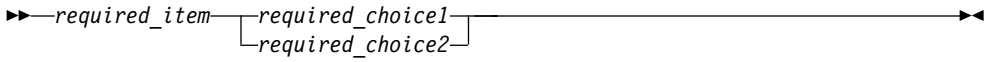
- Optional items appear below the main path.



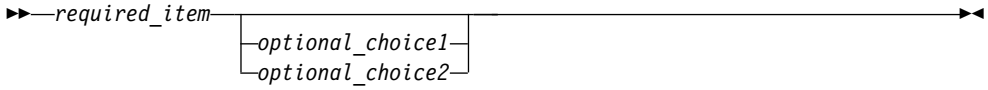
If an optional item appears above the main path, that item has no effect on the execution of the syntax element and is used only for readability.

- If you can choose from two or more items, they appear vertically, in a stack.

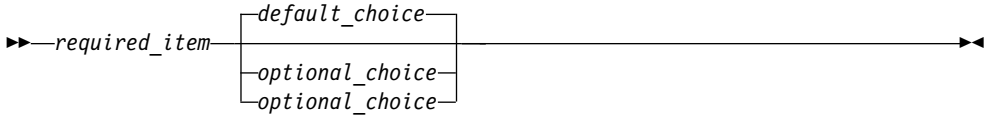
  If you *must* choose one of the items, one item of the stack appears on the main path.



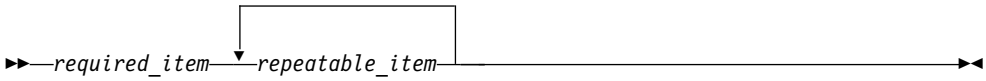
If choosing one of the items is optional, the entire stack appears below the main path.



If one of the items is the default, it appears above the main path, and the remaining choices are shown below.



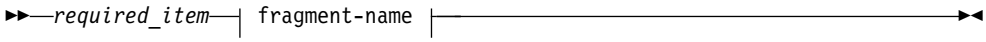
- An arrow returning to the left, above the main line, indicates an item that can be repeated.



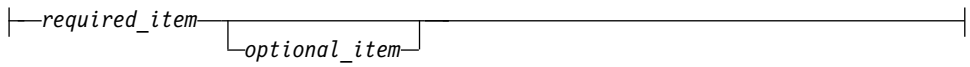
If the repeat arrow contains a comma, you must separate repeated items with a comma.

A repeat arrow above a stack indicates that you can repeat the items in the stack.

- Sometimes a diagram must be split into fragments. The syntax fragment is shown separately from the main syntax diagram, but the contents of the fragment should be read as if they are on the main path of the diagram.



fragment-name:



- In IMS, a b symbol indicates one blank position.
- Keywords, and their minimum abbreviations if applicable, appear in uppercase. They must be spelled exactly as shown. Variables appear in all lowercase italic letters (for example, *column-name*). They represent user-supplied names or values.
- Separate keywords and parameters by at least one space if no intervening punctuation is shown in the diagram.
- Enter punctuation marks, parentheses, arithmetic operators, and other symbols, exactly as shown in the diagram.
- Footnotes are shown by a number in parentheses, for example (1).

Accessibility features for IMS 15

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

Accessibility features

The following list includes the major accessibility features in z/OS products, including IMS 15. These features support:

- Keyboard-only operation.
- Interfaces that are commonly used by screen readers and screen magnifiers.
- Customization of display attributes such as color, contrast, and font size.

Keyboard navigation

You can access IMS 15 ISPF panel functions by using a keyboard or keyboard shortcut keys.

For information about navigating the IMS 15 ISPF panels using TSO/E or ISPF, refer to the *z/OS TSO/E Primer*, the *z/OS TSO/E User's Guide*, and the *z/OS ISPF User's Guide Volume 1*. These guides describe how to navigate each interface, including the use of keyboard shortcuts or function keys (PF keys). Each guide includes the default settings for the PF keys and explains how to modify their functions.

Related accessibility information

Online documentation for IMS 15 is available in IBM Knowledge Center.

IBM and accessibility

See the *IBM Human Ability and Accessibility Center* at www.ibm.com/able for more information about the commitment that IBM has to accessibility.

How to send your comments

Your feedback is important in helping us provide the most accurate and highest quality information. If you have any comments about this or any other IMS information, you can take one of the following actions:

- Click the **Contact Us** tab at the bottom of any IBM Knowledge Center topic.
- Send an email to imspubs@us.ibm.com. Be sure to include the book title and the publication number.

To help us respond quickly and accurately, please include as much information as you can about the content you are commenting on, where we can find it, and what your suggestions for improvement might be.

Part 1. Base Primitive Environment commands

By using Base Primitive Environment (BPE) commands, you can display and update resources that BPE manages.

Base Primitive Environment command information includes:

- Chapter 1, “Introduction to the BPE commands,” on page 3
- Chapter 2, “BPE DISPLAY VERSION command,” on page 7
- Chapter 3, “BPE TRACETABLE commands,” on page 9
- Chapter 4, “BPE USEREXIT commands,” on page 23

Related reference:

BPE configuration parameter member of the IMS PROCLIB data set (System Definition)

© Copyright IBM Corp. 1974, 2017

Chapter 1. Introduction to the BPE commands

BPE commands enable you to display and update resources that BPE manages. Some resource types are defined and owned by BPE itself. These resource types are known as *system resource types.*

Commands that specify system resource types can be issued to any IMS component running in a BPE environment. For example, BPE defines several BPE system trace table types such as DISP, STG, and CBS. These trace tables exist in every BPE address space. Commands to display and update these trace table types can be issued to any BPE address space.

Other resource types are defined and owned by the IMS component that is using BPE services. These resource types are known as *component resource types* or *user-product resource types.* Commands that specify component resource types can be issued only to the IMS component that defines those types. For example, CQS defines several CQS-specific trace tables such as STR, CQS, and INTF. Commands to display and update these trace table types can be issued only to CQS address spaces.

Subsections:

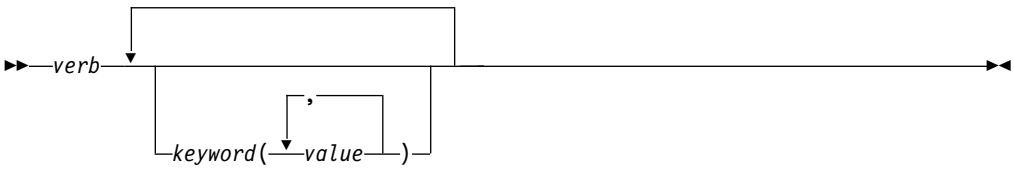
- "BPE command syntax and invocation"
- "BPE command invocation" on page 4
- "BPE wildcard character support" on page 4
- "Specifying IMS component command parameters" on page 5

BPE command syntax and invocation

BPE supports two command formats: a verb-only format, and a verb-resource type format.

The verb-only format consists of a verb, followed by zero or more keyword-value pairs, with the values enclosed in parentheses.

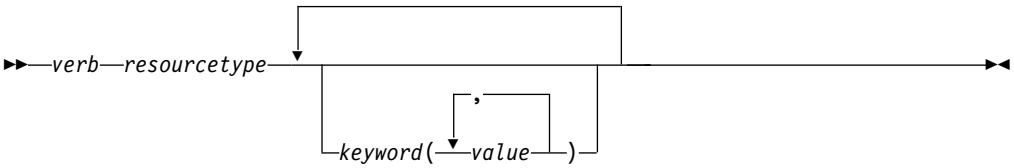
BPE verb-only command syntax



The verb-resource type format consists of a verb, a resource type, and zero or more keyword value pairs.

BPE verb-resource type command syntax

© Copyright IBM Corp. 1974, 2017



verb
A command verb representing an action. Some verb examples are DISPLAY, UPDATE, and REFRESH.

resourcetype
The type of resource that is operated on by the verb. Some resource examples are TRACETABLE and USEREXIT.

keyword(value)
A set of zero or more keywords and values that represent attributes, filters, or other modifiers that apply to the command. For example, NAME() to identify the specific resources or LEVEL() to specify a trace level.

BPE command invocation

Invoke BPE commands only through the z/OS MODIFY command. The following diagram illustrates the general syntax for entering commands through the z/OS modify interface.

BPE command invocation

F | MODIFY
The z/OS command MODIFY.

jobname
The jobname of the address space to which the command is directed.

command
The command being issued.

BPE wildcard character support

Some parameters on BPE commands support wildcard characters for pattern matching. For such parameters, you can use the following wildcard characters:

***** Matches any string with zero or more characters

% Matches exactly one character

The following examples illustrate some uses of wildcard characters.

BE* Matches any string beginning with "BE", of any length. For instance: BE, BEE, BEEBLEBROX.

%%S Matches any three-character string ending with an "S". For instance: IMS, CQS.

R*S*T%R
Matches any string beginning and ending with "R", having an "S", followed by a "T" in the middle, with any number of intervening characters between

the first "R", the "S", and the "T", and exactly one character between the "T" and the final "R". For example, ROASTER, ROSTER, RESORTER, RESCEPTOR, RSTZR.

Specifying IMS component command parameters

BPE commands provide the ability to restrict the resource types upon which a command operates to either those owned by BPE, or to those owned by the IMS component of the address space to which the command is issued. This is done through the OWNER keyword on commands that support OWNER. Use `OWNER(BPE)` to restrict the command operation to resource types that BPE owns and defines (system resource types). Use `OWNER(component_type)` to restrict the command operation to resource types that the IMS component address defines and owns (component resource types). The following table lists the valid values for the OWNER parameter, and the address space types to which they apply:

Table 1. Valid values for the OWNER parameter

OWNER	Address space type
BPE	Any IMS component running in a BPE address space
CQS	Common Queue Server
HWS	IMS Connect
ODBM	Open Database Manager
OM	Operations Manager
REPO	Repository Server
RM	Resource Manager
SCI	Structured Call Interface

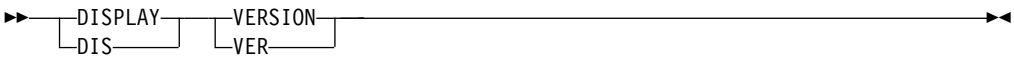
Chapter 2. BPE DISPLAY VERSION command

Use the BPE DISPLAY VERSION command to display both the version of the IMS component that is using BPE, and the version of the BPE in use.

Subsections:

- "Syntax"
- "Keywords"
- "Output"
- "Command example 1" on page 8
- "Command example 2" on page 8

Syntax



Keywords

DISPLAY | DIS
A required parameter, which specifies that the action against the specified resource is to display attributes of the resource.

VERSION | VER
A required parameter, which specifies that the resource types being acted upon are the version number of the IMS component and the BPE in the current address space.

Output

The DISPLAY VERSION command output consists of a single display output line in the following format:

```
BPE00001 comp VERSION=cv.cr.cp BPE VERSION=bv.br.bp
```

- *comp* is the IMS component ID for the address space. It is 1–4 characters long and can have one of the following values:
  - CQS (Common Queue Server)
  - DBRC (Database Recovery Control)
  - HWS (IMS Connect)
  - ODBM (Open Database Manager)
  - OM (Operations Manager)
  - REPO (Repository Server)
  - RM (Resource Manager)
  - SCI (Structured Call Interface)
- *cv.cr.cp* is the full version number of the IMS component, where *cv* is the version, *cr* is the release, and *cp* is the point release. Similarly, *bv.br.bp* indicates the full version number of the BPE running in the address space.

© Copyright IBM Corp. 1974, 2017

Command example 1

Here are some examples of using the DISPLAY VERSION command.

Display the version of a CQS address space.

Command:

```
F CQS1,DISPLAY VERSION
```

Output:

```
BPE0000I CQS VERSION = 1.6.0  BPE VERSION = 1.7.0
```

Command example 2

Display the version of an RM address space.

Command:

```
F RM1,DISPLAY VERSION
```

Output:

```
BPE0000I RM VERSION = 1.4.0  BPE VERSION = 1.7.0
```

Chapter 3. BPE TRACETABLE commands

The TRACETABLE resource type refers to the internal BPE-managed trace tables defined either by BPE (for example: DISP, CBS, STG, LATC), or by the IMS component using BPE (for example: CQS, ODBM, OM, RM, RS, SCI).

- "BPE DISPLAY TRACETABLE command"
- "BPE UPDATE TRACETABLE command" on page 15

BPE DISPLAY TRACETABLE command

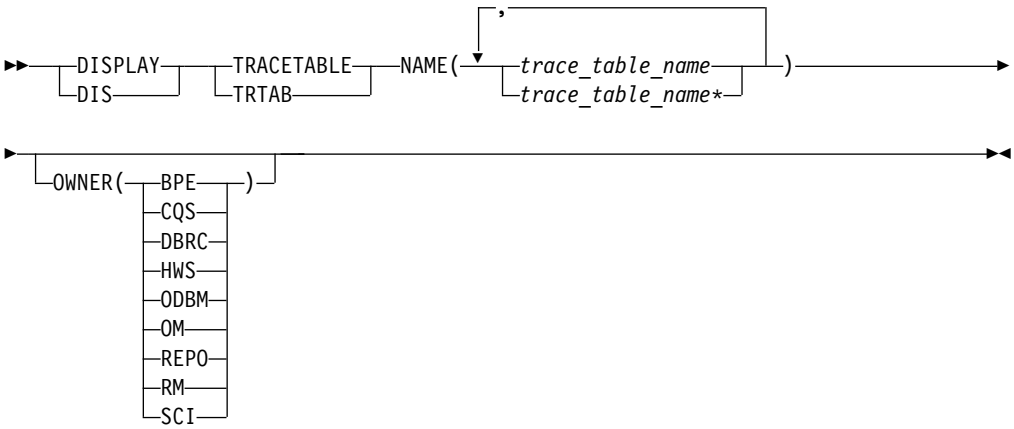
Use the DISPLAY TRACETABLE command to display trace level and number of trace table pages of specified trace tables.

Subsections:

- "Syntax"
- "Keywords"
- "Output" on page 12
- "Command example 1" on page 13
- "Command example 2" on page 13
- "Command example 3" on page 14
- "Command example 4" on page 14
- "Command example 5" on page 15
- "Command example 6" on page 15

Syntax

Use this command to display the current attribute settings for the requested trace tables.



Keywords

DISPLAY | DIS

A required parameter, which displays the attributes of the specified resource.

© Copyright IBM Corp. 1974, 2017

TRACETABLE | TRTAB
A required parameter, which specifies that the resource type being acted upon is a BPE-managed trace table.

NAME(*trace_table_name*)
A required parameter, which specifies the name of the trace table types about which you want attributes displayed. You can specify a single trace table name or a list of trace table names separated by commas. Trace table names can contain wildcard characters. Trace table names can be BPE-defined trace tables or IMS component-defined trace tables.

You can display BPE-defined trace tables for any IMS component address space that is using BPE. These BPE-defined trace table types are available:

AWE Asynchronous work element (AWE) trace table
CBS Control block services trace table
CMD Command trace table
DISP Dispatcher trace table
ERR BPE Error trace table
ERRV BPE error variable trace table
HASH
Hash trace table
LATC Latch trace table
MISC Miscellaneous trace table that is used only by IMS Service for trap traces
SSRV System services trace table
STG Storage service trace table
USRX User exit routine trace table

You can display CQS-defined trace tables only for CQS address spaces. These CQS-defined trace table types are available:

CQS CQS trace table
ERR CQS error trace table
INTF CQS interface trace table
OFLW CQS structure overflow trace table
SEVT CQS structure event trace table
STR CQS client activity trace table

You can display IMS Connect-defined trace tables only for IMS Connect address spaces. These IMS Connect-defined trace table types are available:

ADPT Adapter trace table
CMDT
IMS Connect command activity trace table
ENVT Environment trace table
ERRV IMS Connect error variable trace table
HWSI IMS Connect to OTMA driver trace table
HWSN
IMS Connect to local option driver trace table
HWSO
IMSplex driver (IPDC) trace table
HWSW
IMS Connect to TCP/IP driver trace table
LEPS Language Environment® (LE) preinitialization services trace table
ODBM
ODBM communication trace table
ODDR
ODBM driver trace table
OMDR
Communication protocol activity (SCI calls) trace table

OSDR ODBM socket driver trace table
OSOC ODBM socket communication trace table
OTMA
OTMA communication driver trace table
PCDR Local option driver trace table
PCIF PC interface trace table
RCTR Recorder trace table
RRSI RRS interface trace table
TCPI TCP/IP communication driver trace table

You can display OM-defined trace tables only for OM address spaces. These OM-defined trace table types are available:
CSL CSL trace table
ERR OM error trace table
OM OM processes trace table
PLEX IMSplex trace table for OM processing for a specific IMSplex

You can display RM-defined trace tables only for RM address spaces. These RM-defined trace table types are available:
CSL CSL trace table
ERR RM error trace table
PLEX IMSplex trace table for RM processing for a specific IMSplex
REPO RM repository services trace table
RM RM processes trace table

You can display SCI-defined trace tables only for SCI address spaces. These SCI-defined trace table types are available:
CSL Common Service Layer (CSL) trace table
ERPL SCI Error Parameter List trace table
ERR SCI error trace table
INTF SCI interface trace table
INTP SCI interface parameter trace table
PLEX IMSplex trace table for SCI processing for a specific IMSplex
SCI Structured Call Interface (SCI) processes trace table

You can display DBRC-defined trace tables only for DBRC address spaces. These DBRC-defined trace table types are available:
ERR DBRC error trace table
GRPS DBRC group services trace table
MODF
DBRC module flow trace table
RQST DBRC request trace table

You can display ODBM-defined trace tables only for ODBM address spaces. These ODBM-defined trace table types are available:
CSL CSL trace table
ERR ERR trace table
ODBM
ODBM trace table
PLEX IMSplex trace table

You can display Repository Server-defined trace tables only for REPO address spaces. These Repository Server (RS) trace table types are available:
DIAG
Repository Server diagnostics trace table

OWNER

An optional parameter that specifies the owner of the trace table type or types about which you want attributes displayed. You can specify one of the following values:

BPE For all IMS components that are running in a BPE address space
CQS For CQS address spaces only
DBRC For DBRC address spaces only
HWS For IMS Connect address spaces only
ODBM
For ODBM address spaces only
OM For OM address spaces only
REPO For RS address spaces only
RM For RM address spaces only
SCI For SCI address spaces only

The OWNER parameter acts as a filter to help you select which trace tables you want to display. For example, you could specify `NAME(*) OWNER(CQS)` to display all of the CQS-defined trace table types (CQS, ERR, OFLW, SEVT, STR, and INTF) in a CQS address space. You could specify `NAME(*) OWNER(BPE)` to display all of the BPE-defined trace table types in any BPE-managed address space. If `OWNER` is omitted, then both BPE and component trace tables might be displayed (depending on the tables specified on `NAME`).

Output

The DISPLAY TRACETABLE command output consists of a header line, one line per selected trace table, and one message BPE0032I line indicating that the command has completed. Here is an example.

```
F CQS1,DIS TRTAB NAME(*) OWNER(CQS)
BPE0030I TABLE  OWNER  LEVEL  #PAGES  EXT  #ENTRIES   #CYCLES  CQS1CQS
BPE0000I CQS    CQS    HIGH        4  NO         14         0  CQS1CQS
BPE0000I ERR    CQS    HIGH        4  NO         13         0  CQS1CQS
BPE0000I INTF   CQS    HIGH        8  NO          5         0  CQS1CQS
BPE0000I OFLW   CQS    HIGH       12  NO          4         0  CQS1CQS
BPE0000I SEVT   CQS    HIGH       12  NO          3         0  CQS1CQS
BPE0000I STR    CQS    HIGH        8  NO         87         0  CQS1CQS
BPE0032I DIS TRTAB COMMAND COMPLETED CQS1CQS
```

These columns are in the DISPLAY TRACETABLE output:

TABLE

Specifies the name of the trace table type about which information is being displayed on the current row. Either BPE or the product using BPE owns this trace table.

OWNER

Specifies the IMS component that owns the trace table. BPE-owned trace tables are system trace tables, and exist in all IMS component address spaces that use BPE. Trace tables that are specific to an IMS component show the 1- to 4-character component identifier in this column.

LEVEL

Specifies the current level setting of the trace table. The level of a trace table determines the volume of trace data collected. These levels are possible:

NONE

No trace data is being written to the table.

ERROR
Only traces for error or exception conditions are being written into the table.

LOW Only major event trace entries are written into the table.

MEDIUM
Major event trace entries and some minor event trace entries are written into the table.

HIGH All trace entries are written into the table.

INACTV
The trace table is inactive and cannot be used. This status occurs only when BPE was unable to get any storage for the trace table. No tracing is done for the indicated table type, and you cannot change the level for the trace table with the UPDATE TRACETABLE command. You must restart the address space in order to use the trace table again.

#PAGES
Specifies the number of 4K (4096 byte) pages allocated for the trace table type.

EXT Specifies whether trace entries for the trace tables specified by the NAME keyword should be written to the external trace data set. Values are YES or NO.

YES Specifies that trace entries are written both to the external trace data set and to memory.

NO Specifies that trace entries are written only to memory.

#ENTRIES
Specifies the number of trace entries that have been created in the trace table.

#CYCLES
Specifies the number of times the trace table has been filled and wrapped back to the top.

Command example 1

Here are some examples of using the DISPLAY TRACETABLE command.

The following command displays the status of the BPE dispatcher trace table (DISP).

Command:

```
F CQS1,DISPLAY TRACETABLE NAME(DISP)
```

Output:

```
BPE0030I TABLE  OWNER  LEVEL  #PAGES  EXT  #ENTRIES   #CYCLES
BPE0000I DISP   BPE    HIGH        8  NO       1916         1
BPE0032I DISPLAY TRACETABLE COMMAND COMPLETED
```

Command example 2

The following command displays the status of all CQS traces.

Command:

```
F CQS1,DIS TRTAB NAME(*) OWNER(CQS)
```

Output:

```
F CQS1,DIS TRTAB NAME(*) OWNER(CQS)
BPE0030I TABLE  OWNER  LEVEL  #PAGES  EXT  #ENTRIES   #CYCLES  CQS1CQS
BPE0000I CQS    CQS    HIGH        4  NO         14         0  CQS1CQS
BPE0000I ERR    CQS    HIGH        4  NO         13         0  CQS1CQS
BPE0000I INTF   CQS    HIGH        8  NO          5         0  CQS1CQS
BPE0000I OFLW   CQS    HIGH       12  NO          4         0  CQS1CQS
BPE0000I SEVT   CQS    HIGH       12  NO          3         0  CQS1CQS
BPE0000I STR    CQS    HIGH        8  NO         87         0  CQS1CQS
BPE0032I DIS TRTAB COMMAND COMPLETED CQS1CQS
```

Command example 3

The following command displays the status of all traces in an SCI address space.

Command:

```
 F SCI,DIS TRTAB NAME(*)
```

Output:

```
BPE0030I TABLE  OWNER  LEVEL  #PAGES  EXT  #ENTRIES   #CYCLES
BPE0000I AWE    BPE    HIGH        6  YES      2843         3
BPE0000I CBS    BPE    HIGH        6  YES      3456         4
BPE0000I CMD    BPE    HIGH        2  YES         1         0
BPE0000I CSL    SCI    HIGH        8  YES         0         0
BPE0000I DISP   BPE    HIGH        8  YES      5581         5
BPE0000I ERPL   SCI    HIGH        8  YES         1         0
BPE0000I ERR    BPE    HIGH        2  YES         1         0
BPE0000I ERR    SCI    HIGH        4  YES         1         0
BPE0000I ERRV   BPE    HIGH        8  YES         0         0
BPE0000I HASH   BPE    HIGH        8  YES      3432         3
BPE0000I INTF   SCI    HIGH        8  YES       874         0
BPE0000I INTP   SCI    HIGH       16  YES       873         3
BPE0000I LATC   BPE    HIGH        8  NO       3264         3
BPE0000I MISC   BPE    HIGH        1  NO          0         0
BPE0000I PLEX   SCI    HIGH        8  NO       1724         1
BPE0000I SCI    SCI    HIGH        8  NO          1         0
BPE0000I SSRV   BPE    HIGH        4  YES        28         0
BPE0000I STG    BPE    HIGH        8  NO        324         0
BPE0000I USRX   BPE    HIGH        4  NO          2         0
BPE0032I DIS TRTAB COMMAND COMPLETED
```

Command example 4

The following command displays the status of all OM traces.

Command:

```
F OM,DIS TRTAB NAME(*) OWNER(OM)
```

Output:

```
BPE0030I TABLE  OWNER  LEVEL  #PAGES  EXT  #ENTRIES   #CYCLES
BPE0000I CSL    OM     HIGH        4  YES         0         0
BPE0000I ERR    OM     HIGH        4  YES         0         0
BPE0000I OM     OM     HIGH        4  NO          1         0
BPE0000I PLEX   OM     HIGH        8  NO         20         0
BPE0032I DIS TRTAB COMMAND COMPLETED
```

Command example 5

The following command displays the status of the PLEX trace and all traces beginning with "C" in the RM address space.

Command:

```
F RM,DIS TRTAB NAME(PLEX,C*)
```

Output:

```
BPE0030I TABLE  OWNER  LEVEL  #PAGES  EXT  #ENTRIES   #CYCLES
BPE0000I CBS    BPE    HIGH        6  NO        466         0
BPE0000I CMD    BPE    HIGH        2  NO          1         0
BPE0000I CSL    RM     HIGH        4  NO          0         0
BPE0000I PLEX   RM     HIGH        8  NO        750         0
BPE0032I DIS TRTAB COMMAND COMPLETED
```

Command example 6

The following command displays the status of all ODBM traces.

The following command displays the status of all Repository Server traces.

Command:

```
F ODBM1,DIS TRTAB NAME(*) OWNER(ODBM)
F REPO1,DIS TRTAB NAME(*) OWNER(REPO)
```

Output:

```
BPE0030I TABLE  OWNER  LEVEL  #PAGES  EXT  #ENTRIES   #CYCLES
BPE0000I CSL    ODBM   ERROR       4  NO          0         0
BPE0000I ERR    ODBM   HIGH        4  NO          0         0
BPE0000I ODBM   ODBM   ERROR       4  NO          0         0
BPE0000I PLEX   ODBM   ERROR       8  NO          0         0
BPE0032I DIS TRTAB COMMAND COMPLETED
```

```
BPE0030I TABLE OWNER LEVEL #PAGES EXT #ENTRIES #CYCLES
BPE0000I DIAG  REPO  HIGH  300    NO  489      0
BPE0032I DIS TRTAB COMMAND COMPLETED
```

Related reference:

Chapter 1, “Introduction to the BPE commands,” on page 3

Base Primitive Environment customization exit routines (Exit Routines)

BPE UPDATE TRACETABLE command

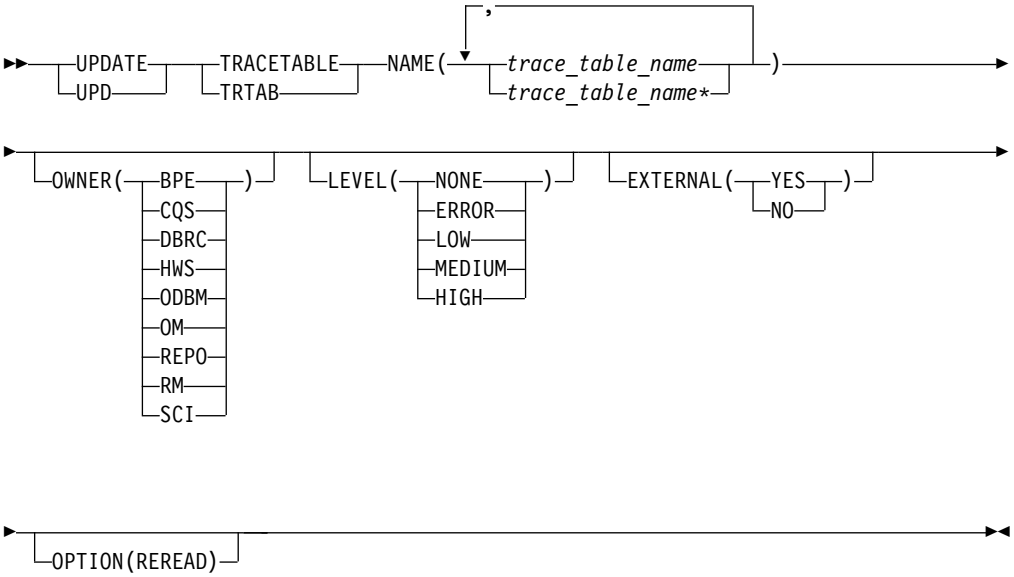
Use the UPDATE TRACETABLE command to update trace level attributes of the specified trace tables.

Subsections:

- “Syntax” on page 16
- “Keywords” on page 16
- “Output” on page 20
- “Examples” on page 20

Syntax

Use this command to change the trace level setting for the requested trace tables.



Keywords

UPDATE | UPD
A required parameter, which specifies that the action against the trace table is to update its attributes.

TRACETABLE | TRTAB
A required parameter, which specifies that the resource type being acted upon is a BPE-managed trace table.

NAME(*trace_table_name*)
A required parameter that specifies the name of the trace table type or types that you want to update. You can specify a single trace table name or a list of trace table names separated by commas. Trace table names can contain wildcard characters. Trace table names can be BPE-defined trace tables or IMS component-defined trace tables.

Note: Trace table types that require explicit commands, for example the HWS RCTR table, is not processed if a generic name, NAME(*), or wildcard pattern, NAME(R%%%), is specified for the trace table name. Trace table types that require explicit commands are processed only when they have been explicitly specified with the NAME parameter: NAME(RCTR).

You can update BPE-defined trace tables for any IMS component address space that is using BPE. These BPE-defined trace table types are available:
AWE Asynchronous work element (AWE) trace table
CBS Control block services trace table
CMD Command trace table
DISP Dispatcher trace table
ERR BPE Error trace table
ERRV BPE error variable trace table

HASH
Hash trace table
LATC Latch trace table
MISC Miscellaneous trace table that is used only by IMS Service for trap traces
SSRV System services trace table
STG Storage service trace table
USRX User exit routine trace table

You can update CQS-defined trace tables only for CQS address spaces. These CQS-defined trace table types are available:
CQS CQS trace table
ERR CQS error trace table
INTF CQS interface trace table
OFLW CQS structure overflow trace table
SEVT CQS structure event trace table
STR CQS client activity trace table

Note: A CQS client activity trace table contains trace records from only one CQS structure.

You can update IMS Connect-defined trace tables only for IMS Connect address spaces. These IMS Connect-defined trace table types are available:
ADPT Adapter trace table
CMDT
IMS Connect command activity trace table
CPDR CICSPORT driver trace table
CPRT CICSPORT communication trace table
ENVT Environment trace table
ERRV IMS Connect error variable trace table
GSVC General service trace table
HWSI IMS Connect to OTMA driver trace table
HWSN
IMS Connect to local option driver trace table
HWSO
IMSplex driver (IPDC) trace table
HWSW
IMS Connect to TCP/IP driver trace table
ICDR IMS Connect to IMS Connect driver trace table
ICNC IMS Connect to IMS Connect communication trace table
ISCC ISC communication trace table
ISDR ISC driver trace table
LEPS Language Environment (LE) preinitialization services trace table
MSCC
MSC communication trace table
MSDR
MSC driver trace table
ODBM
ODBM communication trace table
ODDR
ODBM driver trace table
OMDR
Communication protocol activity (SCI calls) trace table
OSDR ODBM socket driver trace table
OSOC ODBM socket communication trace table
OTMA
OTMA communication driver trace table

PCDR Local option driver trace table
PCIF PC interface trace table
RCDR RMTCICS driver trace table
RCIC RMTCICS communication trace table
RCTR Recorder trace table

Note: The Recorder trace table requires explicit commands.
RRSI RRS interface trace table
TCPI TCP/IP communication driver trace table

You can update OM-defined trace tables only for OM address spaces. These OM-defined trace table types are available:

CSL CSL trace table

ERR OM error trace table

OM OM processes trace table

PLEX IMSplex trace table for OM processing for a specific IMSplex

You can update RM-defined trace tables only for RM address spaces. These RM-defined trace table types are available:

CSL CSL trace table

ERR RM error trace table

PLEX IMSplex trace table for RM processing for a specific IMSplex

REPO RM repository services trace table

RM RM processes trace table

You can update SCI-defined trace tables only for SCI address spaces. These SCI-defined trace table types are available:

CSL CSL trace table

ERPL SCI error parameter list trace table

ERR SCI error trace table

INTF SCI interface trace table

INTP SCI interface parameter trace table

PLEX IMSplex trace table for SCI processing for a specific IMSplex

SCI SCI processes trace table

You can update DBRC-defined trace tables only for DBRC address spaces. These DBRC-defined trace table types are available:

ERR DBRC error trace table

GRPS DBRC group services trace table

MODF
DBRC module flow trace table

RQST DBRC request trace table

You can update ODBM-defined trace tables only for ODBM address spaces. These ODBM-defined trace table types are available:

CSL CSL trace table

ERR Error trace table

ODBM
ODBM trace table

PLEX IMSplex trace table

You can update Repository Server-defined trace tables only for REPO address spaces. These Repository Server (RS) trace table types are available:

DIAG Repository Server diagnostics trace table

OWNER
An optional parameter that specifies the owner of the trace table type or types that you want to update. You can specify one of the following values:

BPE For all IMS components that are running in a BPE address space

CQS For CQS address spaces only

DBRC For DBRC address spaces only

HWS For HWS address spaces only

ODBM
For ODBM address spaces only

OM For OM address spaces only

REPO For RS address spaces only

RM For RM address spaces only

SCI For SCI address spaces only

The OWNER parameter acts as a filter to help you select which trace tables you want to update. For example, you can specify NAME(*) OWNER(CQS) to update all of the CQS-defined trace table types (CQS, ERR, STR, OFLW, SEVT, and INTF) in a CQS address space. You can specify NAME(*) OWNER(BPE) to update all of the BPE-defined trace table types in any BPE-managed address space. If OWNER is omitted, then both BPE and component trace tables might be updated (depending on the tables that are specified on NAME).

LEVEL(*level*)
An optional parameter that sets the new tracing level for the specified trace tables. If LEVEL is omitted, the level of the specified trace tables is not changed. These levels are possible:

NONE
No trace data is being written to the table.

ERROR
Only traces for error or exception conditions are being written into the table.

LOW Only major event trace entries are written into the table.

MEDIUM
Major event trace entries and some minor event trace entries are written into the table.

HIGH All trace entries are written into the table.

Important: You cannot change the level for the trace table type ERR. BPE forces the level to HIGH to ensure that error diagnostics are captured. Any level that you specify for the ERR trace table is ignored.

EXTERNAL
Specifies whether trace entries for the trace tables that are specified by the NAME keyword should be written to the external trace data set. Values are YES or NO.

YES Specifies that trace entries are written both to the external trace data set and to memory. Trace entries are written to the external data set only if you define that data set to IMS by including the EXTTRACE statement in the BPE configuration parameter member (BPECFG=) of the IMS PROCLIB data set.

NO Specifies that trace entries are written only to memory.

If you do not include the EXTERNAL keyword when you issue the command, IMS does not change the current external trace setting (if any) for the affected tables.

You must create or define the data set specified on the EXTTRACE statement. If you do not create or define the data set, an error is generated when you try to open the data set.

OPTION(REREAD)
Specifies that the BPE configuration PROCLIB member that was specified on the BPECFG= parm of the job is reread to obtain updated EXTTRACE settings. Only the BPE configuration PROCLIB member is reread.

The EXTTRACE parameter values in the PROCLIB member at the time it is reread replace the values that were read during initialization. If there is a current external trace data set open, it is closed and deallocated. A new external trace data set is then allocated and opened using the updated EXTTRACE parameters. Other parameters within BPE PROCLIB member are not reprocessed when the member is reread.

With OPTION(REREAD) you can either specify a new external trace data set or modify the characteristics of an existing trace data set without bringing down the address space.

OPTION(REREAD) requires that the address space was started with BPECFG= that specifies a BPE configuration PROCLIB member. If no member was specified, there is no member to reread, and the command will be rejected. If the BPE configuration PROCLIB member is reread and no EXTTRACE statement is found that applies to the current address space, the command is rejected and the existing external trace settings, if any, are kept. If the member is reread and errors occur during allocation or open of the new external trace data set (for example, if the specified data set name does not exist), BPE external trace is stopped. Edit the BPE PROCLIB member and correct the EXTTRACE statement. You can then re-enable BPE external trace by reentering the UPDATE TRACETABLE command with OPTION(REREAD).

Output

The UPDATE TRACETABLE command output consists of message BPE0032I indicating that the command has completed:

```
BPE0032I UPDATE TRACETABLE COMMAND COMPLETED
```

Examples

Command example 1

Update the level of the BPE dispatcher trace table (DISP) to HIGH.

Command:

```
F CQS1,UPDATE TRACETABLE NAME(DISP) LEVEL(HIGH)
```

Output:

```
BPE0032I UPDATE TRACETABLE COMMAND COMPLETED
```

Command example 2

Update the level of all SCI trace tables to MEDIUM.

Important:

- You cannot change the level for the trace table type ERR, even when using a wildcard character to select all tables with a given owner, as shown in this example--NAME(*). BPE forces the level to HIGH to ensure that error diagnostics are captured. The level for trace table type ERR is set to HIGH, but other trace table types are set to MEDIUM.
- You cannot change the level for a trace table type that requires explicit commands when using a wildcard character to select all tables with a given owner, as shown in this example--NAME(*). Trace table types that require explicit commands are processed only when they have been explicitly specified with the NAME parameter.

Command:

```
F SCI,UPD TRTAB NAME(*) OWNER(SCI) LEVEL(MEDIUM)
```

Output:

```
BPE0032I UPDATE TRACETABLE COMMAND COMPLETED
```

Command example 3

Set the level of the IMS Connect Recorder Trace to HIGH to capture message data that IMS Connect sends to and receives from TCP/IP endpoints and OTMA.

Command:

```
F HWS1,UPDATE TRACETABLE NAME(RCTR) OWNER(HWS) LEVEL(HIGH) EXTERNAL(YES)
```

Output:

```
BPE0032I UPDATE TRACETABLE COMMAND COMPLETED
BPE0046I EXTERNAL TRACE DATA SET
IMSTESTL.RCTR.GDG01.G0003V00 OPENED ON VOL=000000
```

Related concepts:

Configuring the IMS Base Primitive Environment for IMS Connect (System Definition)

Related reference:

BPE configuration parameter member of the IMS PROCLIB data set (System Definition)

Chapter 1, “Introduction to the BPE commands,” on page 3

Base Primitive Environment customization exit routines (Exit Routines)

Chapter 4. BPE USEREXIT commands

The USEREXIT resource type refers to the user exit types defined to and managed by either BPE or the IMS component using BPE (for example, CQS).

Note: Throughout this topic, the term *user exit routine* means *user-supplied exit routine*.

- "BPE DISPLAY USEREXIT command"
- "BPE REFRESH USEREXIT command" on page 30

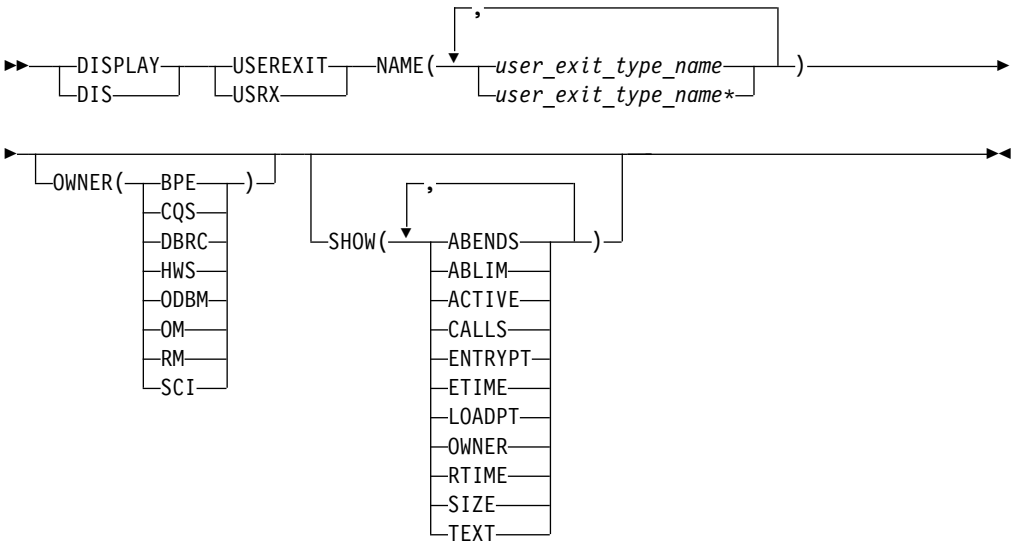
BPE DISPLAY USEREXIT command

Use the BPE DISPLAY USEREXIT command to display attributes of specified user exit types.

Subsections:

- "Syntax"
- "Keywords" on page 24
- "Output" on page 28
- "Command example 1" on page 28
- "Command example 2" on page 29
- "Command example 3" on page 29
- "Command example 4" on page 29
- "Command example 5" on page 30

Syntax



© Copyright IBM Corp. 1974, 2017

Keywords

DISPLAY | DIS
A required parameter, which specifies that the action against the specified resource is to display attributes of the resource.

USEREXIT | USRX
A required parameter, which specifies that the resource type being acted upon is a BPE-managed user exit type.

NAME(*user_exit_type_name*)
A required parameter, which specifies the name of the user exit type or types about which you want attributes displayed. You can specify a single user exit type name or a list of user exit type names separated by commas. User exit type names can contain wildcard characters.

Important: The name or names specified in this parameter are the names of user exit types, *not* the names of individual user exit modules.

BPE and each address space that can use BPE have different user exit types. As specified by OWNER(BPE), BPE's user exit types include:

INITTERM
Initialization-Termination user exit

STATS
BPE system functions statistics user exit

As specified by OWNER(CQS), the following user exit types are defined in all CQS address spaces:

CLNTCONN
Client Connection user exit

INITTERM
Initialization-Termination user exit

OVERFLOW
Queue Overflow user exit

STRSTAT
Structure statistics user exit

STREVENT
Structure event user exit

As specified by OWNER(DBRC), the following user exit types are defined in all DBRC BPE address spaces:

RECONIO
BPE RECON I/O user exit

REQUEST
BPE DBRC request user exit

SECURITY
BPE DBRC security user exit

As specified by OWNER(HWS), the following user exit types are defined in all IMS Connect address spaces:

POR*nnnnn*
IMS Connect Port Message Edit exit. *nnnnn* in the exit type name is the port number in decimal—for example, POR01234. The Port Message Edit exit is a BPE type-2 exit.

XMLADAP
IMS Connect XML adapter user exit

ODBMAUTH
IMS Connect DB security user exit, which enables users to access IMS database resources via ODBA from outside of an IMS dependent region.

ODBMROUT
IMS Connect Routing exit for ODBM, which enables users to override the IMS alias or to select an ODBM.

As specified by OWNER(ODBM), the following user exit types are defined in all ODBM address spaces:

CLNTCONN
ODBM user exit that is called during client registration and deregistration.

INITTERM
ODBM user exit that is called during initialization and termination of the ODBM address space.

INPUT
ODBM user exit that is called to view a CSLDMI FUNC=ODBMCI request to ODBM. This type of user exit can modify the input prior to execution or reject the request before it is processed.

OUTPUT
ODBM user exit that is called to view output (for example, ODBA call output) going out from ODBM to the ODBM client in response to a CSLDMI FUNC=ODBMCI request. This type of user exit can also modify the output before it is returned to the originator.

As specified by OWNER(OM), the following user exit types are defined in all OM address spaces:

CLNTCONN
Client Connection command registration and deregistration user exit

INITTERM
Initialization-Termination user exit

INPUT
Command input user exit

OUTPUT
Output user exit

SECURITY
Security checking user exit

As specified by OWNER(RM), the following user exit types are defined in all RM address spaces:

CLNTCONN
Client Connection and Disconnection user exit

INITTERM
Initialization-Termination user exit

As specified by OWNER(SCI), the following user exit types are defined in all SCI address spaces:

CLNTCONN
Client Connection and Disconnection user exit

INITTERM
Initialization-Termination user exit

OWNER(BPE | CQS | DBRC | HWS | ODBM | OM | RM | SCI)
An optional parameter that specifies the owner of the user exit type or types about which you want attributes displayed. You can specify one of the following values:

BPE
For all IMS components that are running in a BPE address space

CQS
For CQS address spaces only

DBRC
For DBRC address spaces only

HWS
For IMS Connect address spaces only

ODBM
For ODBM address spaces only

OM For OM address spaces only

RM For RM address spaces only

SCI
For SCI address spaces only

The OWNER parameter acts as a filter to help you select the user exit types that you want to display. For example, you could specify `NAME(*) OWNER(CQS)` to display all of the CQS-defined user exit types in a CQS address space. If `OWNER` is omitted, then both BPE and component user exits can be displayed (depending on the exits specified on `NAME`).

SHOW(*attribute*)
An optional parameter that specifies the attributes you want to display about the requested user exits.

When you display information about user exits, each row of display output contains the requested attributes for one user exit module, in columns. Every display for user exits contains the columns labeled EXITTYPE (the type of the exit), and MODULE (the load module name of the exit). Additionally, any of the following attributes can be requested by using the SHOW parameter:

ABENDS
The number of abends that have occurred in the user exit module since the last user exit refresh of that module (or since address space initialization if no refreshes have occurred). BPE keeps track of the number of abends that have occurred in each user exit module. When this number reaches the number defined on the ABLIM= parameter of the EXITDEF statement for the exit's type, BPE stops calling the module. If the user exit module is refreshed, this count is reset to zero, and BPE calls the module again.

The maximum value that can be displayed in this field is 2147483647 (2^{31}-1). If the abend count exceeds this value, 2147483647 is displayed.

ABLIM
The abend limit count for the user exit type, as specified on the ABLIM= parameter on the EXITDEF statement for the user exit type in the BPE exit list PROCLIB member. This is the number of times the user exit module is allowed to abend before BPE stops calling the user exit. A value of 0 indicates that there is no abend limit.

The maximum value that can be displayed in this field is 2147483647 (2^{31}-1).

ACTIVE
The number of currently active instances of the user exit. This is a point-in-time number that represents the number of calls to the user exit that have not yet returned.

The maximum value that can be displayed in this field is 999999. If the active count exceeds this value, 999999 is displayed.

CALLS
The number of calls to the user exit since the last user exit refresh.

For performance reasons, serialization is not obtained when BPE collects this number. For an exit type that can run multiple instances in parallel, this number should be considered an approximation only.

The maximum value that can be displayed in this field is 2147483647 (2^{31}-1). If the call count exceeds this value, 2147483647 is displayed.

ENTRYPT
The entry point address of the user exit module.

ETIME
The total (cumulative) elapsed time spent in the exit module since it was last refreshed, in milliseconds.

For performance reasons, serialization is not obtained when BPE collects this number. For an exit type that can run multiple instances in parallel, this number should be considered an approximation only.

The maximum value that can be displayed in this field is 2147483647 (2^{31}-1). If the elapsed number of milliseconds exceeds this value, 2147483647 is displayed.

LOADPT
The load point address of the user exit module.

OWNER
The IMS component that owns the user exit type. BPE-owned user exit types are system exit types that exist in all IMS component address spaces that use BPE. User exit types that are specific to the component show the 1- to 4-character component identifier in this column (for example, CQS).

RTIME
This is the local date and time that the user exit module was last refreshed (or initially loaded, if no refreshes have occurred). The format of this output field is:

```
yyyy-mm-dd hh:mm:ss.th
```

SIZE The size of the user exit load module, in bytes (displayed in hexadecimal).

TEXT 27 bytes starting from offset +04 from the module's entry point,

translated to EBCDIC, with non-printable characters replaced by periods (.). This is a common location for module identification information. If your user exits contain printable identification data at this point in the module, the TEXT option enables that information to be displayed.

If the SHOW parameter is not specified, the default attributes displayed after the EXITTYPE and MODULE are OWNER, ACTIVE, and ABENDS.

The order in which you list the attributes on the SHOW parameter has no effect on the order the attributes are displayed. BPE determines the order of the attribute columns in the display output. This order is as follows:

1. OWNER
2. ACTIVE
3. ABENDS
4. ABLIM
5. CALLS
6. ETIME
7. RTIME
8. ENTRYPT
9. LOADPT
10. SIZE
11. TEXT

Important: It is possible to request so many attributes that the length of the output line is too long to display with a WTO. If this happens, the command is processed, but some lines might be truncated. The maximum line length that BPE displays is 126 characters.

Output

The DISPLAY USEREXIT command output consists of a header line, one line per user exit module about which information is being displayed, and one message BPE0032I line that indicates that the command completed.

Command:

```
F CQS1,DISPLAY USEREXIT NAME(INITTERM,STRSTAT)
```

Output:

```
BPE0030I EXITTYPE MODULE   OWNER    ACTIVE     ABENDS
BPE0000I INITTERM MYINIT00 CQS           0          0
BPE0000I INITTERM ZZZINIT0 CQS           0          0
BPE0000I STRSTAT  MYSTAT00 CQS           1          2
BPE0032I DISPLAY USEREXIT COMMAND COMPLETED
```

The EXITTYPE and MODULE columns are present for all DISPLAY USEREXIT commands, regardless of what is specified on SHOW. When multiple exit modules are listed for a single user exit type, they are listed in the order in which they are called.

Command example 1

Here are some examples of using the DISPLAY USEREXIT command.

Display the status of the CQS structure event user exit type.

Command:

```
F CQS1,DISPLAY USEREXIT NAME(STREVENT)
```

Output:

```
BPE0030I EXITTYPE MODULE   OWNER    ACTIVE     ABENDS
BPE0000I STREVENT STREVX00 CQS           1          0
BPE0000I STREVENT ZZZSTEV0 CQS           0          0
BPE0032I DISPLAY USEREXIT COMMAND COMPLETED
```

In this example, there are two structure event exit modules defined that are called for CQS structure events. `STREVX00` is called first, followed by `ZZZSTEV0`.

Command example 2

Display the number of calls to, the elapsed time spent in, and the abend limit for all CQS user exit types.

Command:

```
F CQS1,DIS USRX NAME(*) OWNER(CQS) SHOW(CALLS,ETIME,ABLIM)
```

Output:

```
BPE0030I EXITTYPE MODULE        ABLIM      CALLS      ETIME
BPE0000I CLNTCONN CLCONX00          0          2         12
BPE0000I INITTERM MYCQSIT0          0          1          2
BPE0000I INITTERM OEMCQIT0          0          1        162
BPE0000I OVERFLOW OVERFL01          5          3          6
BPE0000I OVERFLOW OVERFL02          5          3         19
BPE0000I OVERFLOW OVERFL03          5          3          9
BPE0000I OVERFLOW OVERFL04          5          3      15593
BPE0000I STREVENT STREVNT0         10        542        628
BPE0000I STRSTAT  STRSTAT0          1         36       1889
BPE0000I STRSTAT  STRSTA10          1         36        241
BPE0032I DIS USRX COMMAND COMPLETED
```

Command example 3

Display the entry point, load point, and size of all of the SCI CLNTCONN user exit modules.

Command:

```
F SCI,DIS USRX NAME(CLNTCONN) SHOW(SIZE,ENTRYPT,LOADPT)
```

Output:

```
BPE0030I EXITTYPE MODULE    ENTRYPT   LOADPT     SIZE
BPE0000I CLNTCONN SCCLCN00 8B864D78 8B864D78 00000458
BPE0000I CLNTCONN SCCLCN10 8BA14200 8BA14200 00001C10
BPE0000I CLNTCONN SCCLCN20 8BA18EE8 8BA18AF0 00000AB0
BPE0032I DIS USRX COMMAND COMPLETED
```

Command example 4

Display the first part of the module text for all of the BPE user exits in the OM address space.

Command:

```
F OM,DIS USRX NAME(*) OWNER(BPE) SHOW(TEXT)
```

Output:

```
BPE0030I EXITTYPE  MODULE   TEXT
BPE0000I INITTERM  MYINIT00 .MYINIT00+20010615+17:47...
BPE0000I STATS     HHGSTAT0 .HHGSTAT0+20010615+08:47...
BPE0032I DIS USRX COMMAND COMPLETED
```

Command example 5

Display the refresh time for all of the RM INITTERM modules.

Command:

```
F RM,DIS USRX NAME(*) OWNER(RM) SHOW(RTIME)
```

Output:

```
BPE0030I EXITTYPE MODULE   RTIME
BPE0000I INITTERM RMINITRM 2001-06-15 16:48:22.39
BPE0032I DIS USRX COMMAND COMPLETED
```

Related reference:

Chapter 1, “Introduction to the BPE commands,” on page 3

Base Primitive Environment customization exit routines (Exit Routines)

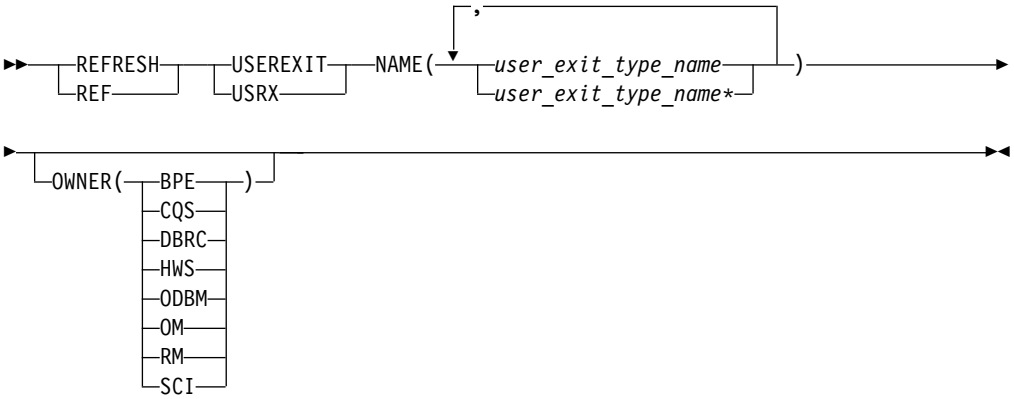
BPE REFRESH USEREXIT command

Use the BPE REFRESH USEREXIT command to load new copies of the user exit modules for specified user exit types.

Subsections:

- “Syntax”
- “Keywords” on page 31
- “Usage notes” on page 33
- “Output” on page 35
- “Examples” on page 35

Syntax



Keywords

REFRESH | REF
A required parameter, which specifies that the action against the specified resources is to refresh the resources.

USEREXIT | USRX
A required parameter, which specifies that the resource type being acted upon is a BPE-managed user exit type.

NAME(*user_exit_type_name***)**
A required parameter, which specifies the name of the user exit type or types that you want to refresh. You can specify a single user exit type name or a list of user exit type names separated by commas. User exit type names can contain wildcard characters.

Unless otherwise stated, all user exit types described here are type-1 user exits.

Important: The names specified in this parameter are the names of user exit types, *not* the names of individual user exit modules.

BPE and each address space that can use BPE have different user exit types. BPE's user exit types, as specified by OWNER(BPE), include the following:

INITTERM
Initialization-Termination user exit

STATS
BPE system functions statistics user exit

User exit types are defined in all CQS address spaces, as specified by OWNER(CQS), and include the following:

CLNTCONN
Client Connection user exit

INITTERM
Initialization-Termination user exit

OVERFLOW
Queue Overflow user exit

STRSTAT
Structure statistics user exit

STREVENT
Structure event user exit

User exit types are defined in all BPE-based DBRC address spaces, as specified by OWNER(DBRC), and include the following:

RECONIO
BPE RECON I/O user exit

REQUEST
BPE DBRC request user exit

SECURITY
BPE DBRC security user exit

User exit types are defined in all IMS Connect address spaces, as specified by OWNER(HWS), and include the following:

POR*nnnnn*
IMS Connect Port Message Edit exit. *nnnnn* in the exit type name is the port number in decimal—for example, POR01234. The Port Message Edit exit is a BPE type-2 exit.

XMLADAP
IMS Connect XML adapter user exit

ODBMAUTH
IMS Connect DB security user exit, which enables users to access IMS database resources via ODBA from outside of an IMS dependent region.

ODBMROUT
IMS Connect Routing exit for ODBM, which enables users to override the IMS alias or to select an ODBM.

User exit types are defined in all ODBM address spaces, as specified by OWNER(ODBM), and include the following:

CLNTCONN
ODBM user exit that is called during client registration and deregistration.

INITTERM
ODBM user exit that is called during initialization and termination of the ODBM address space.

INPUT
ODBM user exit that is called to view a CSLDMI FUNC=ODBMCI request to ODBM. This type of user exit can modify the input prior to execution or reject the request before it is processed.

OUTPUT
ODBM user exit that is called to view output (for example, ODBA call output) going out from ODBM to the ODBM client in response to a CSLDMI FUNC=ODBMCI request. This type of user exit can also modify the output before it is returned to the originator.

User exit types are defined in all OM address spaces, as specified by OWNER(OM), and include the following:

CLNTCONN
Client Connection command registration and deregistration user exit

INITTERM
Initialization-Termination user exit

INPUT
Command input user exit

OUTPUT
Output user exit

SECURITY
Security checking user exit

User exit types are defined in all RM address spaces, as specified by OWNER(RM), and include the following:

CLNTCONN
Client Connection and Disconnection user exit

INITTERM
Initialization-Termination user exit

User exit types are defined in all SCI address spaces, as specified by OWNER(SCI), and include the following:

CLNTCONN
Client Connection and Disconnection user exit

INITTERM
Initialization-Termination user exit

OWNER(BPE | CQS | DBRC | HWS | ODBM | OM | RM | SCI)
An optional parameter that specifies the owner of the user exit type or types that you want to refresh. You can specify one of the following values:

BPE
For all IMS components that are running in a BPE address space.

CQS
For CQS address spaces only.

DBRC
For DBRC address spaces only.

HWS
For IMS Connect address spaces only.

ODBM
For ODBM address spaces only.

OM For OM address spaces only.

RM For RM address spaces only.

SCI
For SCI address spaces only.

The OWNER parameter acts as a filter to help you select the user exit types that you want to refresh. For example, you could specify `NAME(*) OWNER(CQS)` to refresh all of the CQS-defined user exit types in a CQS address space. If `OWNER` is omitted, then both BPE and component user exits can be refreshed (depending on the exits specified on `NAME`).

Usage notes

There are two kinds of BPE user exit routines: type 1 and type 2. Type-1 exits are defined to BPE in BPE exit list PROCLIB members. These members are processed at address space initialization, and processed again when a REFRESH USEREXIT command is issued. This allows the exit specifications to be changed and updated while a BPE address space is running.

Type-2 exits are defined to BPE programmatically at execution time. Type-2 exit type names are typically generated by the IMS component using BPE based on other specifications outside of BPE's control. For example, the name of the IMS Connect Port Message Edit exit type is generated using the TCP/IP port number—for instance, POR01234 for port 1234. Type-2 exits are not specified in the BPE exit list PROCLIB member.

For type-1 exits, the REFRESH USEREXIT command does two things. It causes BPE to reprocess the user exit PROCLIB members specified in the BPE configuration PROCLIB member. It also reloads the user exit modules currently

listed in the user exit PROCLIB members for the types specified on the command. By using this command, you can update your user exits without stopping and restarting the address space.

For type-2 exits, the REFRESH USEREXIT command only reloads the user exit modules that were defined to BPE at execution time. Because type-2 exits are not defined externally to BPE, the definitional values (module names, abend limits) of the exits are not changed.

When you enter the REFRESH USEREXIT command, BPE performs the following processing:

- For type-1 exits:
  - Reads any user exit PROCLIB members that are specified on EXITMBR= statements in the BPE configuration PROCLIB member. Because BPE re-reads these members at the time you issue the command, you can edit the user exit PROCLIB members before issuing the REFRESH command and make changes to the user exit definitions. BPE does not re-read the main BPE configuration PROCLIB member, so you cannot change the names of the user exit PROCLIB members; you can change only their contents.
  - Loads the user exit modules specified on the EXITDEF= statements for the user exit types specified on the command.
- For type-2 exits:
  - Loads new copies of the user exit modules associated with the user exit types specified on the command.
- Quiesces all current user exits. This means that the command waits for any active exits to complete processing and delays any new calls to the current exits. Quiescing all current user exits ensures that no user exit is running while the exit is being refreshed.
- Replaces BPE control block pointers to the previous user exit modules with pointers to the newly loaded modules. These pointers are used to manage the calling of the exits.
- Resumes the user exits and enables calls to be made to the newly-loaded exits.
- Deletes the old copy of the user exits.

BPE loads the new copies of the user exit modules before deleting the old modules. If an error occurs during this process (for example, a module could not be loaded or BPE internal control block storage could not be obtained), BPE fails the command and leaves the old copies of the user exits in effect. All modules of the specified user exit types must be loaded successfully for the command to complete successfully.

When a user exit module is refreshed, its abend count is reset to zero. This means that a user exit module that had reached its abend limit (specified by the ABLIM parameter on the EXITDEF statement) and was no longer being called by BPE is again called.

Important: If you changed the ABLIM parameter for a user exit in the PROCLIB member, the *new* value of ABLIM takes effect after the refresh command.

Considerations for refreshing user exits

- When you refresh a user exit type, BPE reloads all exit modules defined for that type. The new copies of the modules will be at a different virtual address than the old copies. Modules that are re-entrant will operate properly. However, if

your modules are not re-entrant and they store data within themselves, they must be able to tolerate being reloaded and losing the information previously stored within them.

Attention: Code and link edit all user exit modules as re-entrant to avoid this condition.

- If you refresh a previously loaded user exit module, BPE continues to pass the same static work area that was used by the previous copy of the module. If the new version of the module has a different mapping or use of this area than the previous version, the new version must contain toleration code that can handle the old-style formatted data within this static work area.

  Recommendation: Place a version number in the static work area, so that your exits can recognize when they are using a different data structure within this work area.
- If you remove a user exit module from an EXITDEF list and refresh the exits, BPE deletes the static work area associated with the removed exit module. If you later add the module back to the EXITDEF list and refresh the exits, the module gets a new (cleared) static work area, *not* the work area it had previously.
- If your user exits are being managed by link-lookaside (LLA) using virtual lookaside facility (VLF) or an equivalent product, you must ensure that the copies of the modules being refreshed are updated in LLA prior to issuing the REFRESH USEREXIT command.
- If you have user exits that issue z/OS WAITs for long periods of time (for example, a WAIT for an external event that may be delayed, such as a write to operator with reply (WTOR)), then issuing a REFRESH USEREXIT command could cause a performance problem or work stoppage. This is because BPE has to quiesce the user exits in order to process the REFRESH command. BPE must wait until all currently-called user exits complete before it can perform the user exit refresh. BPE prevents any new calls to user exits until after the command completes. If a user exit has been called and does not return to BPE for a long period of time, the REFRESH command is delayed until the exit returns. No other user exits can be called while BPE is waiting, so the processes that are invoking the user exits are also put into a wait state.

  Recommendation: Ensure that your user exits avoid long WAITs, and avoid issuing services that might WAIT.

Output

The `REFRESH USEREXIT` command output consists of message, BPE0032I indicating that the command has completed:

```
BPE0032I REFRESH USEREXIT COMMAND COMPLETED
```

Examples

Here are some examples of using the REFRESH USEREXIT command.

Command example 1

Refresh all user exit modules.

Command:

```
F CQS1,REFRESH USEREXIT NAME(*)
```

Output:

```
BPE0032I REFRESH USEREXIT COMMAND COMPLETED
```

Command example 2

Refresh all user exit modules for the OM command input and output exit types.

Command:

```
F OM,REF USRX NAME(INPUT,OUTPUT)
```

Output:

```
BPE0032I REF USRX COMMAND COMPLETED
```

Related concepts:

z/OS: Changing LLA-managed libraries

Related reference:

Chapter 1, “Introduction to the BPE commands,” on page 3

Base Primitive Environment customization exit routines (Exit Routines)

BPE exit list members of the IMS PROCLIB data set (System Definition)

Part 2. Common Service Layer commands

Use the Common Service Layer commands to work with the CSL.

- Chapter 5, “CSL SHUTDOWN command,” on page 39
- Chapter 6, “STOP csljob commands,” on page 41

© Copyright IBM Corp. 1974, 2017

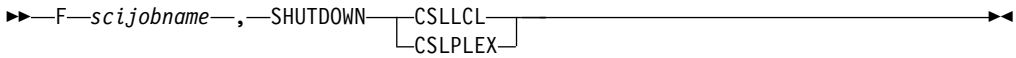
Chapter 5. CSL SHUTDOWN command

Use the CSL SHUTDOWN command to shut down a CSL on one z/OS image or to shut down an entire IMSplex.

Subsections:

- "Syntax"
- "Usage notes"

Syntax



Usage notes

Before issuing the z/OS MODIFY command, you may want to bring down your IMS control regions that are connected to the CSL by issuing a /CHE FREEZE command.

To shut down a CSL on one z/OS image, issue z/OS MODIFY command: F scijobname,SHUTDOWN CSLLCL where *scijobname* is the name of the SCI in the CSL.

This command shuts down the CSL on the z/OS image associated with the SCI that receives the command. Use this version of the command to shut down the CSL on a single z/OS image in an orderly way.

To shut down an entire IMSplex, issue z/OS MODIFY command: F scijobname,SHUTDOWN CSLPLEX, where *scijobname* is the name of the SCI in the CSL.

This command shuts down the CSL managers on all z/OS images in a single IMSplex associated with the SCI that receives the command.

Related tasks:

Shutting down the CSL using z/OS commands (Operations and Automation)

© Copyright IBM Corp. 1974, 2017

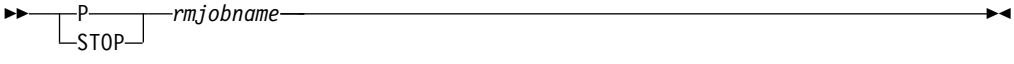
Chapter 6. STOP csljob commands

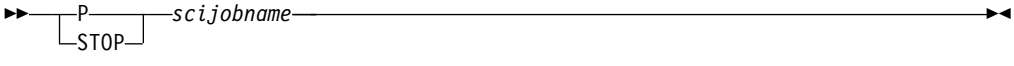
The STOP (or P) *odbmjobname*, *omjobname*, *rmjobname*, and *scijobname* commands are used to shut down the Open Database Manager (ODBM), Operations Manager (OM), Resource Manager (RM), and Structured Call Interface (SCI) address spaces in an IMSplex.

Subsections:

- "Syntax"
- "Keywords"

Syntax





Keywords

odbmjobname
Identifies the job name of the ODBM address space to be stopped.

omjobname
Identifies the job name of the OM address space to be stopped.

rmjobname
Identifies the job name of the RM address space to be stopped.

scijobname
Identifies the job name of the SCI address space to be stopped.

Related tasks:

Shutting down the CSL (Operations and Automation)

© Copyright IBM Corp. 1974, 2017

Part 3. Common Queue Server command

Use the Common Queue Server command to work with the CQS.

- Chapter 7, “STOP cqsjob command,” on page 45

© Copyright IBM Corp. 1974, 2017

Chapter 7. STOP cqsjob command

Use the STOP (P) *cqsjobname* command to shut down CQS normally. The command is rejected if any active IMS subsystems are currently connected to CQS.

Syntax

Keyword

The following keyword can be specified for the STOP *cqsjobname* command.

cqsjobname
Identifies the job name for the CQS to be stopped.

© Copyright IBM Corp. 1974, 2017

Part 4. DBRC commands

Use Database Recovery Control (DBRC) commands to obtain services from DBRC.

Database Recovery Control command information includes:

- Chapter 8, “Introduction to the DBRC commands,” on page 49
- Chapter 9, “BACKUP.RECON command,” on page 61
- Chapter 10, “CHANGE commands,” on page 63
- Chapter 11, “CLEANUP.RECON command,” on page 147
- Chapter 12, “DELETE commands,” on page 161
- Chapter 13, “GENJCL commands,” on page 177
- Chapter 14, “INIT commands,” on page 283
- Chapter 15, “LIST commands,” on page 321
- Chapter 16, “NOTIFY commands,” on page 345
- Chapter 18, “Sample listings from the RECON data set,” on page 393

© Copyright IBM Corp. 1974, 2017

Chapter 8. Introduction to the DBRC commands

Use DBRC commands to obtain services from DBRC. Use these commands to add, change, and delete information in the RECON data set and to generate the JCL and the control statements you need to run the various IMS utilities used in database recovery.

You can also use DBRC API requests to obtain the same services from DBRC.

DBRC batch commands

Use DBRC batch commands to create a backup copy of the RECON data set, check information in the RECON data set, modify information in the RECON data set, delete information from the RECON data set, generate jobs for the various IMS recovery utilities, or to make changes to a RECON data set.

Here is a summary of the DBRC commands:

- Use the BACKUP.RECON command to create a backup copy of the RECON data set.
- Use the CHANGE commands to modify information in the RECON data set.
- Use the DELETE commands to delete information from the RECON data set.
- Use the GENJCL commands to generate jobs for the various IMS recovery utilities.
- Use the INIT commands to make the following changes to the RECON data set:
  - Initialize the RECON data set
  - Register databases and their related records
  - Create change accumulation groups and reuse change accumulation data sets
  - Create DBDS groups, DB groups, and recovery groups
  - Create reuse image copy records
- Use the LIST commands to produce a formatted printout of all or selected parts of the RECON data set.
- Use the NOTIFY commands to add to the RECON information that is normally written there automatically.
- Use the REPAIR.RECON command to make the following changes to the RECON data set:
  - Rebuild a missing or corrupted RECON DMB Table record.
  - Check or correct inconsistencies with the internal representation of the data management block (DMB) number in the RECON header, database, and Fast Path area records.

DBRC commands can be issued using one of the following ways:

- You can issue DBRC commands using the Database Recovery Control utility (DSPURX00).
- You can also issue a variation of some of these commands online using the /RM*xxxxxx* command.
- With the exception of LIST.xx, you can issue DBRC commands using the DBRC application programming interface with the FUNC=COMMAND request.

© Copyright IBM Corp. 1974, 2017

- CICS® users can execute DBRC commands using the CICS-supplied transaction CDBM, that provides a command interface to DBCTL.

Related concepts:

Data set naming conventions (System Administration)

CICS: CDBM - database control interface

DBRC API (System Programming APIs)

Related reference:

Database Recovery Control utility (DSPURX00) (System Utilities)

CICS: Issuing DBRC commands from CICS

DBRC online commands

Certain DBRC commands can be issued from an online IMS environment (/RM*xxxxxx* commands).

The online DBRC commands are:
- /RMCHANGE
- /RMDELETE
- /RMGENJCL
- /RMINIT
- /RMLIST
- /RMNOTIFY

After these commands successfully pass through IMS security, IMS passes these commands to DBRC where they are processed. These commands are processed in a similar manner as the regular DBRC commands of similar name. For example, DBRC processes a /RMCHANGE command in a similar manner as a CHANGE command. The authorization of these online commands can also be controlled in a similar manner as the batch DBRC commands, namely you can use the DBRC Command Authorization exit (DSPDCAX0), a security authorization program like RACF®, or both.

Related concepts:

DBRC security (System Administration)

Related reference:

DBRC Command Authorization exit routine (DSPDCAX0) (Exit Routines)

DBRC command syntax

All DBRC commands adhere to the syntax described here. This syntax is standard, command-language syntax and is similar to that of TSO and Access Method Services.

You can enter commands in uppercase, lowercase, or mixed case format. DBRC translates most command input into uppercase format before processing, regardless of the format that is used. However, DBRC does not translate keyword values and string values into uppercase format. For example, the '*value*' portion of the **USERKEYS** parameter on GENJCL commands, and the **UDATA** parameter on the NOTIFY.UIC and CHANGE.UIC commands are processed in the exact format in which they are entered.

Subsections:

- "Separators"
- "Continuation characters"
- "Comments"
- "Commands"
- "Parameters"

Separators

A blank, a comma, or a comment can be interchanged in a command wherever a separator is needed. More than one separator can be used between parameters.

Continuation characters

Continuation characters are used to continue commands and comments that do not fit on a single line of input.

The two continuation characters used by DBRC are the minus sign (-) and the plus sign (+):

+ Deletes the leading separators from the continued line.

- Does not delete the leading separators from the continued line.

Restriction: A character string enclosed in single quotation marks (for example, ('c...c') can be continued only with a minus continuation character because separators are meaningful in a character string between single quotation marks. DBRC assumes such a character string to be ended if the end of a line is reached before the ending quotation mark is encountered and the last non-blank character in the line is not a minus continuation character. The maximum length of a character string is 256 characters.

Comments

Comments consist of alphanumeric character strings beginning with the symbols (/*) and ending with the symbols (*/).

A comment is assumed to have ended if the end of a line is reached before the character string (*/) is encountered and if the last character in the line is not a continuation character.

Commands

A command consists of a verb, a modifier, and, in most cases, a list of parameters. Exactly one period (.) follows the verb, with no other characters between the verb and the modifier.

Commands can be entered anywhere in columns 1 through 72 of the DBRC SYSIN input stream. Commands can be continued on multiple lines by entering a continuation character as the last non-blank character of the command line.

Columns 73 through 80 of the SYSIN input stream are ignored.

Parameters

Most DBRC commands have both required and optional parameters.

In general, there are no positional keyword parameters in DBRC commands. The keyword parameters are of the following types:

- A keyword by itself
- A keyword with a value:

  ```
  keyword(v)
  ```
- A keyword with a list of values:

  ```
  keyword(v1,v2..)
  ```
- A keyword with a repeating list of values:

  ```
  keyword((v1,v2..),(v1,v2..)..)
  ```

  When you enter a repeating list of values only once for this type of keyword, you can omit the outer set of parentheses like this:

  ```
  keyword(v1,v2)
  ```

Some commands have mutually exclusive parameters. In the parameter descriptions, a vertical bar (|) separates the mutually exclusive parameters. If you specify more that one of these parameters on the same command, you will not get an error, but DBRC will use the last one specified and ignore the others.

Important: The following coding rules apply to DBRC command parameters:

- Certain keywords require values in a specific form.
- The format for hexadecimal input is X'xxx', where X'x' can be any of the characters 0-9 and A-F.
- Any character can be part of a character string.
- A character string that contains blanks or any other special characters must be enclosed in single quotation marks. For example,

  ```
  UDATA('RECOVERY POINT FOR ALL PARTITIONS TAKEN ON IMS12 EACH MONTH')
  ```
- Data set names, data set ddnames, and volume serial numbers can include hyphens.
- Data set names follow the conventions specified in the z/OS JCL manuals.

Except where otherwise noted, optional keywords with values have the following defaults.

Numeric values
: 0

Character values
: blank

If a particular parameter is encountered more than once within a command, the last occurrence of the parameter is used. If mutually exclusive parameters are encountered within a command, the last occurrence is used.

DBRC time stamps

Time stamps are points in time recorded in the RECON data set. Correctly interpreting time stamp formats found in RECON listings, messages, and dumps will help you enter the appropriate time stamp in commands.

Time-stamp precision is to the microsecond.

Standard time stamp format

Certain parameters require a time stamp, which may be entered in one of the following formats:

Compressed:
yydddhhmmssthmiju [offset]

Punctuated:
[yy]yy | ddd | hh | mm | ss | thmiju [offset]

where:

yyyy Is the year (0000 to 9999)

ddd Is the day (000 to 366)

hh Is the hour (0 to 23)

mm Is the minute (0 to 59)

ss Is the second (0 to 59)

thmiju
Is the millionth of a second (000000 - 999999)

| Can be any non-numeric character delimiter including blank except the single quotation mark. If the time stamp contains any blanks, commas, or parentheses, it must be enclosed in single quotation marks. For example,

```
LIST.LOG STARTIME('11.213 08:24:45.123456 -8')
```

or

```
LIST.LOG STARTIME('11,213 08:24:45.123456 PST')
```

offset Can be one of the following:

1. Omitted. The current TIMEZIN value is used.
2. A numeric offset in the form **±h[h[:mm]]** or **±h[h[mm]]** that, when added to UTC, gives local time. **h[h]** is a numeric value from 0 to 14. For the compressed format if **mm** is specified, then **hh** must also be specified. **mm** is a value from the set {00, 15, 30, 45}.

   ±hh:mm is valid only between the values -11:45 to +14:45.

   ±hhmm is valid only between the values -1145 to +1445.
3. A symbolic time zone label.

The time stamp value might have elements truncated on the right, in which case the omitted element's digits are assumed to be zeros.

You can truncate the input at the beginning of any element after `ddd`; so, `yyyy|ddd` is acceptable, as is `yyyy|ddd|hh`. Part of an element cannot be entered: for example, `yyyy|ddd|h` is invalid.

If only two digits are entered for the year, the two high order digits are extrapolated by using the sliding-window method described in “Extrapolation of two-digit year input” on page 57.

The same time stamp value could be entered in the following ways:

```
11213082445123456
11213082445123456-0800
11.213/08:24:45.123456
```

or with blank, commas, or parentheses:

```
'11.213 08:24:45.123456 -8'          '11/213-08.24.45.123456 -8:00'
'11,213 08:24:45.123456 PST'         '11/213-16.24.45.123456 UTC'
'2011 213 16.24.45.123456 +0'
```

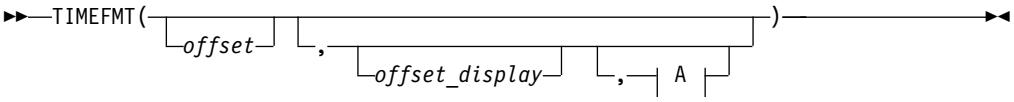
TIMEFMT parameter

Use the optional TIMEFMT parameter to define the form in which time stamps appear in messages, displays, and listings from DBRC. Each parameter associated with TIMEFMT is optional and can be omitted by including only the comma. Time stamps in command output and message output are displayed using the format specified by the TIMEFMT parameter. All formats will now be displayed up to the microsecond precision.

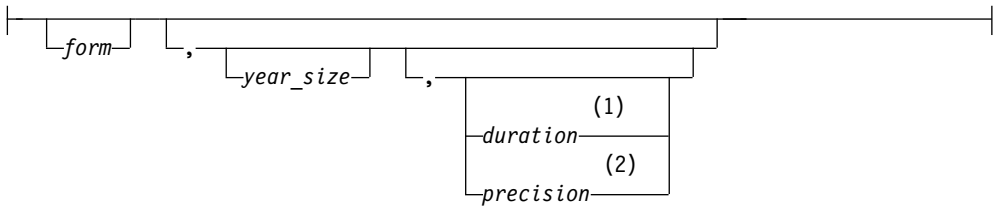
The TIMEFMT parameter can be used on `CHANGE.RECON`, `LIST.xxxx`, and `GENJCL.xxxx` commands or can be specified in a skeletal JCL member.

Restriction: The parameters that are associated with TIMEFMT are positional.

The format of the TIMEFMT parameter sublist is as follows:



A:



Notes:

1 *duration* can be set only on the `CHANGE.RECON` command.

2 precision is only valid on the %SET statement.

If items of the sublist are omitted, the current values from the RECON header are used.

offset
Specifies the offset that is applied to the UTC internal time before display.

U Specifies none. Displays the UTC when the event occurred.

O Specifies origin. Displays the local time when and where the event occurred.

L Specifies current local offset. Displays the current local time equivalent.

offset_display
Specifies the display format of the offset that is appended to the time

L Specifies that the offset is displayed in label format, if a label has been defined for it. If no label is defined, the offset is displayed in numeric format.

O Specifies that the offset is displayed in the numeric (+|- HH:MM) format.

N Specifies that no time zone information is displayed.

form
Specifies whether the time stamp is displayed in punctuated or compressed form.

P Specifies that the time stamp is displayed in punctuated form.

C Specifies that the time is displayed in compressed form.

year_size
Specifies whether all four digits of the year are displayed or only the two low-order digits.

2 Only the units and tens digits of the year are displayed.

4 All four digits of the year are displayed.

duration
Specifies the scope of these choices to be either limited to the current job or used as global overrides to the system defaults. The `duration` subparameter can only be specified on a `CHANGE.RECON` command.

PERM
Indicates that the specified options are in effect for any subsequent DBRC utility job running with the same RECON; that is, these values become the defaults for subsequent jobs.

TEMP
Indicates that the specified options are in effect only for the job in which the command is entered. If neither **PERM** nor **TEMP** is coded, **TEMP** is the default.

precision
Coded only on a %SET statement in skeletal JCL as a number from 1-6. You can use it to control the number of low-order digits contained in time stamps that are output by GENJCL. The default is 6.

As with other parameters, coding a null value causes the corresponding TIMEFMT value to be reset to the GENJCL default. TIMEFMT() resets all values.

TIMEFMT parameter in skeletal JCL

Besides its use on the `CHANGE.RECON` command, the TIMEFMT parameter can be coded on any `LIST.xxx` or `GENJCL.xxx` DBRC command. It can also be specified in a skeletal JCL member as:

```
%SET TIMEFMT(....)
```

Here is an example of the %SET keyword in skeletal JCL:

```
%SET TIMEFMT(,N)
%SELECT RLDS(%SSID,LAST)
LOGEND =%LOGETIM
%ENDSEL
```

And here is what the output from the preceding example of %SET would render:

```
LOGEND =11011131500023456
```

TIMEFMT subparameter order of precedence

The order of precedence of the TIMEFMT subparameters, from highest to lowest, is as follows:

1. RECON defaults

   The RECON defaults are established by the INIT.RECON command. These defaults can be reset using the PERM option of the CHANGE.RECON command.

2. Job-level override

   The RECON defaults can be overridden for the following commands in a SYSIN command stream using either PERM or TEMP options of the CHANGE.RECON command.

3. %SET statements in skeletal JCL members

   The TIMEFMT settings in a %SET statement override the GENJCL default settings, as well as settings from a previous %SET statement in the same member.

4. Command override

   The TIMEFMT settings in a DBRC command override all of the above settings within the scope of that command invocation. For example, a GENJCL.xxx command TIMEFMT setting overrides the TIMEFMT settings on any %SET statement in the applicable skeletal JCL member.

When coded on any command other than CHANGE.RECON, the last parameter, duration, has no meaning; any TIMEFMT values coded override the values that are currently in effect only for the duration of the command. For any omitted values, the values currently in effect from the most recent CHANGE.RECON command with a PERM duration, which override the INIT.RECON defaults, remain in effect.

The values set in the RECON data set by the INIT.RECON command are **TIMEFMT(O,N,P,2,TEMP)**.

The defaults used by GENJCL commands are **TIMEFMT(O,O,C,2,TEMP,6)**.

Recommendation: If your installation operates multiple IMS systems with different time zones and they share data and a RECON; use UTC or LOC for RECON listings, at least, so that all the time stamps listed have a common base.

Specifying zero time stamp values

A zero time stamp is valid only where explicitly allowed for a command keyword value. A zero time stamp value, where permitted, can be entered in either punctuated or compressed format, or can be condensed into a single digit. For example, you can specify a zero time stamp in the following ways:

```
0000000000000000
'00 000 00:00:00.000000'
0
```

If an offset value is given, it is ignored.

Extrapolation of two-digit year input

Input time stamps on DBRC commands are accepted with the two-digit year notation. In this case the century digits are derived for the use of the internal time stamp. The century digits in the internal time stamp are determined from the two entered digits by the following process:

- The entered year digits are compared to a sliding window of 100 years. The lower limit of the window is the current year minus 70. The upper limit is the current year plus 29. A *breakpoint* is defined as the two low-order digits of the lower limit.

  If the entered digits are equal to, or greater than, the breakpoint, the century digits of the lower limit are used. Otherwise, the century digits of the upper limit are used. The upper and lower limits of the window and the breakpoint are established when IMS is initialized. When the year changes, these values are computed again.

Time stamp conversions and examples

In the examples that follow, local time is PST (offset -08:00).

An event occurring at 2007.030 10:00:0.123456 -08:00 (punctuated format) or 0703010000123456 -0800 (compressed format) is recorded in 12 bytes in the RECON data set as `2007030F|18000012|3456032D|` (*yyyydddFhhmmssthmijufqqs*), where:

yyyydddFhhmmssthmiju
: UTC, 8 hours later than the local time

fqqs
: Offset:

  f
  : Flag bits, normally 0

  qq
  : Quarter hours (32/4=8)

  s
  : Sign (D is negative, C is positive)

Note: The offset is always added to the UTC time to obtain the local time. To obtain the UTC time from local time, reverse the sign of the offset and add it to the local time.

The time stamp from a record in the RECON data set, such as a PRILOG record, lists as:

```
'2007.030 10:00:00.123456 -08:00'
```

When the time stamp above is entered in a command with the quotation marks, DBRC finds the record in the RECON data set with the recorded time as:

```
2007030F 18000012 3456032D
```

DBRC uses UTC time internally when searching for records in the RECON data set. The offset portion of the UTC time is ignored.

If no offset is supplied in the time stamp, DBRC uses the TIMEZIN value which might not be correct if a clock change has occurred since the time when the record was written in the RECON data set.

As long as the offsets are included, time stamps from different RECON listings can be used, even if the time stamps are displayed in different formats.

Standard default settings for time-stamp values

The RECON data set time-stamp formats are the standard default.

From the RECON header:

```
TIMEZIN=%SYS
```

OUTPUT FORMAT: DEFAULT = LOCORG NONE PUNC YY No user-defined labels are defined.

Listing an allocation record without offset values lists the time stamp as:

```
11.027 19:03:47.123456
```

The allocation record was written prior to daylight savings change. The time stamp is entered on a command to list the record without an offset. With TIMEZIN=%SYS set in the RECON data set, the offset is obtained from current z/OS clock which has changed. The record is not found in the RECON data set.

DBRC commands affected by the time-stamp format

These DBRC commands have time stamps as part of their syntax.

- CHANGE.BKOUT
- CHANGE.CA
- CHANGE.IC
- CHANGE.PRILOG
- CHANGE.SECLOG
- CHANGE.UIC
- DELETE.ALLOC
- DELETE.CA
- DELETE.IC
- DELETE.LOG
- DELETE.RECOV
- DELETE.REORG
- DELETE.UIC
- GENJCL.CA
- GENJCL.RECOV
- GENJCL.USER
- LIST.HISTORY
- LIST.LOG
- NOTIFY.ALLOC
- NOTIFY.BKOUT
- NOTIFY.CA
- NOTIFY.IC
- NOTIFY.PRILOG
- NOTIFY.RECOV
- NOTIFY.REORG
- NOTIFY.SECLOG
- NOTIFY.UIC

For more information on the standard time-stamp format, see “Standard time stamp format” on page 53.

DBRC commands that require full-precision time stamp parameters

The following commands use time stamps to identify specific RECON records or information within RECON records that is being changed. A full-precision time stamp is required in these cases.

- CHANGE.BKOUT: UORTIME parameter
- CHANGE.CA: RECTIME parameter
- CHANGE.IC: RECTIME parameter
- CHANGE.PRILOG (for RLDS/SLDS): STARTIME and DSSTART parameters
- CHANGE.SECLOG (for RLDS/SLDS): STARTIME and DSSTART parameters
- CHANGE.UIC: RECTIME parameter
- DELETE.ALLOC: RECTIME parameter
- DELETE.CA: RECTIME parameter
- DELETE.IC: RECTIME parameter
- DELETE.LOG: STARTIME parameter
- DELETE.RECOV: RECTIME parameter
- DELETE.REORG: RECTIME parameter
- DELETE.UIC: RECTIME parameter
- LIST.LOG: STARTIME parameter
- NOTIFY.ALLOC: STARTIME parameter
- NOTIFY.ALLOC: ALLTIME parameter (only when specified with the DEALTIME parameter).
- NOTIFY.PRILOG (for RLDS/SLDS): STARTIME parameter
- NOTIFY.PRILOG (for RLDS/SLDS): STARTIME parameter

DBRC no longer ignores the low-order digits following the tenth of a second. For example, if you want to modify information for the following image copy:

```
-------------------------------------------------------------------------------

IMAGE
  RUN     = 2011.114 16:28:41.123456 -07:00*  RECORD COUNT =0
  STOP    = 0000.000 00:00:00.000000 +00:00   BATCH      USID=0000000000

 IC1
  DSN=DHVNTZ02.IC1                                  FILE SEQ=0001
  UNIT=3400                              VOLS DEF=0001 VOLS USED=0001
                                         VOLSER=USER01

-------------------------------------------------------------------------------
```

Enter one of the following CHANGE.IC commands:

```
CHANGE.IC  DBD(DHVNTZ02) DDN(HIDAM) RECTIME('2011.114 16:28:41.123456 -07:00') -
ICDSN(NEWDSN)
CHANGE.IC  DBD(DHVNTZ02) DDN(HIDAM) RECTIME(11114162841123456-0700) -
ICDSN(NEWDSN)
CHANGE.IC  DBD(DHVNTZ02) DDN(HIDAM) RECTIME(11114162841123456) -
ICDSN(NEWDSN)
```

Creation of new records involving time stamps (for example, NOTIFY.IC) or displaying log records in a time range (for example, LIST.LOG FROMTIME) enable the user to be as precise as they choose when specifying the time stamp. With the exception of the parameters listed above, all other DBRC command time stamp parameters only require the minimum specification of year and day. The rest of the time stamp will be padded with zeros.

Related reference:

"CHANGE.RECON command" on page 118

"Skeletal JCL syntax" on page 225

Chapter 9. BACKUP.RECON command

Use the BACKUP.RECON command to create backup copies of the RECON data set from the Copy1 RECON.

The BACKUP.RECON command first opens the RECON data set and any needed cleanup is done to remove any RECON inconsistencies. The command then invokes the IDCAMS REPRO command, using its normal defaults, to create the backup copy. Any restrictions applicable to the normal use of the REPRO command apply to this command. The data set receiving the backup copy must be empty.

Subsections:

- "Syntax"
- "Keywords"
- "Usage notes"
- "Example of creating backups of a RECON data set" on page 62

Syntax

Keywords

The following keywords are valid for the BACKUP.RECON command:

RECON1 | RECON2 | BOTH
Mutually exclusive, optional keywords that you use to specify the backup data set to which the RECON data set is copied.

RECON1
Copies the RECON data set to the backup data set specified by the BACKUP1 DD statement of your JCL.

RECON2
Copies the RECON data set to the backup data set specified by the BACKUP2 DD statement of your JCL.

BOTH
Copies the RECON data set to the data sets specified by the BACKUP1 and BACKUP2 DD statements of your JCL.

Usage notes

If parallel RECON access is used, RECON activity for all DBRC instances is quiesced. A check for the existence of retained locks related to the COPY1 RECON data set is completed before the IDCAMS REPRO command is invoked. If any retained locks exist, the IDCAMS REPRO command is not invoked, and the command fails.

© Copyright IBM Corp. 1974, 2017

In parallel RECON access mode, any retained locks that exist for the RECON data sets must be resolved. Ensure that the SMS server (SMSVSAM) is active and Transactional VSAM (TVS) is active on all processors and resubmit the command.

If your RECON RECORDSIZE is greater than 32 KB, the BACKUP.RECON command will fail if the back up is to a sequential data set. Perform the following steps to back up a RECON data set to a sequential file (for example, to a tape):

1. Create a backup KSDS using the BACKUP.RECON command
2. Use DFSMSdss to copy the backed-up KSDS to the sequential file

Example of creating backups of a RECON data set

In this example, two backup copies of the Copy1 RECON are created.

```
//BKUP     JOB
//BACKUP1  DD   . . .
//BACKUP2  DD   . . .
⋮
//SYSIN    DD   *
     BACKUP.RECON BOTH
/*
```

To create a sequential backup, the BACKUPx DD statement must include appropriate DCB parameters. The BLKSIZE specified must be larger than the maximum RECORDSIZE defined in the RECON data set, but less than 32K. For example, DCB=(RECFM=VB,LRECL=32756,BLKSIZE=32760).

Chapter 10. CHANGE commands

Use the CHANGE commands to modify information in a RECON data set.

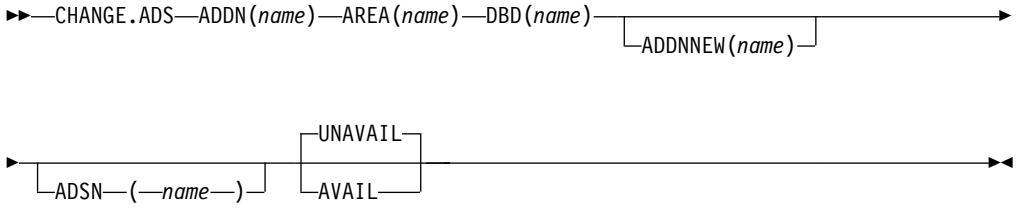
CHANGE.ADS command

Use the CHANGE.ADS command to change DEDB ADS information in the RECON data set. The CHANGE.ADS command fails if you issue it while the area is in use.

Subsections:

- "Syntax"
- "Keywords"
- "Example of changing an ADS record" on page 64

Syntax



Keywords

ADDN(*name***)**
Required keyword you use to specify the area data set ddname of the ADS being changed.

AREA(*name***)**
Required keyword you use to identify the ADS being changed, by area name.

DBD(*name***)**
Required keyword you use to identify the ADS being changed, by database name

ADDNNEW(*name***)**
Optional keyword you use to identify the ADS being changed, by new ddname.

ADSN(*name***)**
Optional keyword you use to identify the ADS being changed, by new data set name.

UNAVAIL | AVAIL
Mutually exclusive, optional keywords you use to change the ADS record to indicate its availability.

UNAVAIL
Indicates that the ADS is unavailable.

© Copyright IBM Corp. 1974, 2017

AVAIL
Indicates that the ADS is available. The CHANGE.ADS AVAIL command fails if the area needs to be recovered.

If neither AVAIL nor UNAVAIL is specified but ADSN is specified, the value defaults to UNAVAIL.

Example of changing an ADS record

In this example, an ADS record in the RECON data set is being changed.

```
//CHGADS   JOB
.
.
.
//SYSIN    DD   *
     CHANGE.ADS DBD(DBD001) AREA(AREA002)  -
                ADSN(ADSN004) ADDN(ADDN004)
/*
```

Related reference:

"INIT.ADS command" on page 283

CHANGE.BKOUT command

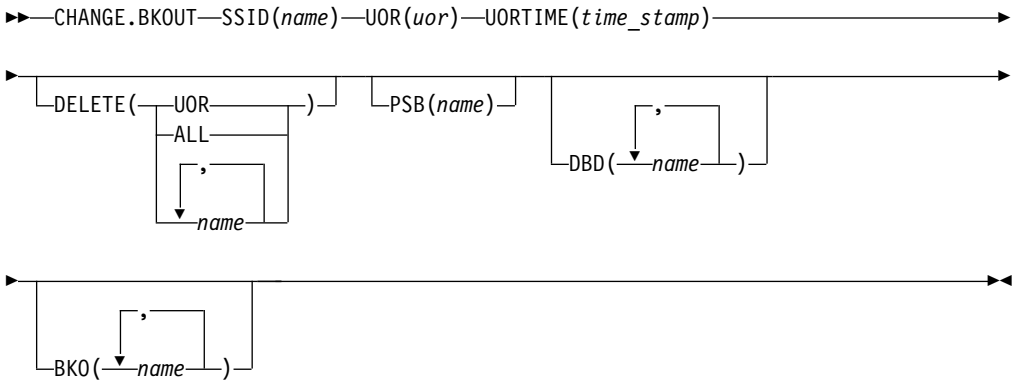
Use the CHANGE.BKOUT command to add, change, or delete a unit of recovery (UOR) in the backout record that is associated with a specified subsystem.

Recommendation: Use the CHANGE.BKOUT command when a dynamic backout failure has occurred and certain known backout records are invalid. Using the CHANGE.BKOUT command incorrectly can result in a loss of recovery integrity.

Subsections:

- "Syntax"
- "Keywords"
- "Example of using the CHANGE.BKOUT command" on page 66

Syntax



Keywords

The following keywords are valid for the CHANGE.BKOUT command:

SSID(*name***)**
Required keyword that specifies the subsystem for which the backout record is

to be changed. The name is an alphanumeric string, with up to eight characters, that represents any valid subsystem name.

UOR(*uor***)**
Required keyword that you use in conjunction with the UORTIME keyword to identify a unit of recovery in the backout record. The recovery token (*uor*) is 16-byte field that describes a specific UOR in the backout record. The value for this keyword must be expressed as 32 hexadecimal digits.

This keyword can specify a unit of recovery that currently exists in the backout record or one that is to be added to the record.

UORTIME(*time_stamp***)**
Required keyword that specifies the time of the UOR described above. The value is the beginning time of the UOR (found in the X'5607' log record). The time stamp must be in standard form.

DELETE(UOR | ALL | *name...***)**
Optional keyword used to delete some or all of the information related to the unit of recovery that is specified by the required keywords described above.

UOR
Deletes the entire UOR defined by the required UOR and UORTIME keywords described above. If you do not specify DELETE(UOR), CHANGE.BKOUT assumes you are changing an existing UOR or adding a UOR that is not currently in the backout record.

If you specify DELETE(UOR), all other optional keywords are ignored.

If the UOR does not exist in the backout record, the command fails.

ALL
Specifies that the database entries for the specified UOR and UORTIME are deleted, but that the UOR prefix information is left intact, if you also specify database names in the DBD keyword, the BKO keyword, or both. If you do not specify database names in DBD or BKO, CHANGE.BKOUT acts as if DELETE(UOR) is specified. You can use the ALL option to replace, or substantially alter, a database list within a UOR entry of a backout record without disturbing the control data in the UOR's prefix. You can also use the ALL option to delete all database entries in the UOR except those that are listed in the DBD or BKO keywords.

name
Specifies up to eight database names for use with the DELETE keyword. Use a comma to separate each specified name. If you list all of the databases associated with the specified unit of recovery, CHANGE.BKOUT acts as if DELETE(ALL) is specified.

If any listed database name is not in the specified UOR, the command fails.

Restrictions for HALDB databases: When using the CHANGE.BKOUT command for HALDB databases, *name* specifies the name of a HALDB partition.

The following optional keywords can only be used if you do not specify DELETE(UOR). If the UOR already exists in the backout record, you must provide at least one of the optional keywords. If the UOR does not exist in the backout record, it is added. In this case, you must specify the PSB and either the DBD keyword or the BKO keyword.

You can specify either the BKO keyword, the DBD keyword, or both. However, the same database name cannot appear in both the BKO and the DBD keywords, because a database cannot be both backed out and require a backout at the same time.

PSB(*name*)
Optional keyword that identifies the PSB associated with the UOR. To add a UOR to the backout record, you must specify PSB(*name*). If the UOR defined by the required keywords already exists in the backout record, the specified PSB name replaces the current PSB name.

DBD(*name*...)
Optional keyword that identifies databases associated with the specified UOR. Up to eight database names can be listed with the DBD keyword. The database names listed here identify the databases that require backout for this unit of recovery. This keyword can be used to change the status of an existing database entry to backout required.

Restrictions for HALDB databases: When using the CHANGE.BKOUT command for HALDB databases, *name* specifies the name of a HALDB partition.

BKO(*name*...)
Optional keyword that identifies databases to which the UOR applies. Use BKO to identify databases that have already been backed out from this UOR. Up to eight database names can be specified with the BKO keyword. This keyword can be used to change the status of an existing database entry to backout completed.

Restrictions for HALDB databases: When using the CHANGE.BKOUT command for HALDB databases, *name* specifies the name of a HALDB partition.

Example of using the CHANGE.BKOUT command

In this example, all database entries for the specified UOR and UORTIME are deleted, except those that are listed in the DBD and BKO keywords.

```
//CHGBKOUT JOB
.
.
.
//SYSIN    DD  *
     CHANGE.BKOUT  SSID(SYS3)-
                   UOR(E2E8E2F340404000000000600000003)-
                   UORTIME(07093134502123456) DELETE(ALL)-
                   DBD(DATA1,DATA2,DATA3C)-
                   BKO(DATA4,DATA5,DATA3A)
/*
```

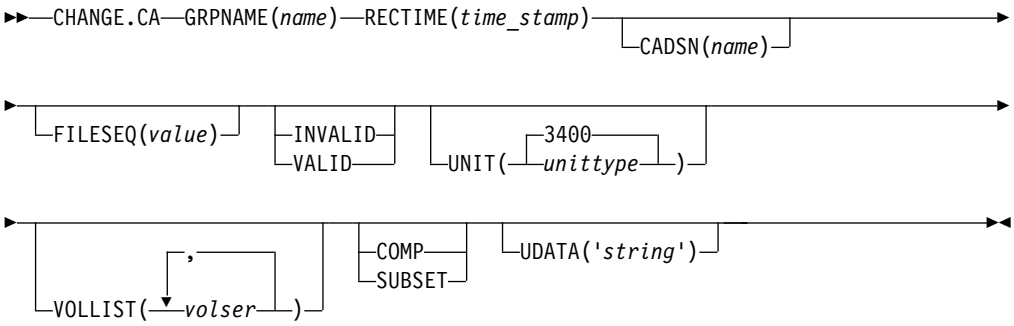
CHANGE.CA command

Use the CHANGE.CA command to change information about a specified run of the Change Accumulation (CA) utility for an identified CA group in the RECON data set.

Subsections:

- "Syntax" on page 67
- "Keywords" on page 67
- "Examples" on page 68

Syntax



Keywords

GRPNAME(*name*)
Required keyword you use to specify the name of the CA group for which information is to be changed.

RECTIME(*time_stamp*)
Required keyword you use to identify the change accumulation run record that you are changing.

Use the STOP time marked with an asterisk (*) from the listing of the CA record. The time stamp must be in standard form.

CADSN(*name*)
Optional keyword you use to specify the new name of the change accumulation data set in the identified record.

FILESEQ(*value*)
Optional keyword you use to specify a new file sequence number that is to be recorded in the identified record.

INVALID | VALID
Mutually exclusive, optional keywords you use to specify whether a change accumulation data set is to be used as input for a subsequent run of change accumulation or database recovery.

INVALID
Specifies that the change accumulation data set is not to be used as input for a subsequent run of change accumulation or database recovery. If an invalidated change accumulation data set is subsequently reused for output by change accumulation, it is automatically marked as valid and is used.

VALID
Specifies that the previously invalidated change accumulation data set is available for use as input to a subsequent run of change accumulation or database recovery. Use this keyword only if the change accumulation data set was previously marked as invalid and is now valid.

Specifying the INVALID keyword causes the STOPTIME and RUNTIME of the change accumulation record to be swapped. This prevents duplicate records in the RECON data set. Specifying the VALID keyword causes the STOPTIME and RUNTIME to be swapped back.

UNIT(3400 | *unittype*)
Optional keyword you use to change the unit type of the volumes on which the change accumulation data set resides. The unit type can be up to eight alphanumeric characters long.

VOLLIST(*volser*)
Optional keyword that can be listed, that you use to replace the volume serial numbers of the change accumulation data set in the specified change accumulation run record. You can substitute up to 255 volume serial numbers in the variable field; each can be up to six alphanumeric characters long.

SUBSET | COMP
Mutually exclusive, optional keywords you use to indicate the change accumulation status.

SUBSET
Indicates that when the CA was created, a subset of logs were processed and the CA's stop time is the start time of the first unprocessed log volume.

COMP
Indicates that when the CA was created, a complete set of logs were processed and the CA's stop time is the stop time of the last log volume that was processed.

You do not need to use this keyword under normal conditions. Checking is not done to verify that the use of this keyword is consistent with the value of the CA stop time. This keyword value is used by the GENJCL.CA and GENJCL.RECOV processes. Incorrect use can result in invalid generated JCL.

UDATA('*string*')
Optional keyword you use to specify up to 80 bytes of information about the identified change accumulation data set. You can use the variable field of this keyword to describe how the change accumulation data set was created. The *string* value must be enclosed in single quotation marks if blanks or any other special characters are included.

Examples

Example 1: Changing a change accumulation run record

In this example, a change accumulation run record in the RECON data set is being changed. The INVALID keyword indicates that the identified data set is not to be used as input to a subsequent run of a utility. The VOLLIST, FILESEQ, and CADSN keywords indicate additional fields in the record that are to be changed.

```
//CHGCA    JOB
.
.
.
//SYSIN    DD   *
     CHANGE.CA  GRPNAME(CAGRP2)  RECTIME(07065020433523456)   -
       INVALID  CADSN(IMS.CAGRP2.CA.CADSN2)  -
       VOLLIST(VOLCA1) FILESEQ(4)
/*
```

Example 2: Changing the user data in a change accumulation run record

In this example, information in a change accumulation record that is identified by the DBD, DDN, and RECTIME keywords is to be changed in the RECON data set. The changed user data is included in the UDATA parameter.

```
//CHGCA    JOB
⋮
//SYSIN    DD   *
  CHANGE.CA   GRPNAME(CAGRP2)  -
    RECTIME(09152104355891200) —
    UDATA('LAST CA OF THE WEEK')
/*
```

Related reference:

"INIT.CAGRP command" on page 287

"DBRC time stamps" on page 52

CHANGE.CAGRP command

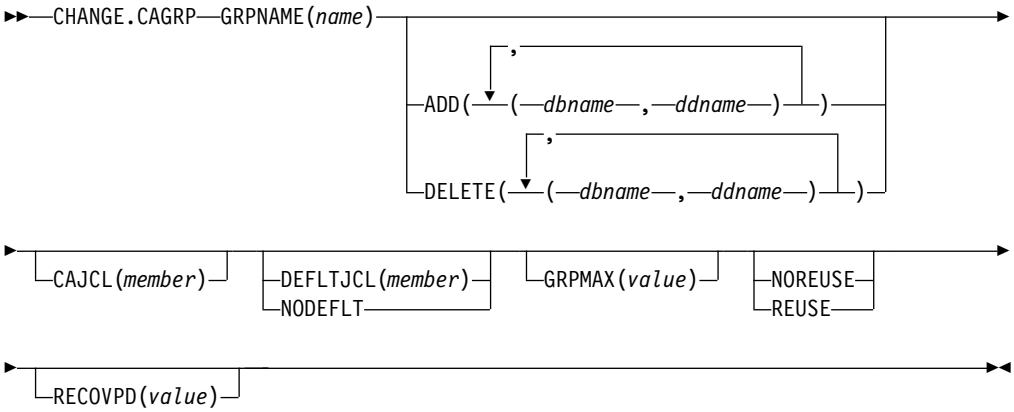
Use the CHANGE.CAGRP command to modify information contained in a specified CA group record in the RECON data set. You can also use this command to change the names of DBDSs that are members of a CA group.

Restriction: HALDB, ILDS, and PHIDAM Primary Index DBDSs are not recoverable and changes to them are not logged. The CHANGE.CAGRP command does not support these data sets.

Subsections:

- "Syntax"
- "Keywords"
- "Usage notes" on page 71
- "Examples of using the CHANGE.CAGRP command" on page 71

Syntax



Keywords

GRPNAME(*name***)**
Required keyword you use to specify the name of the CA group whose record you want to modify.

ADD(*dbname***,***ddname***) | DELETE(***dbname***,***ddname***)**
Mutually exclusive, optional keywords you use to specify the member you want to add to or delete from the specified CA group.

ADD
Specifies members you are adding to the identified CA group. A group cannot have more than 32767 members.

Restriction: User-recoverable databases cannot be added to a CA group.

DELETE
Specifies members you are deleting from the identified CA group. When you have deleted a member from a CA group, DBRC does not know previous change accumulation activities on that DBDS.

Specify a list of one or more members in the variable field; each member is a pair of names enclosed in parentheses. *dbname* is the database name of the member. For HALDB databases, *dbname* specifies the name of a HALDB partition. *ddname* is the symbolic name of the DD statement.

If you delete all the members of a group, the record of that group is deleted from the RECON data set.

CAJCL(*member***)**
Optional keyword you use to change the name of a member of a partitioned data set of skeletal JCL. The member is used to generate the JCL for a run of the Change Accumulation utility when you issue a GENJCL.CA command for the specified CA group.

DEFLTJCL(*member***) | NODEFLT**
Mutually exclusive, optional keywords you use to specify the implicit skeletal JCL default member for the CA group.

DEFLTJCL
Specifies an implicit skeletal JCL default member for the CA group. GENJCL.CA uses the default member you specify in order to resolve keywords you have defined.

NODEFLT
Specifies that no skeletal JCL default member is to be used for the CA group.

GRPMAX(*value***)**
Optional keyword you use to modify the maximum number of change accumulation data sets that DBRC is to maintain for the specified CA group. *value* must be a decimal number from 2 to 1024.

When the number of times you run the Change Accumulation utility for the specified group exceeds the GRPMAX value, the record with the earliest change accumulation stop time beyond the recovery period for the group is deleted for CA groups defined with either REUSE or NOREUSE keywords.

NOREUSE | REUSE
Mutually exclusive, optional keywords you use to indicate whether change accumulation data sets are to be reused.

NOREUSE
Indicates that change accumulation data sets that were already used for the specified CA group cannot be reused as output data sets in subsequent runs of the Change Accumulation utility. Any existing, unused change accumulation run records for the specified CA group are deleted when you specify the NOREUSE keyword.

REUSE
Indicates that change accumulation data sets that were already used for the specified CA group can be reused as output data sets in subsequent runs of the Change Accumulation utility.

If GRPMAX is higher than the number of existing data sets for the group, use the INIT.CA command to add additional data sets; otherwise reuse does not take place. See the explanation for GRPMAX.

RECOVPD(0 | *value*)
Optional keyword you use to modify the recovery period for a specified CA group. The recovery period is the amount of time before the current date for which DBRC maintains change accumulation information in the RECON data set. For example, if the recovery period of a change accumulation group is 14 days, DBRC maintains sufficient change accumulation execution records for at least 14 days.

To determine whether the change accumulation execution record falls within the recovery period, subtract the RECOVPD value from the current time. Any change accumulation execution records with stop times that are newer than the calculated time is to be kept in the RECON data set.

For *value*, specify a decimal number from 0 to 999 that represent the number of days the change accumulation execution records are to be kept in the RECON data set. If you specify 0 (the default), there is no recovery period.

Usage notes

If you issue the CHANGE.CAGRP command and specify the GRPMAX and RECOVPD values that are less than the existing values, any used change accumulation data sets with stop times that are beyond the recovery period are deleted until the number of remaining change accumulation data sets equals the specified GRPMAX value.

If you issue the DELETE.CA command, any specified change accumulation data set record is deleted regardless of the RECOVPD or GRPMAX values.

Attention: If the GRPMAX value is lowered by using the CHANGE.CAGRP command, the GRPMAX value is recorded regardless of whether the oldest change accumulation data sets can be deleted, because they are within the recovery period.

Examples of using the CHANGE.CAGRP command

Here are several examples of things you can do using the CHANGE.CAGRP command.

Example of adding DBDSs to the existing CA group CAGRP1

In this example, the DBDSs identified by the *dbname* and *ddname* parameters are to be added to the existing CA group, CAGRP1.

```
//CHGCAGRP JOB
.
.
.
//SYSIN    DD   *
      CHANGE.CAGRP GRPNAME(CAGRP1) ADD((DB1,DD1),(DB2,DD2))
```

Example of deleting DBDSs from the CA group CAGRP1

In this example, the DBDSs identified by the *dbname* and *ddname* parameters are to be deleted from the CA group, CAGRP1.

```
//CHGCAGRP JOB
⋮
//SYSIN    DD   *
       CHANGE.CAGRP GRPNAME(CAGRP1) DELETE((DB3,DD3),(DB4,DD4))
```

Example of changing a CA group record

In this example, a CA group record in the RECON data set is being changed. The changes are made to prevent the reuse of CA data sets by the Change Accumulation utility. It also renames the member of the partitioned data set of skeletal JCL that is used for generating the JCL that is needed for the Change Accumulation utility for this CA group.

```
//CHGCAGRP JOB
⋮
//SYSIN    DD   *
     CHANGE.CAGRP  GRPNAME(CAGRP3)  NOREUSE  CAJCL(JCLCA)
/*
```

Example of changing a CA group record

In this example, the retention period for the change accumulation data sets for CA group CAGRP1 is being changed to 90 days.

```
//CHGCAGRP JOB
⋮
//SYSIN    DD   *
      CHANGE.CAGRP GRPNAME(CAGRP1) RECOVPD(90)
```

Related tasks:

Adjusting GRPMAX when it is reached or it is too high (System Administration)

Related reference:

"INIT.CAGRP command" on page 287

CHANGE.DB command

Use the CHANGE.DB command to change the information about a database or a Fast Path DEDB area. This information is contained in a database record or area record in the RECON data set.

If you specify the keywords SHARELVL, TYPEFP, or TYPEIMS while the database or an area of the DEDB is in use, the command fails.

You can also use the CHANGE.DB command to remove a rare authorization inconsistency between the subsystem (SSYS) record and the database or area (DB/AREA) record. This inconsistency occurs when either the SSYS record still has an entry for the DB/AREA in its authorized databases/areas list but the DB/AREA record no longer contains the SSID entry in its associated subsystem information list or the SSID entry is still in the DB/AREA but the SSYS record either no longer exists in the RECON data set or no longer contains an entry for the DB/AREA. Use the AUTH keyword for this purpose.

Restriction: When issuing a CHANGE.DB command during a HALDB online reorganization:

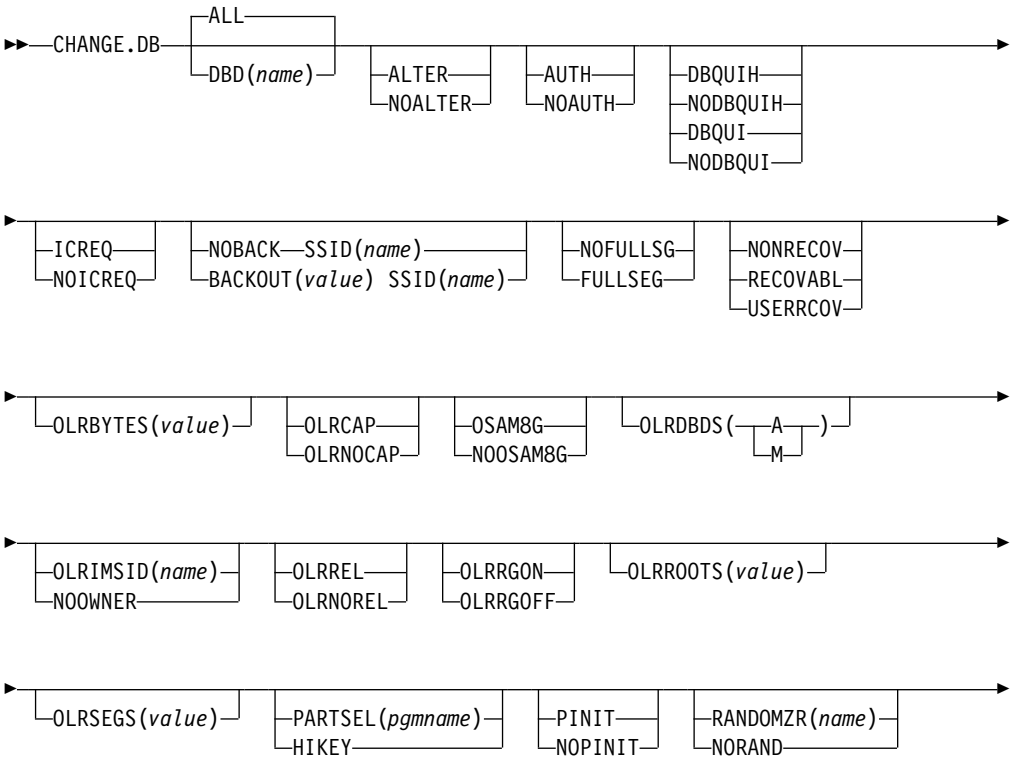
- The CHANGE.DB command can be specified only for a HALDB partition (TYPE=PART).
- HALDB online reorganization must have already been attempted for this partition (the M-through-V data sets must exist).
- The ALL keyword can be used only when issuing a CHANGE.DB ALL OLRCAP command. All HALDB databases that are available to be marked as online reorganization capable are marked "OLRCAP." Those unavailable for change (such as HALDB databases that are currently authorized) are skipped and are not marked as OLRCAP.

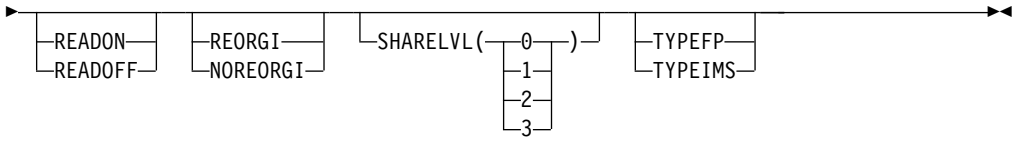
Subsections:

- "Syntax"
- "Keywords" on page 74
- "Usage notes" on page 84
- "Example of changing a record for a database identified with the DBD keyword" on page 84

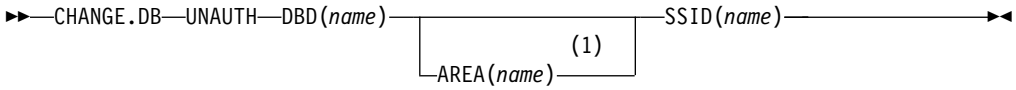
Syntax

Use one of the following two diagrams for the CHANGE.DB command.



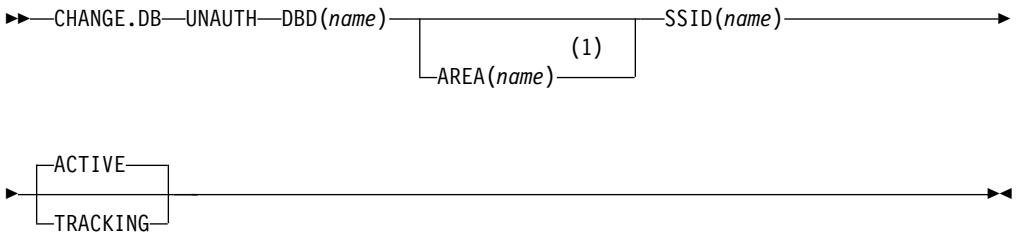


Or:



Notes:

1 Required if using this command for a Fast Path DEDB.



Notes:

1 Required if using this command for a Fast Path DEDB.

Keywords

ALL | DBD(*name*)
Mutually exclusive, optional keywords you use to identify the database for which the record is to be changed.

Restriction: If you specify the UNAUTH keyword, you must specify the DBD name. The ALL keyword is not valid with UNAUTH.

ALL
Specifies that you are changing all the databases registered in the RECON data set.

If any of the databases are HALDB masters or partitions, and you have specified any of the restricted keywords (for example, the PINIT or NOPINIT keywords), a warning message is issued and command processing continues.

DBD
Specifies that you are changing the record of a single database.

Restrictions for HALDB databases: For HALDB databases, specifies the name of a HALDB partition or the HALDB master (if you want to change all partitions of the HALDB master). For HALDB databases, you can use the CHANGE.DB command only as defined in the following table:

Table 2. Keyword definitions for HALDB partition and master

CHANGE.DB	*name* **= TYPE=HALDB**	*name* **= TYPE=PART**
ALTER | NOALTER	Yes	No
AUTH | NOAUTH	Yes (Global)	Yes
DBQUIH | NODBQUIH | DBQUI | NODBQUI	Yes	Yes
ICREQ | NOICREQ	Yes	No
NOBACK | BACKOUT	No	Yes
NOFULLSEG | FULLSG	N/A	N/A
NONRECOV | RECOVABL | USERRCOV	Yes	No
OLRBYTES	Yes	Yes
OLRCAP | OLRNOCAP	Yes	No
OLRDBDS(A | M)	No	Yes
OLRIMSID(name) | NOOWNER	No	Yes
OLRREL | OLRNOREL	No	Yes
OLRRGON | OLRRGOFF	No	Yes
OLRROOTS	Yes	Yes
OLRSEGS	Yes	Yes
OSAM8G | NOOSAM8G	Yes	No
PARTSEL | HIKEY	Yes	No
PINIT | NOPINIT	Yes	Yes
READON | READOFF	No	Yes
REORGI | NOREORGI	Yes	Yes
SHARELVL	Yes	No
TYPEFP | TYPEIMS	No	No
UNAUTH	No	Yes

ALTER | NOALTER

Optional keywords that set or remove the ALTER IN PROGRESS status of a HALDB database or a Fast Path DEDB database.

For HALDB databases, the ALTER keyword sets the ALTER IN PROGRESS flag in the RECON records of all of the partitions in the HALDB database. The ALTER COUNT for the HALDB is updated to the number of partitions involved in the alter process. The NOALTER keyword turns off the ALTER IN PROGRESS flag in all of the partition records and clears the counter in the HALDB record.

For DEDB databases, you can specify the ALTER keyword with the AREA keyword that identifies the area that is being altered. If AREA is not specified, this simulates the replace randomizer function, and the ALTER COUNT for the DEDB is updated. The NOALTER keyword resets the alter status for the entire DEDB and, if specified, the AREA keyword is ignored.

The ALTER IN PROGRESS status cannot be changed if any of the partitions in a HALDB or any of the areas in a DEDB are authorized.

AUTH | NOAUTH
Mutually exclusive, optional keywords you use to specify whether the database is authorized to participate in data sharing.

AUTH
Indicates that authorization processing for data sharing is permitted for the database.

NOAUTH
Indicates that authorization processing for data sharing is prohibited for the database.

DBQUIH | NODBQUIH | DBQUI | NODBQUI
Mutually exclusive, optional keywords you use to specify database quiesce options. These keywords cannot be specified with the ALL keyword.

DBQUIH
Specifies that the QUIESCE HELD flag for the database is to be set on. If the command is issued for a Fast Path DEDB, the command will set the quiesce in progress flag for all areas of the Fast Path DEDB. If the command is issued for a HALDB master, the command will set the quiesce in progress flag for all partitions of the HALDB.

Restriction: DBQUIH is valid only if the QUIESCE IN PROGRESS flag is on or the DBQUI keyword is also included in the command.

NODBQUIH
Specifies that the QUIESCE HELD flag for the database is to be set off. If the command is issued for a Fast Path DEDB, the command will reset the quiesce in progress flag for all areas of the Fast Path DEDB. If the command is issued for a HALDB master, the command will reset the quiesce in progress flag for all partitions of the HALDB.

DBQUI
Specifies that the QUIESCE IN PROGRESS flag for the database is to be set on. If the command is issued for a Fast Path DEDB, the command will set the quiesce in progress flag for all areas of the Fast Path DEDB. If the command is issued for a HALDB master, the command will set the quiesce in progress flag for all partitions of the HALDB.

Restriction: DBQUI and PINIT keywords cannot be specified together.

NODBQUI
Specifies that the QUIESCE IN PROGRESS flag for the database is to be set off. If the command is issued for a Fast Path DEDB, the command will reset the quiesce in progress flag for all areas of the Fast Path DEDB. If the command is issued for a HALDB master, the command will reset the quiesce in progress flag for all partitions of the HALDB.

Restriction: NODBQUI is valid only if the QUIESCE HELD flag is off or the NODBQUIH keyword is also included in the command.

FULLSEG | NOFULLSG

Mutually exclusive, optional keywords that specify whether the full segment image is logged in the X'5950' log record when the segment is updated by a Replace (REPL) call. These keywords are valid only for Fast Path DEDBs.By specifying these keywords, all areas for the specified DEDB are changed to the indicated setting. This setting is saved in the database record for this DEDB

and is used as the default setting for any areas that are registered later for this DEDB. These keywords cannot be specified with the ALL keyword.

This setting is not changed in any areas if the specified DEDB is currently authorized, in which case you receive a DSP0045I message. The DEDB must be unauthorized before you can successfully issue the command.

FULLSEG
Indicates that the full segment image is to be logged.

NOFULLSG
Indicates that only the updated portion of a segment is to be logged.

ICREQ | NOICREQ
Mutually exclusive, optional keywords used to specify whether DBRC should enforce the requirement of taking an image copy of a nonrecoverable or user-recoverable database after it has been loaded or reorganized. This option affects all DBDSs within the DB.

ICREQ
Used to reset the IMAGE COPY NEEDED option. When this option is active, DBRC will set IC NEEDED ON in the DBDS record and increment the IMAGE COPY NEEDED COUNT in the DB record following initial load (PROCOPT=L) or a reorganization of the DBDS.

NOICREQ
Used to disable the IC NEEDED requirement. DBRC will not set IC NEEDED ON in the DBDS record or increment the IMAGE COPY NEEDED COUNT in the DB record following initial load or a reorganization of the DBDS. The IC NEEDED flag can still be set manually with the CHANGE.DB ICON command.

Restrictions:

- If the DB is recoverable, NOICREQ cannot be specified without also specifying NONRECOV or USERRCOV.
- NOICREQ will reset all DBDS IC NEEDED values to OFF and the DB IMAGE COPY NEEDED COUNT to 0.
- If the DB is changed to RECOVABL, this option is reset to ICREQ and IC NEEDED is set ON in all DBDSs; and the DB IMAGE COPY NEEDED COUNT is set accordingly.
- ICREQ | NOICREQ cannot be specified with the ALL keyword.

NOBACK SSID (*name*) | BACKOUT(*value*) SSID (*name*)
Mutually exclusive, optional keywords you use to specify whether the database needs backout by any subsystem. Do not use these keywords for a DEDB.

NOBACK
Indicates that the specified subsystem does not need to back out the database. You use this keyword to delete backout information from the specified database record.

If the held AUTH state and ENCODED state are zero, and if the BACKOUT-NEEDED flag is on, using the NOBACK keyword causes the associated subsystem information to be deleted from the database record.

BACKOUT
Indicates that the specified subsystem needs to back out the database the specified number of times. You need to specify the subsystem name with

the SSID keyword. If you do not specify the SSID keyword with the BACKOUT keyword, this command fails.

SSID(*name***)**
Required keyword specifying which subsystem encountered the backout errors.

name is any valid subsystem name.

With UNAUTH, SSID indicates which entry is to be removed from the associated subsystem information list of the database specified with the DBD keyword, or which SSYS record is to be changed by removing the specified DB/AREA from the authorized databases/areas list.

SSID is required with either the BACKOUT or UNAUTH keywords If coded without either BACKOUT or UNAUTH, the command fails.

NONRECOV | RECOVABL | USERRCOV
Mutually exclusive, optional keywords used to specify whether update logs of the specified database are recorded in the RECON data set.

NONRECOV
Specifies that no recovery is to be performed on the database. The effect of the NONRECOV keyword for full-function databases is different from that for Fast Path databases . When NONRECOV is specified for full-function databases, IMS logs backout information. When NONRECOV is specified for Fast Path databases, IMS does not log backout information.

RECOVABL
Specifies that the database is recoverable and all updates performed for the DBDSs or areas are to be registered in the RECON data set.

USERRCOV
Specifies that update allocations for the Fast Path DEDB are not to be recorded in the RECON data set. Database updates are logged by IMS, but DBRC cannot be used to manage recovery for the database. Recovery is allowed with JCL manually generated by the user. This keyword is valid only for Fast Path DEDBs.

Notes for USERRCOV:

1. USERRCOV should be used only if a critical need to reduce database open time exists and recoverability is required.
2. USERRCOV requires the user to provide their own manually created JCL for change accumulation and recovery.

Restrictions for NONRECOV, RECOVABL, and USERRCOV:

- HALDB master databases are supported by these keywords, but HALDB partitions are not.
- These keywords are not valid when ALL is specified with the CHANGE.DB command.
- You cannot take concurrent image copies of nonrecoverable or user-recoverable databases.
- If the database is registered as RECOVABL or USERRCOV, VIO data sets cannot be used for the output log (IEFRDER) in any job that updates the database. Temporary log data sets, such as VIO, are deleted at job termination, and therefore are not usable for recovery.
- NONRECOV does not support DEDBs with SDEPs.

OLRBYTES(*value*)
Optional parameter to specify a RECON value that indicates the number of bytes of a HALDB partition that have been moved by HALDB online reorganization. The value can range 0 - 47244640256. The value can be specified in decimal or hexadecimal equivalent. Hexadecimal values must be enclosed in quotation marks, such as X'B00000000'.

The OLRBYTES keyword is not valid for, or with, the following items:

- ALL, UNAUTH, OLRCAP, and OLRNOCAP keywords
- HALDB master
- HALDB partition that is marked as not capable of running OLR
- Fast Path databases

OLRCAP | OLRNOCAP
Mutually exclusive, optional keywords that specify if the HALDB can be reorganized online using HALDB OLR. There is no default and these keywords are valid only when the DBD is a HALDB master.

OLRCAP
Specifies (in the master database record and the partition database record) that the HALDB is capable of being reorganized online.

OLRNOCAP
Specifies (in the master database record and the partition database record) that the HALDB is not capable of being reorganized online. If the HALDB OLR cursor is active or if any M-through-V and Y DBDSs are active, the command fails. If the command completes successfully and the M-through-V and Y DBDSs and all of their associated records exist in the RECON data set, the M-through-V and Y DBDSs and all of their associated records are deleted from the RECON data set for all partitions.

OLRREL | OLRNOREL
Mutually exclusive, optional keywords that indicate whether the ownership of the HALDB OLR can be resumed by another IMS when the owning IMS abnormally terminates.

Restriction: In order to have the status changed, the HALDB OLR must be owned by an IMS that has abnormally terminated. The updated status is not reflected in a /DISPLAY DB OLR command output until the OLR is resumed.

OLRROOTS(*value*)
Optional parameter to specify a RECON value that indicates the number of roots of a HALDB partition that have been moved by HALDB online reorganization. The value can range 0 - 2147483647. The value can be specified in decimal or hexadecimal equivalent. Hexadecimal values must be enclosed in quotation marks, such as X'7FFFFFFF'.

The OLRROOTS keyword is not valid for, or with, the following items:

- ALL, UNAUTH, OLRCAP, and OLRNOCAP keywords
- HALDB master
- HALDB partition that is marked as not capable of running OLR
- Fast Path databases

OLRSEGS(*value*)
Optional parameter to specify a RECON value that indicates the number of segments of a HALDB partition that have been moved by HALDB online reorganization. The value can range 0 - 47244640256. The value can be

specified in decimal or hexadecimal equivalent. Hexadecimal values must be enclosed in quotation marks, such as X'B00000000'.

The OLRSEGS keyword is not valid for, or with, the following items:

- ALL, UNAUTH, OLRCAP, and OLRNOCAP keywords
- HALDB master
- HALDB partition that is marked as not capable of running OLR
- Fast Path databases

OLRRGON | OLRRGOFF
Mutually exclusive, optional keywords that specify whether a HALDB OLR is in progress. These keywords are valid only for a HALDB partition.

OLRRGON
Specifies that a HALDB OLR is in progress. Both the A-through-J and the M-through-V data sets contain active data.

OLRRGOFF
Specifies that a HALDB OLR has completed and that either the A-through-J set of data sets or the M-through-V set of data sets contain active data. The flag is changed to indicate that the inactive data sets are now the active data sets. `OLREORG CURSOR ACTIVE = NO` is set in the DB record. The OLRIMSID is cleared.

Note: You might need to manually delete either the DB (HALDB) records or to add a stop time to them using the DELETE.REORG and NOTIFY.REORG commands.

OLRIMSID(*name*) | NOOWNER
Mutually exclusive, optional keywords that specify or change the name of the IMS that owns the HALDB OLR.

OLRDBDS(A | M)
Optional keyword that indicates the active set of DBDSs for this partition is either A-through-J and X or M-through-V and Y.

A Specifies that the A-through-J and X data sets are active.

M Specifies that the M-through-V and Y data sets are active.

If HALDB OLR is cursor active, this keyword indicates the set of DBDSs that are the input data sets.

OSAM8G | NOOSAM8G
Mutually exclusive, optional keywords that specify in the master database record whether a HALDB PHDAM or PHIDAM database supports up to 4 or 8 gigabytes of data in OSAM database data sets. There is no default, and these keywords are valid only when DBD is specified for a HALDB master.

OSAM8G
Specifies that the maximum capacity of OSAM data sets is 8 gigabytes of data. If the HALDB can be reorganized online (OLR capable), the command fails. Only a database that is not OLR capable can be changed to OSAM8G. If the database is currently OLR capable, you must specify OLRNOCAP and OSAM8G to change the maximum size to 8 gigabytes.

NOOSAM8G
Specifies that the maximum capacity of OSAM data sets is 4 gigabytes of data.

PINIT | NOPINIT
Mutually exclusive, optional keywords used to specify whether a HALDB partition needs to be initialized. Use this keyword after deleting and redefining one or more partition data sets without changing the database definition.

If the database specified by DBD keyword is a HALDB master, the change is made to all partitions. Otherwise, only the partition that is specified is changed.

Restrictions:
1. PINIT and NOPINIT keywords are not valid with ALL.
2. PINIT and DBQUI keywords cannot be specified together.

PINIT
Indicates that the partition needs to be initialized by using the DB Prereorganization utility or the HALDB Partition Data Set Initialization utility.

NOPINIT
Indicates that the partition does not need to be initialized.

RANDOMZR(*name*) | NORAND
Optional keywords used to change the name of the randomizing module for the Fast Path DEDB. If the keyword is specified and each IMS uses its own dedicated ACB library for ACBSHR=N, Fast Path verifies that the IMS accessing the DEDB is using this randomizer. Specify NORAND if you do not want the randomizer name to be verified. For ACBSHR=Y, randomizer name verification is never performed regardless of which keyword is specified.

RANDOMZR(*name*)
Specifies the name of the randomizing module for the Fast Path DEDB.

NORAND
Clears the name of the randomizing module for the Fast Path DEDB.

READON | READOFF
Mutually exclusive, optional keywords you use to specify whether the database can be restricted to read-only processing only. Do not use either keyword for a DEDB.

READON
Specifies that the database can be authorized only for read processing.

READOFF
Specifies that the database can be authorized for both read processing and update processing.

SHARELVL(0 | 1 | 2 | 3)
Optional keyword that you use to specify the level of data sharing for which authorized subsystems can share a database. You cannot specify this keyword for authorized DL/I databases.

The numbers 0, 1, 2, and 3 define the four types of data sharing levels.

0 Indicates that the database cannot be shared.

1 Indicates that the database can be shared by one IMS subsystem authorized for update and other IMS subsystems authorized only for read processing (no integrity processing). 1 can also indicate that the database can be shared by multiple IMS subsystems that have been authorized only for read processing. Level 1 is known as database-level sharing.

2 Indicates that the database can be shared by multiple, concurrent subsystems that have been authorized for update in a single-host processor environment. Level 2 is known as intrahost, block-level sharing.

3 Indicates that the database can be shared by multiple, concurrent subsystems that have been authorized for update in a multiple-host processor environment. Level 3 is known as interhost, block-level sharing.

Restrictions:

- The SHARELVL keyword must be greater than 0 for concurrent image copies.
- If you are using IRLM, and have specified SHARELVL 2 or 3, ensure that the VSAM SHAREOPTIONS (3 3) keyword is also specified in the database data set.
- The SHARELVL keyword applies to all areas in the DEDB.
- If you change a DEDB from level 0 or 1 to level 2 or 3, the first coupling facility structure name (CFSTR1) for all VSO areas in the DEDB is set to the name of the area. If you change a DEDB from level 2 or 3 to level 0 or 1, DBRC resets any specified coupling facility structure names to zeros, resets the LKASID keyword to NOLKASID, and resets the MAS keyword to NOMAS.

TYPEFP | TYPEIMS

Mutually exclusive, optional keywords you use to change the RECON record structure to a Fast Path DEDB or a DL/I database.

TYPEFP

Specifies that the database is a Fast Path DEDB and that the record structure in the RECON data set must be changed from IMS to Fast Path.

TYPEIMS

Specifies that the database is a DL/I database and that the record structure in the RECON data set must be changed from Fast Path to IMS.

UNAUTH

Removes an entry from the associated subsystem information list in the database specified by the DBD keyword, or removes an entry from the authorized databases/areas list in the SSYS record specified by the SSID keyword. You must specify the following keywords when you use UNAUTH:

DBD(*name***)**
For the database name

AREA If the database is a Fast Path DEDB

SSID(*name***)**
For the IMS subsystem ID

Restrictions: The following restrictions apply when you use the UNAUTH keyword:

- If any keywords other than DBD, AREA, and SSID are specified for UNAUTH, the command fails.
- If AREA is specified without UNAUTH, the command fails.
- If the inconsistency between the SSYS and DB/AREA records as described above does not exist, the command fails.

AREA(*name*)
Required when UNAUTH is specified for a Fast Path DEDB. The *name* value is the name of the DEDB area. If you specify AREA without UNAUTH, the command fails.

PARTSEL(*pgmname*) | HIKEY
Mutually exclusive, optional keywords that are used to either identify or delete a user partition selection exit program name for a HALDB database. Deleting the partition selection exit program name causes all of the partition selection strings to be cleared. Adding or changing a partition selection exit program name will leave the partition high keys or partition selection strings intact. In all cases, all of the partitions are marked PINIT (partition initialization needed). The command will fail if NOPINIT is also specified. Use the CHANGE.PART command to specify new partition high keys or partition selection strings.

PARTSEL(*pgmname*)
Identifies a user Partition Selection Exit program name for a HALDB. The value is a program name that is up to 8 characters long.

Restriction: This keyword is not valid with ALL.

HIKEY
Removes an existing Partition Selection Exit program that is associated with a HALDB.

Restriction: This keyword is not valid with ALL.

REORGI | NOREORGI
Mutually exclusive, optional parameters that are used by online reorganization products (other than the HALDB Online Reorganization function that is delivered with IMS) to inform DBRC they are active (they intend to reorganize the database) to provide data integrity for full function databases. When the REORG INTENT flag is set to ON, database authorization fails for other reorganization products, batch access, and online image copy processes.

REORGI
Indicates that the database is going to be reorganized and sets the REORG INTENT flag to ON in the database record of the RECON data set if the following conditions have been met:
- Database type is valid. Must be full function including HALDB databases.
- The REORG INTENT flag is off.
- No batch jobs are authorized to the database.
- HALDB Online Reorganization (OLR) cannot have cursor active set (OLREORG CURSOR ACTIVE=YES).
- The database cannot be owned by an IMS OLRIMSID=imsid.

NOREORGI
Indicates that the database is not going to be reorganized (or has just completed reorganization) and sets the REORG INTENT flag to OFF in the database record of the RECON data set if the following conditions have been met:
- Database type is valid. Must be full function including HALDB databases.
- The REORG INTENT flag is on.

Usage notes

If a CHANGE.DB command is issued against a HALDB master that has authorized or opened partitions, either of the following command or command combination will need to be run to cause the changes to take effect for the authorized or opened partitions:

- /START DB *HALDB_master* OPEN command
- /DBRECOVERY DB *HALDB_master* command followed by a /START DB *HALDB_master* command

Example of changing a record for a database identified with the DBD keyword

This example specifies changes to be made to a record in the RECON data set for the database identified with the DBD keyword. The level of data sharing is specified, and the database needs one backout.

```
//CHGDB    JOB
⋮
//SYSIN    DD    *
     CHANGE.DB   DBD(THISDBD) NOAUTH READOFF SHARELVL(2) -
                 BACKOUT(1) SSID(IMSID1)
/*
```

Related concepts:

Non-recovery option (Database Administration)

Levels of data sharing (System Administration)

Coordinating VSAM data set definitions with share options (System Administration)

Nonrecoverable full-function databases (Database Administration)

Related reference:

/RMxxxxxx commands (Commands)

"LIST.DB command" on page 324

"CHANGE.DBDS command"

Related information:

DSP0045I (Messages and Codes)

CHANGE.DBDS command

Use the CHANGE.DBDS command to change the information about a DBDS. This information is contained in a DBDS record in the RECON data set.

If you specify DSN, DDNNEW, or AREANEW while the database or an area of a DEDB is in use, the command fails.

Requirement: The CHANGE.DBDS requires access to the definitions of IMS databases. If the IMS system manages ACBs, the definitions are retrieved from the IMS catalog and the name of the IMS catalog must be defined in the RECON header or the CHANGE.DBDS command itself. If the IMS catalog that is named in the RECON data set or in the command is not available, the command fails.

If the IMS system uses DBD, PSB, and ACB libraries, the IMS DBDLIB data set must be identified in the job stream for the Database Recovery Control utility (DSPURX00) with a ddname of IMS.

Some attributes of a HALDB DBDS can be changed with this command. Use the CHANGE.PART command if you want to change all the DBDSs of a HALDB partition.

Restrictions for HALDB databases: For HALDB databases, you can use this command only as indicated in the following table.

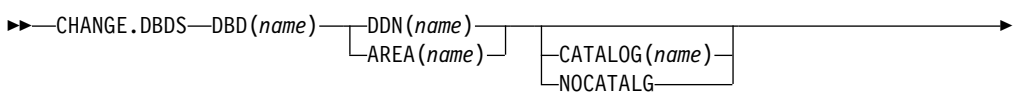
Table 3. CHANGE.DBDS for HALDB databases

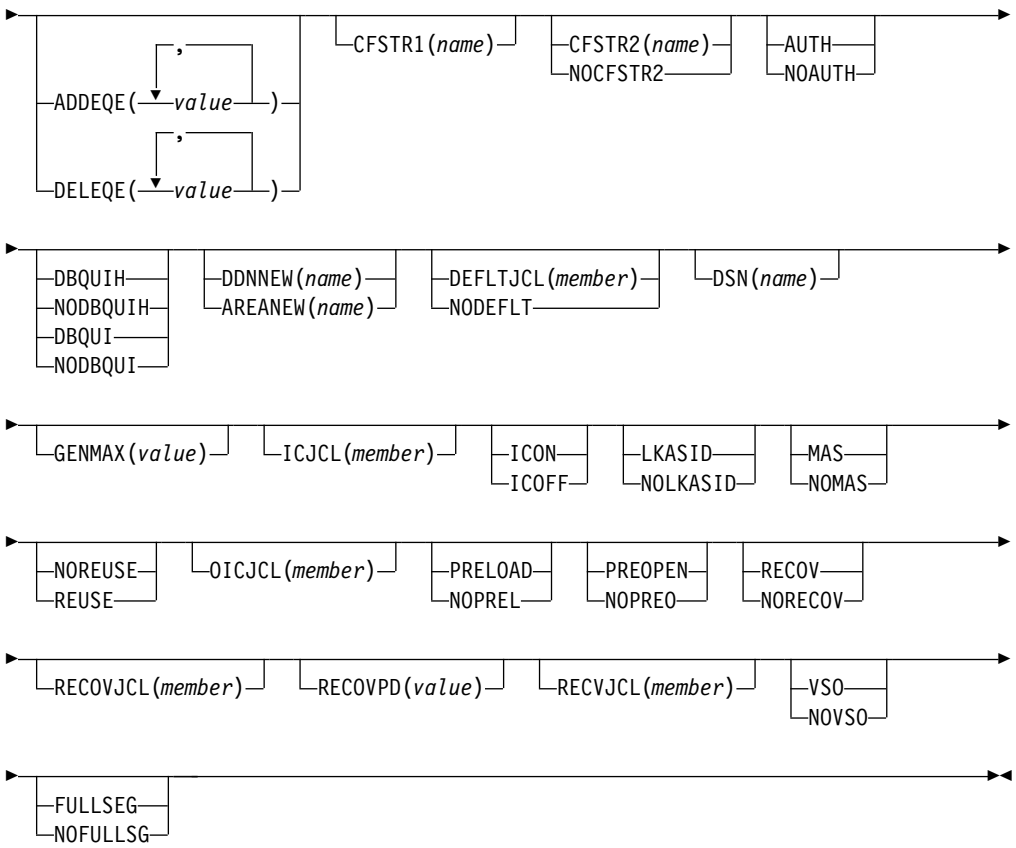
CHANGE.DBDS	Partition data DBDS	Partition index/ILDS DBDS
ADDEQE | DELEQE	Yes	Yes
CFSTR1	N/A	N/A
CFSTR2 | NOCFSTR2	N/A	N/A
AUTH | NOAUTH	N/A	N/A
DBQUIH | NODBQUIH | DBQUI | NODBQUI	No	No
DDNEW | AREANEW	No	No
DEFLTJCL | NODEFLT	Yes	No
DSN	No	No
GENMAX	Yes	No
ICJCL	Yes	No
ICON | ICOFF	Yes	No
NOREUSE | REUSE	Yes	No
OICJCL	Yes	No
LKASID | NOLKASID	N/A	N/A
PRELOAD | NOPREL	N/A	N/A
PREOPEN | NOPREO	N/A	N/A
RECOV | NORECOV	Yes	Yes
RECOVJCL	Yes	No
RECOVPD	Yes	No
RECVJCL	Yes	No
VSO | NOVSO	N/A	N/A
FULLSEG | NOFULLSG	N/A	N/A

Subsections:

- "Syntax"
- "Keywords" on page 86
- "Example of changing a record for a Fast Path DEDB" on page 92

Syntax





Keywords

DBD(*name*)
Required keyword you use to identify by its database name the DBDS or DEDB area whose record is to be changed.

Restriction for HALDB databases: For HALDB databases, *name* specifies the name of a HALDB partition.

DDN(*name*) | AREA(*name*)
Mutually exclusive, required keywords you use to identify the DBDS or DEDB area whose record is to be changed. When you specify DDN, you specify the ddname of the DBDS. When you specify AREA, you specify the name of the area.

CATALOG(*name*) | NOCATALG
Mutually exclusive, optional parameters that you use to indicate whether IMS references the IMS catalog or the DBDLIB during the processing of the CHANGE.DBDS command.

If these parameters are omitted, the value of CATALOG= in the RECON header determines whether IMS uses the IMS catalog or DBDLIB during the processing of this command.

CATALOG(*name*)
Specifies the IMS catalog that IMS references for the processing of the CHANGE.DBDS command.

Specify *name* as an eight-character alphanumeric string that matches the name of an IMS catalog database that is registered in the RECON data set

as a HALDB database. Any IMS catalog name that is less than eight-characters is padded with blanks.

The IMS catalog name is validated during the CHANGE.DBDS command. If the catalog does not exist, is not defined as a HALDB database, or is not available for use, the command fails.

NOCATALG
Specifies that the CHANGE.DBDS command uses the DBD library instead of the IMS catalog.

ADDEQE(*value*) | DELEQE(*value*)
Mutually exclusive, optional keywords you use to change the error queue elements of a DBDS.

ADDEQE
Adds error queue elements to a DBDS. Error queue elements indicate that an I/O error occurred on the data set and that the data set therefore needs to be recovered. Processing continues except for that part of the data set described by the error queue element. Error queue elements can be added only when the DBDS is not in use. The value specified in the variable field is one or more 4-byte hexadecimal values enclosed in quotation marks; for example, `ADDEQE(X'00002345', X'00012345', ...)`.

DELEQE
Deletes error queue elements from a DBDS. Deletion of error queue elements indicates that recovery processing has occurred on that data set. Error queue element deletions can be done only when the DBDS is not in use. The value specified in the variable field is one or more 4-byte hexadecimal values enclosed in quotation marks; for example, `DELEQE(X'00002345', X'00012345', ...)`.

When you add an EQE to a database, the RECOV-NEEDED flag is automatically turned on. When you delete an EQE, and no other EQE flags exist for the database, the RECOV needed flag is turned off. Do not specify RECOV or NORECOV when specifying the ADDEQE or DELEQE keywords.

CFSTR1(*name*)
Optional keyword you specify to change the name of the first coupling facility structure for the identified VSO area. Adhere to the z/OS coupling facility structure naming conventions. This keyword is valid only for VSO areas of DEDBs that are defined with SHARELVL(2 | 3). The area name is the default if the area is changed to VSO and the DEDB is SHARELVL(2 | 3). CFSTR1 is not allowed if the area is authorized, unless it is also being changed from NOVSO to VSO.

CFSTR2(*name*) | NOCFSTR2
Mutually exclusive, optional keywords you use to update or remove the name of the second coupling facility structure for the identified VSO area. These keywords are valid only for VSO areas of DEDBs that are defined with SHARELVL(2 | 3). These keywords are allowed for an authorized area only if it is being changed from NOVSO to VSO.

CFSTR2(*name*)
Specifies the new name of the second coupling facility structure. Adhere to the z/OS naming conventions for coupling facility structures.

Restriction: CFSTR2 cannot be specified with MAS or cannot be specified if the area is already specified as a multi-area coupling facility structure.

NOCFSTR2
Removes the name of the second coupling facility structure (CFSTR2).

AUTH | NOAUTH
Mutually exclusive, optional keywords you use to specify whether the area is available for authorization processing. The AUTH and NOAUTH keywords are valid only if you have specified the AREA keyword.

AUTH
Specifies that the area is available for authorization processing.

NOAUTH
Specifies that authorization processing is prohibited for the area.

If you specify the CHANGE.DBDS AREA(*name*) RECOV command, all ADSs that belong to that area are set to unavailable status at the same time.

DBQUIH | NODBQUIH | DBQUI | NODBQUI
Mutually exclusive, optional keywords you use to specify database quiesce options. These keywords are valid only if you have specified the AREA keyword.

DBQUIH
Specifies that the QUIESCE HELD flag for the DEDB area is to be set on.

Restriction: DBQUIH is valid only if the QUIESCE IN PROGRESS flag is on or the DBQUI keyword is also included in the command.

NODBQUIH
Specifies that the QUIESCE HELD flag for the DEDB area is to be set off.

DBQUI
Specifies that the QUIESCE IN PROGRESS flag for the DEDB area is to be set on.

NODBQUI
Specifies that the QUIESCE IN PROGRESS flag for the DEDB area is to be set off.

Restriction: NODBQUI is valid only if the QUIESCE HELD flag is off or the NODBQUIH keyword is also included in the command.

DDNNEW(*name*) | AREANEW(*name*)
Mutually exclusive, optional keywords you use to change either the database ddname of the specified DBDS or the area name of the specified Fast Path DEDB area in the RECON data set.

When you specify this keyword, the new ddname replaces the existing ddname for all records in the RECON data set that correspond to the specified DBDS.

You must supply a ddname for the IMS DBDLIB data set in the JCL for the CHANGE.DBDS command. The new ddname must be defined in the IMS DBD library and its numeric data set identifier must be unchanged; it also cannot already exist in the RECON data set.

AREANEW is valid only if you have specified the AREA keyword.

DEFLTJCL(*member*) | NODEFLT
Mutually exclusive, optional keywords you use to specify an implicit skeletal JCL default member for the DBDS.

DEFLTJCL
Specifies the new implicit skeletal JCL default member for the DBDS. The specified member is used by the GENJCL.IC, GENJCL.OIC, and GENJCL.RECOV commands to resolve keywords you have defined.

NODEFLT
Removes the default JCL name from the DBDS record when you do are not using a default JCL member.

DSN(*name***)**
Optional keyword you use to change the data set name of the identified DBDS. You cannot use this keyword for a DEDB area.

FULLSEG | NOFULLSG

Mutually exclusive, optional keywords that specify whether the full segment image is logged in the X'5950' log record when the segment is updated by a Replace (REPL) call. These keywords are valid only for Fast Path DEDBs.These keywords cannot be specified if the area is in use. To change attributes of area records for DBRC, the areas must be stopped.

The keywords update the attributes for area records in the RECON data set and override the default attributes that have been set by the INIT.DB or CHANGE.DB commands.

FULLSEG
Indicates that the full segment image is to be logged.

NOFULLSG
Indicates that only the updated portion of a segment is to be logged.

GENMAX(*value***)**
Optional keyword you use to change the maximum number of image copy data sets DBRC is to maintain for the specified DBDS or DEDB area. *value* must be a decimal number from 2 to 255.

If the value you specify is smaller than the number of image copy data sets currently existing for the specified DBDS, records of image copy data sets that are beyond the recovery period are deleted from RECON until the number reaches the specified GENMAX value. Records of image copy data sets with the oldest time stamps are deleted until the number that remains equals the specified GENMAX value.

If you are increasing the GENMAX value and REUSE is specified, use the INIT.IC command to create additional image copy records in the RECON data set. If the number of data sets does not equal GENMAX, reuse of the data sets does not take place and you eventually run out of available data sets for the utility.

ICJCL(*member***)**
Optional keyword you use to change the name of the member of the partitioned data set of skeletal JCL. The GENJCL.IC command uses this name to generate the JCL for a run of the Database Image Copy utility for the specified DBDS or DEDB area.

ICON | ICOFF
Mutually exclusive, optional keywords you use to specify whether a database needs an image copy.

ICON
Specifies that a DBDS needs to have an image copy taken. If the image copy status of a DBDS is IC RECOMMENDED, issuing a CHANGE.DBDS

ICON command changes the image copy status (in the RECON) from IC RECOMMENDED to IC NEEDED. In the associated database record, the IC NEEDED counter increases to indicate how many DBDSs need an image copy.

ICOFF
Specifies that a DBDS does not need an image copy. If the image copy status of a DBDS is IC RECOMMENDED or IC NEEDED, issuing a CHANGE.DBDS ICOFF command resets (turns off) the image copy status. If the image copy status was IC NEEDED before issuing the CHANGE.DBDS ICOFF command, the IC NEEDED counter in the associated database record is decreased by one after the CHANGE.DBDS ICOFF command is issued. The IC NEEDED counter indicates the number of DBDSs that need an image copy.

LKASID | NOLKASID
Mutually exclusive optional keywords you use to specify whether local data caching for the specified area is to be used for buffer lookaside on read requests. The LKASID option is valid only for VSO areas that are specified as SHARELVL(2 | 3). These keywords are allowed for an authorized area only if it is being changed from NOVSO to VSO.

LKASID
Indicates that buffer lookaside is to be performed on read requests for this area.

NOLKASID
Indicates that buffer lookaside is not to be performed on read requests for this area.

For VSO DEDB areas that use a single-area structure, the LKASID value that is defined in the RECON data set for the area takes precedence over the DFSVSMxx PROCLIB member's DEDB LKASID value.

For VSO DEDB areas that use a multi-area structure, LKASID or NOLKASID must be specified by using the DFSVSMxx PROCLIB member. The LKASID value that is defined in the RECON data set for the area is ignored.

MAS | NOMAS
Mutually exclusive, optional keywords you use to specify whether a shared VSO area is to reside in a multi-area coupling facility structure.

MAS
Indicates that the area is to reside in a multi-area coupling facility structure. A second coupling facility structure name (CFSTR2) cannot be specified if MAS is specified. If a second structure is already defined, the second structures name is removed.

Restriction: MAS can be specified only if the DEDB is specified as SHARELVL (2 or 3) and either VSO is specified or the area is already specified as a VSO DEDB. Also, CFSTR2 cannot be specified with MAS or cannot be specified if the area is already specified as a multi-area coupling facility structure.

NOMAS
Indicates that the coupling facility structure (or structures) should contain data for only this area.

NOREUSE | REUSE
Mutually exclusive, optional keywords you use to indicate whether image copy data sets can be reused for subsequent image copy jobs.

NOREUSE
Indicates that image copy data sets already used for the specified DBDS or DEDB area are not to be reused for subsequent image copies. Any existing, unused image copy data set records for the specified DBDS or DEDB area are deleted.

REUSE
Indicates that image copy data sets already used for the specified DBDS or DEDB area can be made available for reuse by subsequent image copies. You cannot specify REUSE if RECON contains any nonstandard image copy data set records for the DBDS or DEDB area.

If GENMAX is higher than the number of existing data sets for the group, use the INIT.IC command to add additional data sets; otherwise, reuse of the data sets does not take place.

OICJCL(*member*)
Optional keyword you use to change the name of the partitioned data set member of skeletal JCL. You cannot use this keyword for a DEDB area. The GENJCL.OIC command uses this name to generate the JCL for a run of the Online Database Image Copy utility for the specified DBDS.

PRELOAD | NOPREL
Mutually exclusive, optional keywords you use to specify whether a VSO DEDB area is to be loaded the next time it is opened.

PRELOAD
Indicates that the area is to be loaded into the data space or coupling facility structure the next time that it is opened. Selecting this option also causes the area to be preopened.

NOPREL
Indicates that the VSO area is not to be loaded into the data space or coupling facility structure the next time that it is opened. CIs are copied into a data space when they are read for the first time.

PREOPEN | NOPREO
Mutually exclusive, optional keywords you use to specify whether a VSO DEDB area is to be opened either, after the first checkpoint following the next control region initialization, or when the next /STA AREA command is processed.

PREOPEN
Indicates that the area is to be opened the next time the control region is started or a /STA AREA command is processed. This option is valid for both VSO and non-VSO areas.

NOPREO
Indicates that the area is not to be preopened the next time the control region is started or a /STA AREA command is processed. You cannot specify this keyword with the PRELOAD keyword.

RECOV | NORECOV
Mutually exclusive, optional keywords you use to specify whether a DBDS or DEDB area needs to be recovered.

RECOV
Specifies that the DBDS or area needs to be recovered. A

RECOVER-NEEDED counter in the associated database record is increased to indicate the number of DBDSs that need to be recovered.

NORECOV
Specifies that the DBDS or DEDB area does not need to be recovered. A RECOVERY-NEEDED counter in the associated database record is decreased to indicate the number of DBDSs that have been recovered.

RECOVJCL(*member***)**
Optional keyword you use to change the name of a member of the partitioned data set of skeletal JCL. The GENJCL.RECOV command uses the member to generate the JCL for a run of DBRC for the specified DBDS or DEDB area.

RECOVPD(*value***)**
Optional keyword you use to change the recovery period for a specified DBDS or DEDB area.

value must be a number from 0 to 999 that represents the number of days the image copies are to be kept in the RECON data set. A 0 indicates no recovery period.

RECVJCL(*member***)**
Optional keyword you use to specify the name of the skeletal JCL member to be used for the GENJCL.RECEIVE command.

VSO | NOVSO
Mutually exclusive, optional keywords you use to specify whether an area resides in virtual storage the next time the control region is initialized or when the next /STA AREA command is processed.

VSO
Indicates that the area is to reside in virtual storage. Areas defined with SHARELVL(0 | 1) are read into and written from a z/OS data space. Areas defined with SHARELVL(2 | 3) use the coupling facility to share data between connected subsystems.

NOVSO
Indicates that this area is not to reside in virtual storage.

If an area was previously specified as SHARELVL(2 | 3), changing the area to NOVSO clears the coupling facility structure names and resets the LKASID setting to NOLKASID. NOVSO cannot be specified if the area is in use.

Example of changing a record for a Fast Path DEDB

This example specifies changes to be made to the record in the RECON data set for the Fast Path DEDB that is identified by the DBD and AREA keywords. The image copy data sets for the specified DEDB area are not reused, and the maximum number of images that DBRC maintains is two. In addition, the image copy data sets for the specified DBDS are kept in the RECON data set for at least 15 days.

```
//CHGDBDS  JOB

.
.
.
//SYSIN    DD   *
     CHANGE.DBDS DBD(DB3)  AREA(DD3)  NOREUSE -
                 GENMAX(2) RECOVPD(15)
/*
```

Example using the DBDLIB instead of the default catalog name

```
//CHANGEDBDS JOB

⋮
 //IMS      DD   DSN=IMS.DBDLIB,DISP=SHR
 //SYSIN    DD   *
      CHANGE.DBDS  DBD(DBD002)  DDN(DDN003) GENMAX(2) REUSE -
                   ICJCL(ICJCLX) RECOVJCL(RECOVJCX) DSN(DSN003) –
                   NOCATALG
 /*
```

Example using the catalog name GREGCD00 instead of the default catalog name

```
//CHANGEDBDS JOB

⋮
//SYSIN    DD   *
      CHANGE.DBDS  DBD(DBD002)  DDN(DDN003) GENMAX(2) REUSE -
                   ICJCL(ICJCLX) RECOVJCL(RECOVJCX) DSN(DSN003) –
                   CATALOG(GREGCD00)
 /*
```

Related concepts:

Defining a VSO DEDB area (Database Administration)

Coupling facility structure naming convention (Database Administration)

Recovery period of image copy data sets and GENMAX (System Administration)

Related reference:

"LIST.DBDS command" on page 326

"CHANGE.PART command" on page 99

"INIT.DBDS command" on page 294

/RMxxxxxx commands (Commands)

Related information:

DSP0045I (Messages and Codes)

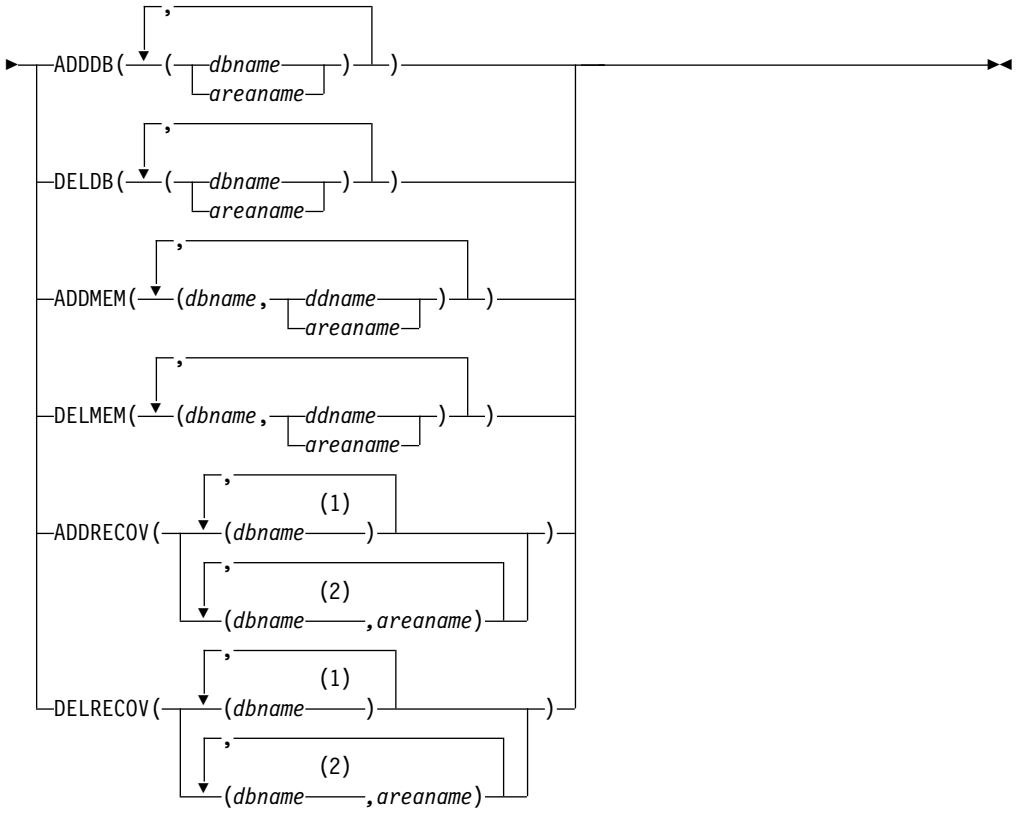
CHANGE.DBDSGRP command

Use a CHANGE.DBDSGRP command to change the information about a DBDS or database group. This information is contained in a DBDS group record in the RECON data set.

Subsections:

- "Syntax"
- "Keywords" on page 94
- "Example of changing a group of DBDSs" on page 96

Syntax



Notes:

1 Full-function database name

2 Fast Path DEDB name

Keywords

GRPNAME(*name***)**
Required keyword you use to identify the DBDSGRP to be changed. A record with that name must already exist.

ADDDB(*dbname* | *areaname***)** | **DELDB(***dbname* | *areaname***)** | **ADDMEM(***dbname***,***ddname* | *areaname***)** | **DELMEM(***dbname***,***ddname* | *areaname***)** | **ADDRECOV(***dbname***,***areaname***)** | **DELRECOV(***dbname***,***areaname***)**
Mutually exclusive, optional keywords you use to identify the member or members to be added to or deleted from the group. A member can belong to any number of DB groups or DBDS groups but can belong only to one recovery group.

ADDDB(*dbname***)**
Identifies one or more database or area names to be added to a DB group. The name specified can be a full-function database name, a Fast Path DEDB name, or an area name.

DELDB(*dbname***)**
Identifies one or more database or area names to be deleted from a DB group. The name specified can be a full-function database name, a Fast Path DEDB name, or an area name.

ADDMEM(*dbname*,*ddname*)**
Identifies one or more members to be added to a DBDS group, where *dbname* is the database name and *ddname* is the DD statement name or the area name.

DELMEM(*dbname*,*ddname*)**
Identifies one or more members to be deleted from a DBDS group, where *dbname* is the database name and *ddname* is the DD statement name or the area name.

ADDRECOV(*dbname*,*areaname*)**
Identifies one or more databases or DEDB areas to be added to a recovery group, where *dbname* is the database (or DEDB) name and *areaname* is the DD statement name or the area name.

DEDB areas can be a part of a recovery group, but DEDB databases cannot. HALDB databases can be a part of a recovery group, but individual HALDB partitions cannot.

If you add a DEDB area to the recovery group, you must specify both the DEDB area and the DEDB database that the area belongs to. That is, to add a DEDB area to the recovery group, you must specify both *dbname* and *areaname*. You cannot specify only *areaname*.

If you add a full-function database to the recovery group, you do not need to specify *areaname* because *areaname* applies only to Fast Path databases.

If the group specified is not a recovery group, the command fails with message DSP0077I.

A database or area can belong to only one recovery group. If any of the members specified by ADDRECOV already belong to another recovery group, the command fails with message DSP0078I.

DELRECOV(*dbname*,*areaname*)**
Identifies one or more DBs or DEDB areas to be deleted from a recovery group, where *dbname* is the database (or DEDB) name and *areaname* is the DD statement name or the area name.

If a DEDB area is to be deleted from the recovery group, both *dbname* and *areaname* must be specified. If the group specified is not a recovery group, the command fails with message DSP0077I.

If you delete all the members of a group, the record of that group is deleted from the RECON data set.

Restrictions for HALDB databases: For HALDB databases, *dbname* specifies the name of a HALDB partition or the HALDB master. When specifying HALDB database names, use the CHANGE.DBDSGRP command only as defined in the following table:

Table 4. Specifying HALDB master or partition database names for the CHANGE.DBDSGRP command

CHANGE.DBDSGRP	*dbname* **= HALDB Master**	*dbname* **= Partition**
ADDDB	Yes	Yes
DELDB	Yes	Yes
ADDMEM	No	Yes
DELMEM	No	Yes

Table 4. Specifying HALDB master or partition database names for the CHANGE.DBDSGRP command (continued)

CHANGE.DBDSGRP	*dbname* = HALDB Master	*dbname* = Partition
ADDRECOV	Yes	No
DELRECOV	Yes	No

Example of changing a group of DBDSs

In this example, a group of DBDSs is changed.

```
//CHGDBGRP   JOB
⋮
//SYSIN      DD  *
     CHANGE.DBDSGRP    GRPNAME(GRP1)  -
                       ADDMEM((DB1,DD1),(DB2,DD2))
/*
```

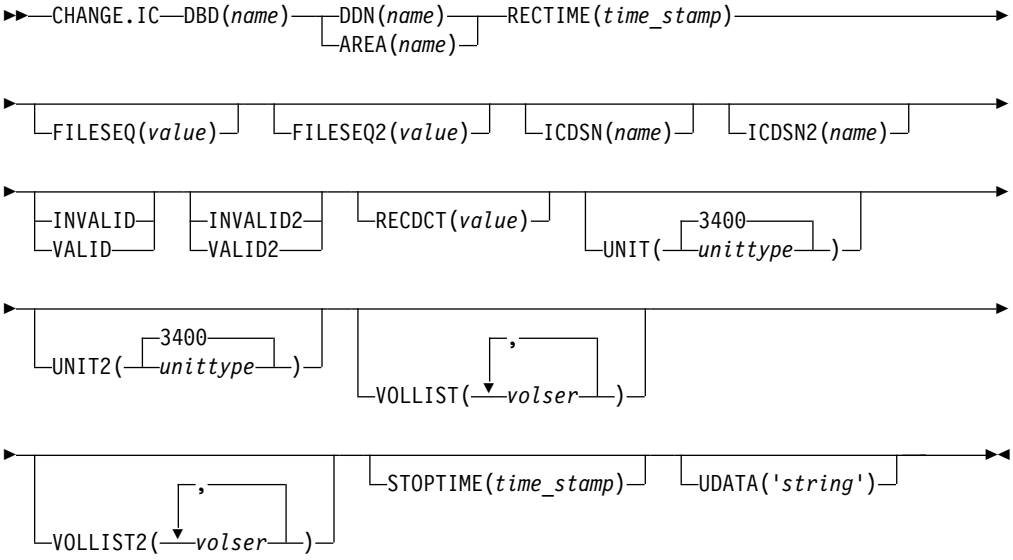
CHANGE.IC command

Use a CHANGE.IC command to modify information contained in an image copy record in the RECON data set.

Subsections:

- "Syntax"
- "Keywords"
- "Examples" on page 98

Syntax



Keywords

DBD(*name***)**

Required keyword you use to identify the database name of the DBDS whose image copy record is to be modified.

Restrictions for HALDB databases: For HALDB databases, *dbname* specifies the name of a HALDB partition.

DDN(*name***) | AREA(***name***)**
Mutually exclusive, required keywords you use to identify the name of the DBDS or DEDB area to which the image copy record being modified is related.

RECTIME(*time_stamp***)**
Required keyword you use to identify the specific image copy data set record to be changed.

Use the Run time marked with an asterisk (*) from the listing of the IMAGE record. The time stamp must be in standard form.

FILESEQ(*value***)**
Optional keyword you use to change the file sequence number in the record of the identified image copy data set.

FILESEQ2(*value***)**
Optional keyword you use to change or add the file-sequence number in the record of the identified duplicate image copy data set.

ICDSN(*name***)**
Optional keyword you use to change the data set name of the identified image copy data set.

ICDSN2(*name***)**
Optional keyword you use to change or add the data set name of the identified duplicate image copy data set in an image copy record.

To change the name of the duplicate image copy data set, a record of the first image copy data set must exist in the RECON data set.

INVALID | VALID
Mutually exclusive, optional keywords that you use to prevent or permit the use of an image copy data set as input to a subsequent run of the Database Recovery utility (DFSURDB0).

INVALID
Prevents the use of the specified image copy data set as input to a subsequent run of the Database Recovery utility. If the invalidated image copy data set is reused, it is automatically marked as valid.

VALID
Permits the use of a previously invalidated image copy data set as input to a subsequent run of the Database Recovery utility.

The Database Recovery utility expects this image copy data set as input unless it is marked as INVALID, in which case the utility expects a duplicate image copy data set as input.

INVALID2 | VALID2
Mutually exclusive, optional keywords that you use to prevent or permit the use of a duplicate image copy data set as input to a subsequent run of the Database Recovery utility (DFSURDB0).

INVALID2
Prevents the use of the specified, duplicate image copy data set as input to a subsequent run of the Database Recovery utility. If the invalidated, duplicate image copy data set is subsequently reused, it is automatically marked as valid.

VALID2
Permits the use of a previously invalidated, duplicate image copy data set as input to a subsequent run of the Database Recovery utility.

If both INVALID2 and VALID2 are specified, the last one specified is used.

RECDCT(*value***)**
Optional keyword you use to change the count of the records in the corresponding image copy data set in the specified image copy record. CHANGE.IC *value* must be a decimal number up to 2 147 483 647.

UNIT(3400 | *unittype*)
Optional keyword you use to change the unit type that is recorded in the specified image copy record. The unit type can be up to 8 alphanumeric characters.

UNIT2(3400 | *unittype*)
Optional keyword you use to change the unit type that is recorded in the specified duplicate image copy record. The unit type can be up to 8 alphanumeric characters.

VOLLIST(*volser***)**
Optional keyword you use to change, in the image copy record, the volume serial numbers of the volumes on which the identified image copy data set resides.

VOLLIST2(*volser***)**
Optional keyword you use to change or add, in the image copy record, the volume serial numbers of the volumes on which the identified duplicate image copy data set resides.

STOPTIME(*time_stamp***)**
Optional keyword you use to specify the time when an image copy has completed. The time stamp must be in standard form and cannot be less than the image copy start time. If this is an HSSP CIC that is in progress, specifying a valid stop time terminates the HSSP CIC and resets the in-progress indicators in the IC record and the DBDS record.

UDATA('*string*')
Optional keyword you use to specify up to 80 bytes of information about the identified, standard image copy data set. You can use the variable field of this keyword to describe how the standard image copy data set was created. The *string* value must be enclosed in single quotation marks if blanks or any other special characters are included.

Examples

Example 1: Changing an image copy record

In this example, information in an image copy record that is identified by the DBD, DDN, and RECTIME keywords is to be changed in the RECON data set. The new data set names of both image copy data sets (specified in the ICDSN and ICDSN2 keywords) follow the default naming convention. The volume serial numbers on which the image copy data sets reside are also to be changed as specified in the VOLLIST and VOLLIST2 keywords.

```
 //CHGIC    JOB

⋮
 //SYSIN    DD   *
   CHANGE.IC   DBD(DBDKSDS1)  DDN(DDNKSDS1)  -
```

```
    ICDSN(IMS.DBDKSDS1.DDNKSDS1.IC.ICDSN02) -
    ICDSN2(IMS.DBDKSDS1.DDNKSDS1.IC2.ICDSN02)   -
    VOLLIST(ICVOL1,ICVOL2,ICVOL3) FILESEQ2(2)   -
    VOLLIST2(ICVOL4) RECTIME(07092131414323456)
/*
```

Example 2: Changing the user data in an image copy record

In this example, information in an image copy record that is identified by the DBD, DDN, and RECTIME keywords is to be changed in the RECON data set. The changed user data is included in the UDATA parameter.

```
//CHGIC    JOB
⋮
//SYSIN    DD   *
  CHANGE.IC   DBD(DBDKSDS1)  DDN(DDNKSDS1)  -
    RECTIME(09152104355891200) –
    UDATA('RECOVERY POINT FOR ALL PARTITIONS TAKEN ON IMS12 EACH MONTH')
/*
```

Related reference:

"INIT.DBDS command" on page 294

"DBRC time stamps" on page 52

CHANGE.PART command

Use a CHANGE.PART command to change attributes of a HALDB partition.

The changes apply to all the DBDSs of the partition. Some attributes such as skeletal JCL member names can be changed for a single DBDS of a partition by a CHANGE.DBDS command.

Changing the following fields requires the partition to be initialized before it can be used:

- DSN prefix (DSNPREFX)
- Randomizing module name (RANDOMZR)
- Relative block number (HIBLOCK)
- Number of root anchor points (ANCHOR)
- Bytes (BYTES)
- Partition selection string or partition high key (KEYSTRNG)
- Block sizes (BLOCKSZE)

Requirement: If KEYSTRNG is being set or the partition is being enabled, the IMS must reference the database definition when processing the CHANGE.PART command. If the IMS system manages ACBs, the definitions are retrieved from the IMS catalog and the name of the IMS catalog must be defined in the RECON header or the CHANGE.PART command itself. If the IMS system uses DBD, PSB, and ACB libraries, the IMS DBDLIB data set must be identified in the job stream for the Database Recovery Control utility (DSPURX00) with a ddname of IMS.

Use the DISABLE keyword to temporarily remove a partition from active use. Use the ENABLE keyword to return the partition to active use. You would normally DISABLE a partition prior to deleting it, but do not want to remove all the recovery related information that DBRC maintains until the changes have been tested. A partition marked as DISABLE can be deleted or made active again by marking it ENABLE. A partition that is changed from DISABLE to ENABLE must be recovered before it can be used.

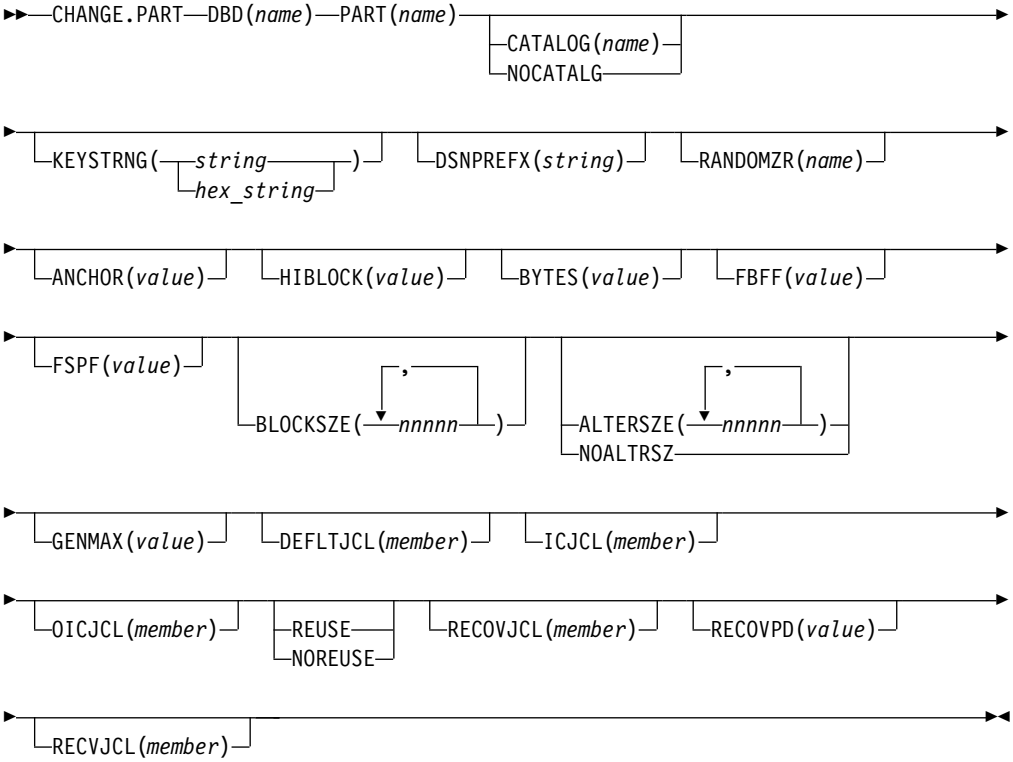
Restriction: No other keywords are allowed on a CHANGE.PART command if either DISABLE or ENABLE is specified.

Subsections:

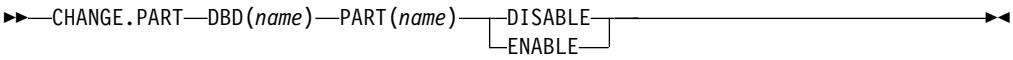
- "Syntax"
- "Keywords"
- "Examples of changing a partition" on page 106

Syntax

Use one of the following diagrams for the CHANGE.PART command.



Or:



Keywords

DBD(*name*)
Required keyword used to identify the HALDB database.

PART(*name*)
Required keyword you use to identify the partition.

CATALOG(*name*) | NOCATALG
Mutually exclusive, optional parameters that you use to indicate whether IMS references the IMS catalog or the DBDLIB during the processing of the CHANGE.PART command.

If these parameters are omitted, the value of CATALOG= in the RECON header determines whether IMS uses the IMS catalog or DBDLIB during the processing of this command.

CATALOG(*name*)
Specifies the IMS catalog that IMS references for the processing of the CHANGE.PART command.

Specify *name* as an eight-character alphanumeric string that matches the name of an IMS catalog database that is registered in the RECON data set as a HALDB database. Any IMS catalog name that is less than eight-characters is padded with blanks.

The IMS catalog name is validated during the CHANGE.PART command. If the catalog does not exist, is not defined as a HALDB database, or is unavailable for use, the command fails.

NOCATALG
Specifies that the CHANGE.PART command uses the DBD library instead of the IMS catalog.

KEYSTRNG(*string* | *hex_string*)
Optional keyword used to change a HALDB partition high key value or a selection string for use by a partition selection exit. Specified as a character value up to 256 characters long or a hexadecimal value up to 512 characters long. Character values must be alphanumeric (with no embedded blanks or commas unless the string is enclosed by single quotation marks). Unless enclosed by single quotation marks, the character string will be folded to uppercase. Hexadecimal values must be enclosed by single quotation marks and preceded by the letter X, for example: `KEYSTRNG(X'D7C1D9E3D2C5E8')`.

If no partition selection routine is specified for the HALDB, KEYSTRNG defines a partition high key value which cannot be longer than the root key length. If the high key length is less than the defined root key length, the high key value is padded with hex 'FF's up to the defined root key length. The partition high key values must be unique for each partition within a HALDB database.

If a partition selection routine is specified for the HALDB, KEYSTRNG defines a Partition Selection String which is passed to the partition selection routine. The Partition Selection String may be up to 256 bytes long and consist of simple character information. If it contains non-printable characters, it must be identified using hex notation. A hex character string is enclosed by single quotation marks and prefixed with an X.

The command fails if the partition is in use, or any other partition affected by a change to the high key value is in use. The partition and all other partitions affected by a change to the high key value are marked as needing partition initialization (PINIT is set on) and cannot be used until initialized.

Requirement: When you issue a CHANGE.PART KEYSTRNG command, IMS must reference the database definition of the partition. If the IMS system manages ACBs, the definitions are retrieved from the IMS catalog and the name of the IMS catalog must be defined in the RECON header or the CHANGE.PART command itself. If the IMS system uses DBD, PSB, and ACB libraries, the IMS DBDLIB data set must be identified in the job stream for the Database Recovery Control utility (DSPURX00) with a ddname of IMS.

DSNPREFX(*string*)
Optional keyword used to change the data set name prefix for the partition

data sets contained in a HALDB database. Specified as a value, up to 37 characters long, that is a valid JCL data set name.

RANDOMZR(*name*)
Optional keyword used to change the name of the randomizing module for HALDB PHDAM databases only. A randomizing module controls root segment placement in, or retrieval from, the PHDAM HALDB.

ANCHOR(*value*)
Optional keyword used to change the number of RAPs (root anchor points) desired in each control interval or block in the root addressable area of a PHDAM HALDB. The value specified must be between 1 and 255. Typical values are from 1 to 5.

HIBLOCK(*value*)
Optional keyword used to change the maximum relative block number value that the user wants to enable a randomizing module to produce for this HALDB database. This value determines the number of control intervals or blocks in the root addressable area of an PHDAM HALDB. The value may range between 1 and 16,777,215 (2**24 - 1). This keyword is valid only for a PHDAM HALDB.

BYTES(*value*)
Optional keyword used to change the maximum number of bytes of a HALDB record that can be stored into the root addressable area in a series of inserts unbroken by a call to another HALDB record. The value has a range of 0 to 16,777,215 (2**24 - 1). This keyword is valid for PHDAM HALDB databases only.

A value of 0 (zero) means that all bytes are addressable. Setting BYTES to 0 is equivalent to omitting BYTES on the DBD definition

FBFF(*value*)
Optional keyword used to change the free block frequency factor (fbff) which specifies that every *n*th control interval or block in this data set group is left as free space during database load or reorganization (where FBFF=*n*). The range of FBFF includes all integer values from 0 to 100 except 1.

FSPF(*value*)
Optional keyword used to change the free space percentage factor. It specifies the minimum percentage of each control interval or block that is to be left as free space in this data set group. value may be any number between 0 and 99.

BLOCKSZE(*nnnnn*)
Optional keyword used to change the block size for OSAM data sets. Specify an even numeric value no greater than 32766. The block size value is used by OSAM only. You may specify up to 10 values, one for each data set group defined in the DBD. Omitted values remain unchanged.

ALTERSZE(*nnnnn*) | NOALTRSZ
Mutually exclusive, optional keywords that either set or clear new OSAM block sizes or VSAM control interval (CI) sizes for the output partition data sets of a HALDB alter operation.

After an alter operation is started on a HALDB database, the block or CI sizes of the partition data sets cannot be changed or cleared until after the alter operation is complete for the database and online change is performed.

ALTERSZE(*nnnnn*)
For the HALDB alter function, specifies new OSAM block sizes or VSAM CI sizes for the output partition data sets of an alter operation.

Requirements:

- The new sizes must be stored in the RECON data set before the INITIATE OLREORG OPTION(ALTER) command is issued to start the alter operation.
- The INITIATE OLREORG command fails if the ALTER option is specified, but no changes to the database structure are detected in the input database member. Consequently, you cannot change the block or CI sizes by using the ALTER option unless you are also making a structural change to the database.

Specify even numeric values no greater than 32766. You may specify up to 10 values, one for each data set group defined in the DBD. Omitted values remain unchanged.

The values must be specified as positional, comma-separated values. The value in the first position applies to the first data set group. The value in the second position applies to the second data set group, and so on.

For example, the following ALTERSZE keyword sets a new block or CI size for the third data set group, but leaves the sizes unchanged for the first and second data set groups, as well as for the fourth through tenth data set groups, if they exist: `ALTERSZE(,,4096)`.

The ALTERSZE values are stored in the partition record in the RECON data set only until the alter operation is complete and online change is performed. After online change is complete, new OSAM block sizes are saved in the OSAM BLOCK SIZE field of the partition record. VSAM CI sizes are not saved in the partition record.

If ALTERSZE values have already been set, you can change the values by reissuing the CHANGE.PART command and specifying new ALTERSZE values for the data sets that need to change.

To back out a change so that an output data set will have the same block or CI size as its corresponding input data set, specify the block or CI size of the input data set on the ALTERSZE keyword and reissue the command.

For OSAM data sets, after you back out the ALTERSZE value, the value IMS displays for the data set in the ALTER BLOCK SIZE field of the partition record is 0, which indicates that the HALDB alter function will not change the block size.

For VSAM data sets, after you back out the ALTERSZE value, the value that IMS displays for the data set in the ALTER BLOCK SIZE field of the partition record is the value that you entered to back out the ALTERSZE value. IMS does not validate that this value is the same as the CI size of the input data set. If the output data sets do not exist, IMS creates the output data sets automatically with the CI sizes that are set in the ALTER BLOCK SIZE field, even if the sizes are different than the CI sizes of the input data sets. If output data sets exist and their CI sizes do not match the ALTERSZE values specified for them, the alter of the partition fails, in which case the discrepancy must be corrected and the alter of the partition resumed after all other partitions complete alter processing.

To view the ALTERSZE values for a partition, use the LIST.DB command. If the block or CI size of a data set is not changing, its ALTERSZE value is 0. A 0 indicates that the block or CI size of the output data set will be the same as that of the input data set. If none of the block or CI sizes of the data sets in a partition are changing, the ALTER SIZE output field is not displayed.

NOALTRSZ
For the HALDB alter function, clears all ALTERSZE values from a partition record in the RECON data set.

After ALTERSZE values are cleared, if new ALTERSZE values are not set before an alter operation starts, the output data sets are created with the same block or CI sizes as the input data sets.

After all of the ALTERSZE values are cleared by the NOALTRSZ keyword, the ALTER SIZE output field is no longer displayed.

GENMAX(*value*)
Optional keyword used to change the maximum number of image copies that DBRC is to maintain for the partition DBDSs. If you identify a partition DBDS with the NOREUSE keyword, the oldest image copy beyond the recovery period is deleted when the number of image copies exceeds the GENMAX value. If you identify it with the REUSE keyword, the oldest image copy beyond the recovery period is reused. Specified as a numeric value from 2 to 255. All partition DBDSs will be changed to use this GENMAX value. The CHANGE.DBDS command can be used to change this for individual partition DBDSs.

DEFLTJCL(*member*)
Optional keyword used to change the implicit skeletal JCL default member for a HALDB Partition DBDS. The specified member is used by the GENJCL.IC, GENJCL.OIC, and GENJCL.RECOV commands in order to resolve keywords you have defined. All partition DBDSs will be changed to use this DEFLTJCL member. The CHANGE.DBDS command can be used to change this for individual partition DBDSs.

ICJCL(*member*)
Optional keyword used to change the name of a member of a partitioned data set that contains skeletal JCL. When you issue a GENJCL.IC command, DBRC uses this member to generate the JCL to run the Database Image Copy utility (or the Database Image Copy 2 utility) for the partition DBDS specified on the GENJCL command. All partition DBDSs will be changed to use this ICJCL member. The CHANGE.DBDS command can be used to change this for individual partition DBDSs.

NOREUSE | REUSE
Mutually exclusive, optional keywords used to change whether the supported image copy utilities are to reuse previously used image copy data sets. REUSE enables the GENJCL.IC command or the GENJCL.OIC command to generate a job that causes the supported image copy utilities to reuse the oldest image copy data set (for the DBDS specified on the GENJCL command) when the GENMAX value for it is exceeded. REUSE requires that you create empty image copy data sets for future use by the supported image copy utilities. In addition, you must use an INIT.IC command to record their existence in the RECON data set. The NOREUSE keyword prohibits such actions. All partition DBDSs will be changed to use the keyword specified. The CHANGE.DBDS command can be used to change this for individual partition DBDSs.

OICJCL (*member*)
Optional keyword used to change the name of a member of a partitioned data set that contains skeletal JCL. When you issue a GENJCL.OIC command, DBRC uses this member to generate the JCL to run the Online Image Copy utility for the partition DBDS specified on the GENJCL command. Appropriate

partition DBDSs will be changed to use this OICJCL member. The CHANGE.DBDS command can be used to change this for individual partition DBDSs.

RECOVJCL(*member*)
Optional keyword used to change the name of a member of a partitioned data set that contains skeletal JCL. When you issue the GENJCL.RECOV command, DBRC uses this member to generate the JCL to run the Database Recovery utility for the partition DBDS specified on the GENJCL command. All partition DBDSs will be changed to use this RECOVJCL member. The CHANGE.DBDS command can be used to change this for individual partition DBDSs.

RECOVPD(*value*)
Optional keyword used to change the recovery period for a specified partition DBDS. Specify a numeric value from 0 to 999 that represents the number of days you want DBRC to maintain sufficient recovery-generation information in the RECON data set. All partition DBDSs will be changed to use this RECOVPD value. The CHANGE.DBDS command can be used to change this for individual partition DBDSs.

RECVJCL(*member*)
Optional keyword used to change the name of the skeletal JCL member to be used by the GENJCL.RECEIVE command. All partition DBDSs will be changed to use this RECVJCL member. The CHANGE.DBDS command can be used to change this for individual partition DBDSs.

DISABLE | ENABLE
Mutually exclusive, required keywords used to change the state of the partition.

DISABLE marks the partition as unavailable for use by IMS and most utilities. In most situations, a partition marked DISABLE will not be known by IMS (it is treated as if it is not registered to DBRC). DBRC LIST commands will list information about partitions marked DISABLE and the HALDB Partition Definition utility (PDU) displays partitions which are marked DISABLE. The partition still counts towards the maximum number of partitions which may be defined for a HALDB.

Although a partition that is marked DISABLE is not removed from DBRC groups (a CAGROUP, DBDSGRP, DBGROUP and RECOVGRP), it is generally not processed as part of the group when the group is used in a DBRC command. This also applies to implied groups used in a command. The one exception to this is the GENJCL.CA command and the execution of the IMS Database Change Accumulation utility. Changes will be accumulated for partitions that are members of a CA Group, even when marked DISABLE. GENJCL.IC and GENJCL.RECOV commands fail for partitions that are marked DISABLE. If the commands specify a group of any kind, partitions that are marked DISABLE are skipped, that is, JCL is not generated and no message is issued.

ENABLE makes a partition which had been made DISABLE available for use by IMS again. The partition is set to a Recovery Needed status and must be recovered before being used by IMS. Setting a partition to ENABLE has the same effect as adding a new partition to the HALDB. If the HALDB uses a partition selection exit, all the other partitions are set to a Partition Initialization Needed status. If the HALDB uses high key values, the partition with the next high key value is set to a Partition Initialization Needed status.

Requirement: When you issue a CHANGE.PART ENABLE command, IMS must reference the database definition of the partition. If the IMS system manages ACBs, the definitions are retrieved from the IMS catalog and the name of the IMS catalog must be defined in the RECON header or the CHANGE.PART command itself. If the IMS system uses DBD, PSB, and ACB libraries, the IMS DBDLIB data set must be identified in the job stream for the Database Recovery Control utility (DSPURX00) with a ddname of IMS.

Examples of changing a partition

This example specifies changes to be made to the partition that is identified by the DBD and NAME keywords. The high key for the partition is changed, as well as the name of the randomizer routine. The maximum number of images that DBRC maintains is 12. In addition, the image copy data sets for the specified DBDS are kept in the RECON data set for at least 15 days. The partition and the next one will be marked as needing partition initialization as a result of the change.

```
 //CHGPART JOB
.
.
.
//IMS      DD   DSN=IMS.DBDLIB,DISP=SHR
...
//SYSIN DD *
  CHANGE.PART DBD(DB3) PART(PART3) KEYSTRNG(X'D7C1D9E3D2C5E8') -
                                    RANDOMZR(RNDNAME2) GENMAX(12) RECOVPD(15)
/*
```

The following example changes the partition identified by the DBD and PART keywords to a DISABLE status.

```
//CHGPART JOB
...
//SYSIN DD *
CHANGE.PART DBD(DB3) PART(PART3) DISABLE
/*
```

Example using the DBDLIB instead of the default catalog name

```
//CHANGEPART JOB
.
.
.
//IMS      DD  DSN=IMS.DBDLIB,DISP=SHR
//SYSIN    DD  *
  CHANGE.PART DBD(DBHDOK01) PART(PDHDOKE) KEYSTRNG('ABCDEFGHIJKL--

MNOPQRSTUVWXYZ0123456789 ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJ-
KLMNOPQRSTUVWXYZ0123456789- ABCD--

EFGHIJKLMNOPQRSTUVWXYZ0123456789') -
  DSNPREFX(TESTSYS1.PROJECT) -
BLOCKSZE(32000,,4000) –
NOCATALG
/*
```

Example using the catalog name GREGCD00 instead of the default catalog name

```
//CHANGEPART JOB
.
.
.
//SYSIN    DD  *
  CHANGE.PART DBD(DBHDOK01) PART(PDHDOKE) KEYSTRNG('ABCDEFGHIJKL--
```

```
MNOPQRSTUVWXYZ0123456789 ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJ-
KLMNOPQRSTUVWXYZ0123456789- ABCD--

EFGHIJKLMNOPQRSTUVWXYZ0123456789') -
  DSNPREFX(TESTSYS1.PROJECT) -
BLOCKSZE(32000,,4000) –
CATALOG(GREGCD00)
/*
```

Related reference:

HALDB Partition Data Set Initialization utility (DFSUPNT0) (Database Utilities)

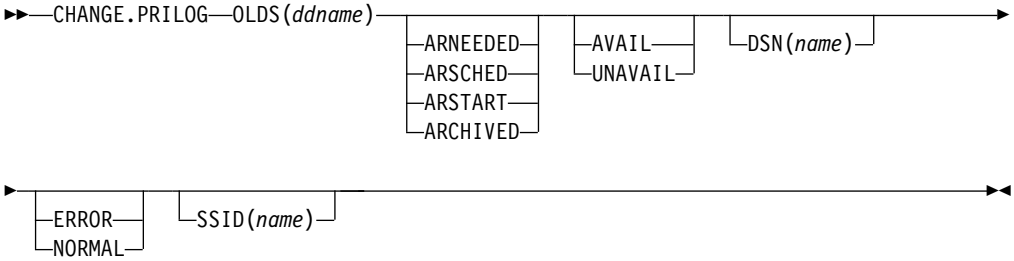
CHANGE.PRILOG OLDS command

Use the CHANGE.PRILOG OLDS command to change information about a primary online data set (PRIOLDS) in the RECON data set.

Subsections:

- "Syntax"
- "Keywords"
- "Example of renaming an OLDS" on page 108

Syntax



Attention: For this command you must specify the OLDS keyword. If OLDS is not specified, DBRC will default to processing an RLDS, which will not provide the desired results.

Keywords

OLDS(*ddname***)**
Required keyword you use to specify the OLDS for which the RECON record is to be changed.

ARNEEDED | ARSCHED | ARSTART | ARCHIVED
Mutually exclusive, optional keywords you use to change the archive status of an OLDS. You cannot change the archive status after an OLDS has been archived.

ARNEEDED
Indicates that the OLDS was closed by IMS and needs to be archived.

ARSCHED
Indicates that the GENJCL.ARCHIVE command has been issued for the OLDS.

ARSTART
Indicates that the Log Archive utility is currently archiving the OLDS.

ARCHIVED
Indicates that the OLDS has been archived and is available for reuse.

AVAIL | UNAVAIL
Mutually exclusive, optional keywords you use to change the PRIOLDS to indicate its availability.

AVAIL
Indicates that the OLDS contains valid data and can be used as input to the Log Archive utility.

UNAVAIL
Indicates that the OLDS contains invalid data and should not be used as input to the Log Archive utility.

DSN(*name***)**
Optional keyword you use to change the name of a primary OLDS. *name* can be up to 44 characters long.

ERROR | NORMAL
Mutually exclusive, optional keywords you use to change the specified PRIOLDS to indicate whether it contains errors.

ERROR
Changes the RECON record to indicate that a specified OLDS contains errors, so IMS is unable to close the OLDS properly. Close the OLDS before it is used as input to the Log Archive utility.

If you use dual logging, the subsystem uses the data in the error-free OLDS (in other words, the SECOLDS) to close the OLDS marked in error.

If you do not use dual logging, the subsystem uses the next-OLDS to close the OLDS that is marked in error.

NORMAL
Changes the record of the PRIOLDS, which was previously marked as containing errors, to indicate that the data set is now available for use as input to any log utility. When you specify NORMAL for an OLDS, the record immediately indicates that neither the secondary OLDS nor the next-OLDS is needed in order to close the specified OLDS.

SSID(*name***)**
Optional keyword you use to specify the name of the IMS subsystem that created the OLDS for which the RECON record is to be changed. The SSID is an eight—character string consisting of any alphanumeric characters that represent a valid IMS subsystem identification name.

If you do not specify SSID, DBRC uses the default subsystem identifier in the RECON header record. Use the INIT.RECON or CHANGE.RECON command to set the default subsystem identifier in the RECON header record. If you have not specified a default in the RECON header record, you must specify SSID.

Example of renaming an OLDS

In this example, the IMS online subsystem, IMSA, that creates the PRIOLDS, renames an OLDS.

```
//CHGPRILG JOB
:
//SYSIN    DD   *
     CHANGE.PRILOG  OLDS(DFSOLP02) -
                    DSN(IMS.NEWLOG) SSID(IMSA)
/*
```

CHANGE.PRILOG RLDS command

Use the CHANGE.PRILOG RLDS command to change information about a primary RLDS (or an SLDS that a batch subsystem created) in the RECON data set.

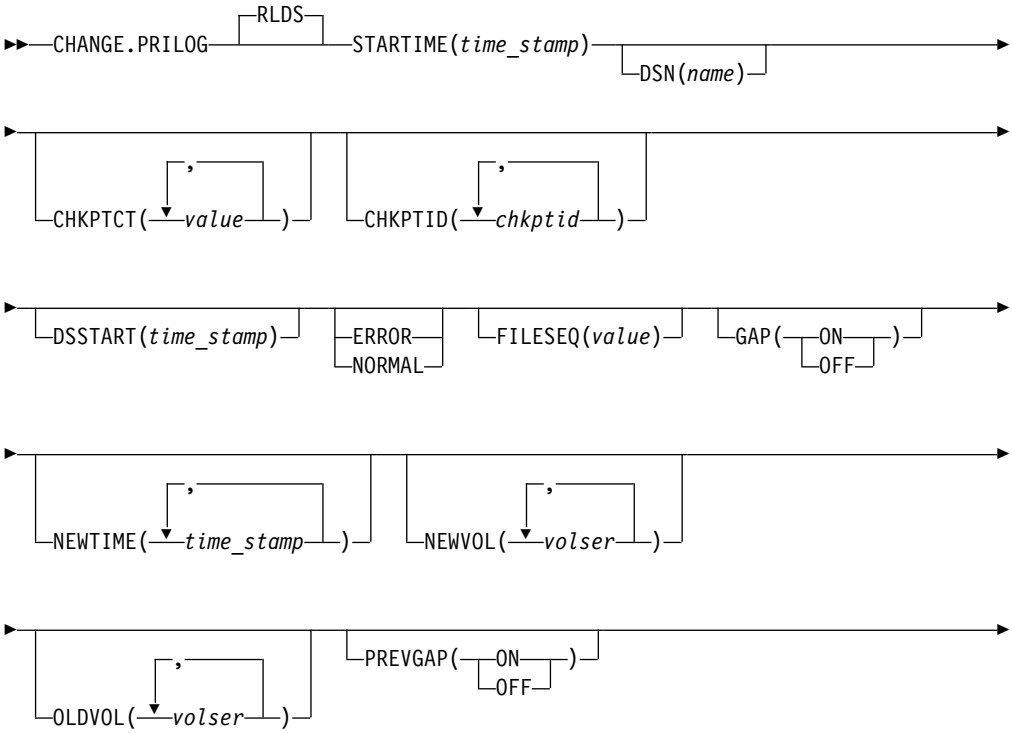
Use the NOTIFY.PRILOG RLDS command to add a PRILOG record or to add data set entries to an existing PRILOG record.

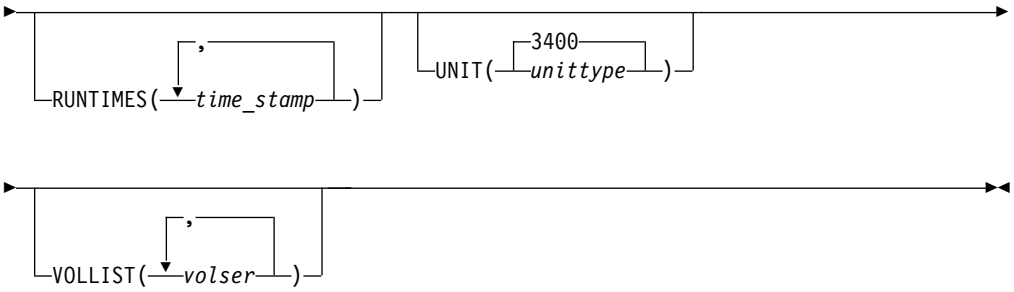
All the information you can change resides in a data set entry of the PRILOG record. Each CHANGE.PRILOG command you issue changes only one data set entry. If the log has multiple data sets, you must use the DSSTART keyword to identify the data set entry to be changed.

Subsections:

- "Syntax"
- "Keywords" on page 110
- "Examples of using the CHANGE.PRILOG RLDS command" on page 113

Syntax





Keywords

RLDS
Optional keyword you use to specify that a PRILOG record is to be changed.

STARTIME(*time_stamp***)**
Required keyword you use to specify the starting time stamp of the PRILOG record that is to be changed. The time stamp must be in standard form.

DSN(*name***)**
Optional keyword you use to change data set name. *name* can be up to 44 characters.

CHKPTCT(*value***)**
Optional keyword you use to change the number of checkpoints completed on each volume of the data set. Specify a value for each volume designated in the OLDVOL or NEWVOL keywords. If OLDVOL is specified without NEWVOL, the number of values for CHKPTCT equals the number of volume serial numbers that appear with OLDVOL. If NEWVOL is specified, the number of values for CHKPTCT equals the number of volume serial numbers that appear in NEWVOL.

The values for CHKPTCT are:

0 No checkpoints on the volume

1 A single checkpoint on the volume

2 More than one checkpoint on the volume

CHKPTID(*chkptid***)**
Optional keyword you use to change the oldest checkpoint ID for any active PST on each volume of the data set. Specify one checkpoint ID for each volume listed in OLDVOL or NEWVOL. If OLDVOL is specified without NEWVOL, the number of checkpoint IDs equals the number of volumes listed in OLDVOL. If NEWVOL is specified, the number of checkpoint IDs equals the number of volumes listed in NEWVOL.

The checkpoint ID must be in standard form for a time stamp. You can specify a zero time value.

DSSTART(*time_stamp***)**
Optional keyword you use to specify the starting time of the data set entry to be changed. The DSSTART keyword is required if the PRILOG has multiple data set entries; it is optional if the PRILOG has only one data set entry. The time stamp must be in standard form.

ERROR | NORMAL
Mutually exclusive, optional keywords you use to change the data set entry to indicate whether it contains errors.

ERROR
Changes the data set to indicate that it contains errors and should not be used as input to any DBRC-controlled run of a recovery utility.

NORMAL
Changes a data set which was previously marked as containing errors to indicate that it is now available for use as input to any recovery utility.

DBRC selects the required log data sets from the PRILOG (or SECLOG) records. These can contain RLDS entries, SLDS entries, or both. If you issue a CHANGE.PRILOG RLDS ERROR command, DBRC automatically uses the corresponding SECLOG entry, if one exists. If a SECLOG entry does not exist, or if it is marked in error, the GENJCL commands that require log data for this time frame fail.

FILESEQ(*value***)**
Optional keyword you use to specify the file sequence number on the volume. Specify this keyword only if you specify a VOLLIST keyword. The value you substitute in the variable field must be a decimal number from 1 to 9999.

GAP(ON | OFF)
Optional keyword you use to set (ON) or reset (OFF) the GAP flag in a tracking PRILOG record.

NEWTIME(*time_stamp***)**
Optional keyword you use to change the stop times of any but the last volume of the data set. If you specify NEWTIME, you must also specify OLDVOL and NEWVOL. The following keyword sets are mutually exclusive:
- NEWTIME, NEWVOL, OLDVOL
- RUNTIMES, VOLLIST

If you specify NEWTIME, you must specify one less time stamp than the number of volume serial numbers specified in NEWVOL. This is because the stop time of the last volume specified in NEWVOL cannot be changed with this command. Each time stamp is used as the volume stop time of the corresponding volume serial number specified by NEWVOL. If not specified, the stop time of the new volume is the same as the stop time of the last-specified old volume.

Each time stamp you specify must be greater than the previous time stamp. The first time stamp in NEWTIME must be greater than or equal to the stop time of the volume immediately preceding the changed volumes. Each time stamp must be in standard form.

NEWVOL(*volser***)**
Optional keyword you use to change the volume serial number of one or more volumes of the data set. If you specify NEWVOL, you must also specify OLDVOL. The following keyword sets are mutually exclusive:
- NEWTIME, NEWVOL, OLDVOL
- RUNTIMES, VOLLIST

The volume serial numbers that you specify in NEWVOL replaces the corresponding volume serial numbers specified in the OLDVOL keyword. You do not need to specify the same number of volume serial numbers in NEWVOL and OLDVOL. You cannot specify a volume serial number in NEWVOL that is the same as one that already exists in the PRILOG record.

You can specify from 1 to 255 volume serial numbers.

Use the NEWTIME keyword to change the time stamps as well as the serial numbers of the volumes.

OLDVOL(*volser*)
Optional keyword you use to change the volume serial number of one or more volumes of the data set. If you specify OLDVOL, you must also specify one of the following: NEWVOL, CHKPTCT, or CHKPTID.

The volume serial numbers you specify are those of the volumes being changed. Each volume serial number specified in OLDVOL must match a volume serial number in the PRILOG record.

You can specify from 1 to 255 volume serial numbers.

PREVGAP(ON | OFF)
Optional keyword you use to set (ON) or reset (OFF) the PREV-GAP flag in a tracking PRILOG record.

RUNTIMES(*time_stamp*)
Optional keyword you use to change the stop times of any but the last volume of the data set. This keyword is provided for compatibility with previous releases of DBRC. Use NEWTIME OLDVOL NEWVOL to change the stop times of log volumes. If you do specify RUNTIMES, you must also specify VOLLIST. The following keyword sets are mutually exclusive:
- NEWTIME, NEWVOL, OLDVOL
- RUNTIMES, VOLLIST

You can specify up to 255 time stamps on the RUNTIMES keyword. Each time stamp must be in standard form.

Each time stamp in the variable field must correspond to a volume in the variable field of the VOLLIST keyword. The variable fields of the RUNTIMES and VOLLIST keywords must each contain the same number of entries. Each time stamp in the variable field of the RUNTIMES keyword must be greater than the previous time stamp.

The first time stamp in the variable field of the RUNTIMES keyword must be greater than the time stamp specified for the STARTIME keyword. The last time stamp in the variable field of the RUNTIMES keyword must be equal to the stop time of the corresponding primary RLDS as specified in the record being changed. You cannot use this command to change the stop time of the primary RLDS.

UNIT(3400 | *unittype*)
Optional keyword you use to change the unit type of the device on which the data set resides. The unit type can be up to 8 alphanumeric characters long.

VOLLIST(*volser*)
Optional keyword you use to change the record of the volume serial numbers of the volumes that contain the data set. This keyword is provided for compatibility with previous releases of DBRC. Use the new keyword set, NEWTIME, OLDVOL, and NEWVOL to change the volume serial numbers of volumes in the data set.

If you specify the VOLLIST keyword, you must also specify the RUNTIMES keyword. See the description of the RUNTIMES keyword for an explanation of how the two keywords interact. The following keyword sets are mutually exclusive:
- NEWTIME, NEWVOL, OLDVOL
- RUNTIMES, VOLLIST

Examples of using the CHANGE.PRILOG RLDS command

Here are some examples of using the CHANGE.PRILOG RLDS command.

Example of changing volume serial numbers

In this example, some volume serial numbers are changed for a log which contains a single data set. The example PRILOG record in the RECON data set has six volumes–VOL001, VOL002, VOL003, VOL004, VOL005, and VOL006– and a start time of 07233124329923456. The serial numbers of the third and fourth volumes can be changed with the following command:

```
//CHGPRILG JOB
⋮
//SYSIN    DD   *
     CHANGE.PRILOG  RLDS STARTIME(07233124329923456)   -
                    OLDVOL(VOL003,VOL004)         -
                    NEWVOL(VOL007,VOL008,VOL009)
/*
```

Example of marking primary RLDS for errors

In this example, one data set of a log is being marked as containing errors.

```
//CHGPRILG JOB
⋮
//SYSIN    DD   *
     CHANGE.PRILOG  RLDS STARTIME(07054121212023456) -
                    DSSTART(07054121212023456) ERROR
/*
```

Related reference:

"NOTIFY.PRILOG command (for RLDS)" on page 359

"DBRC time stamps" on page 52

CHANGE.PRILOG SLDS command

Use the CHANGE.PRILOG SLDS command to change information in the RECON data set about a primary SLDS for an online system.

Use the CHANGE.PRILOG RLDS command to change information about an SLDS that a batch subsystem created, because DBRC considers such data to be an RLDS.

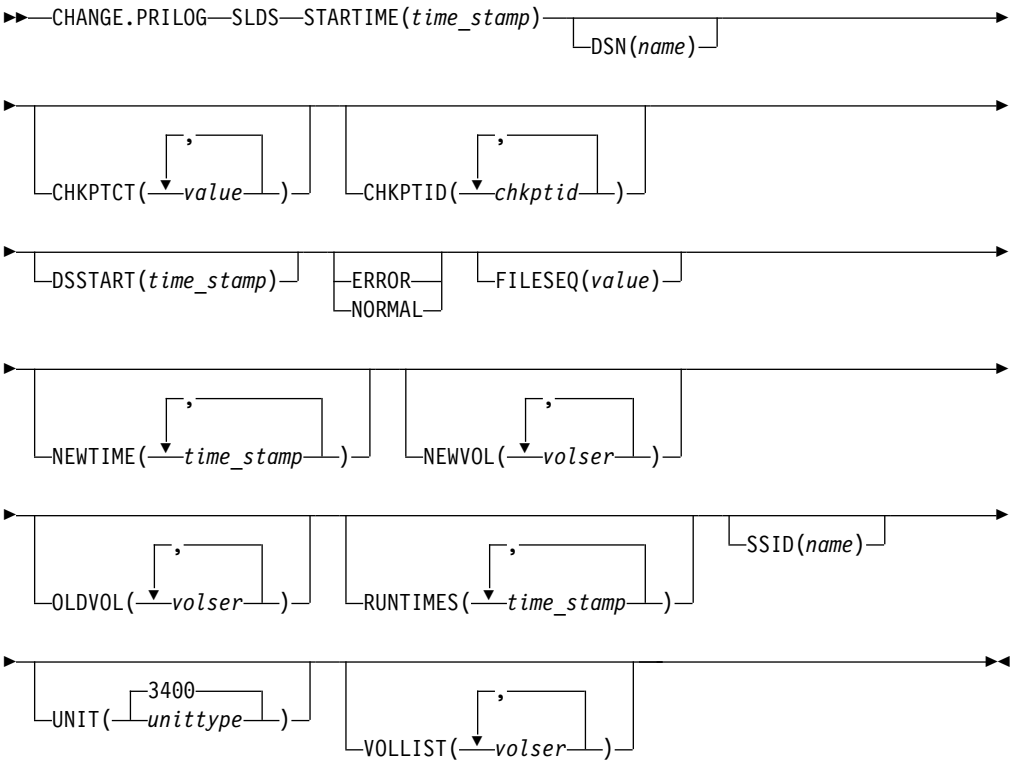
Use the NOTIFY.PRILOG command (for SLDS) to add a PRISLD record or to add data set entries to an existing PRISLD record.

All the information you can change resides in a data set entry of the PRISLD record. Each CHANGE.PRILOG command you issue changes only one data set entry. If the log has multiple data sets, you must use the DSSTART keyword to identify the data set entry to be changed.

Subsections:

- "Syntax" on page 114
- "Keywords" on page 114
- "Examples of using the CHANGE.PRILOG SLDS command" on page 117

Syntax



Attention: For this command you must specify the SLDS keyword. If SLDS is not specified, DBRC will default to processing an RLDS, which will not provide the desired results.

Keywords

The following keywords are valid for the CHANGE.PRILOG SLDS command:

SLDS
Required keyword you use to specify that a PRISLD record is to be changed.

STARTIME(*time_stamp*)
Required keyword you use to specify the starting time stamp of the PRISLD record that is to be changed. The time stamp must be in standard form.

DSN(*name*)
Optional keyword you use to change data set name. *name* can be up to 44 characters.

CHKPTCT(*value*)
Optional keyword you use to change the number of checkpoints completed on each volume of the data set. Specify a value for each volume designated in the OLDVOL or NEWVOL keywords. If OLDVOL is specified without NEWVOL, the number of values for CHKPTCT equals the number of volume serial numbers that appear with OLDVOL. If NEWVOL is specified, the number of values for CHKPTCT equals the number of volume serial numbers that appear in NEWVOL.

The values for CHKPTCT are:

0 No checkpoints on the volume

1 A single checkpoint on the volume

2 More than one checkpoint on the volume

CHKPTID(*chkptid*)
Optional keyword you use to change the oldest checkpoint ID for any active PST on each volume of the data set. Specify one checkpoint ID for each volume listed in OLDVOL or NEWVOL. If OLDVOL is specified without NEWVOL, the number of checkpoint IDs equals the number of volumes listed in OLDVOL. If NEWVOL is specified, the number of checkpoint IDs equals the number of volumes listed in NEWVOL.

The checkpoint ID must be in standard form for a time stamp. You can specify a zero time value.

DSSTART(*time_stamp*)
is a keyword you use to specify the starting time of the data set entry to be changed. The DSSTART keyword is required if the PRISLD has multiple data set entries. The keyword is optional if the PRISLD has only one data set entry. The time stamp must be in standard form.

ERROR | NORMAL
Mutually exclusive, optional keywords you use to change the data set entry to indicate whether it contains errors.

ERROR
Optional keyword you use to change the data set entry to indicate that it contains errors.

NORMAL
Optional keyword you use to change a data set entry which was previously marked as containing errors to indicate that it is normal.

DBRC selects the required log data sets from the PRILOG (or SECLOG) records. These can contain RLDS entries, SLDS entries, or both. If you issue a CHANGE.PRILOG SLDS ERROR command, DBRC automatically uses the corresponding SECLOG entry, if one exists. If a SECLOG entry does not exist, or if it is marked in error, the GENJCL commands that require log data for this time frame fail.

FILESEQ(*value*)
Optional keyword you use to specify the file sequence number on the volume. Specify this keyword only if you specify a VOLLIST keyword. The value you substitute in the variable field must be a decimal number from 1 to 9999.

NEWTIME(*time_stamp*)
Optional keyword you use to change the stop times of any but the last volume of the data set. When you specify NEWTIME, you must also specify NEWVOL and OLDVOL. The following two keyword sets are mutually exclusive:
- NEWTIME, NEWVOL, OLDVOL
- RUNTIMES, VOLLIST

When you specify NEWTIME, you must specify one less time stamp than the number of volume serial numbers specified in NEWVOL. This is because the stop time of the last volume specified in NEWVOL cannot be changed with this command. Each time stamp is used as the volume stop time of the corresponding volume serial number specified by NEWVOL. If not specified, the stop time of the new volume is the same as the stop time of the last—specified old volume.

Each time stamp you specify must be greater than the previous time stamp. The first time stamp in NEWTIME must be greater than or equal to the stop time of the volume prior to the changed volumes. Each time stamp must be in standard form.

NEWVOL(*volser*)
Optional keyword you use to change the volume serial number of one or more volumes of the data set. When you specify NEWVOL, you must also specify OLDVOL. The following keyword sets are mutually exclusive:
- NEWTIME, NEWVOL, OLDVOL
- RUNTIMES, VOLLIST

The volume serial numbers you specify in NEWVOL replace the corresponding volume serial numbers specified in the OLDVOL keyword. You do not need to specify the same number of volume serial numbers in NEWVOL and OLDVOL. You cannot specify a volume serial number in NEWVOL that is the same as one that already exists in the PRISLD record.

You can specify from 1 to 255 volume serial numbers.

Use the NEWTIME keyword if you want to change the time stamps as well as the serial numbers of the volumes.

OLDVOL(*volser*)
Optional keyword you use to change the volume serial number of one or more volumes of the primary SLDS. When you specify OLDVOL, you must also specify one of the following: NEWVOL, CHKPTCT, or CHKPTID.

The volume serial numbers you specify are those of the volumes to be changed. Each volume serial number specified must match a volume serial number in the PRISLD record.

You can specify from 1 to 255 volume serial numbers.

RUNTIMES(*time_stamp*)
Optional keyword you use to change the stop times of any but the last volume of the data set. This keyword is provided for compatibility with previous releases of DBRC. Use the NEWTIME, OLDVOL, and NEWVOL keywords to change the stop times of log volumes. If you do specify RUNTIMES, you must also specify VOLLIST. The following keyword sets are mutually exclusive:
- NEWTIME, NEWVOL, OLDVOL
- RUNTIMES, VOLLIST

You can specify up to 255 time stamps on the RUNTIMES keyword. Each time stamp must be in standard form.

Each time stamp in the variable field must correspond to a volume in the variable field of the VOLLIST keyword. The variable fields of the RUNTIMES and VOLLIST keywords must each contain the same number of entries. Each time stamp in the variable field of the RUNTIMES keyword must be greater than the previous time stamp.

The first time stamp in the variable field of the RUNTIMES keyword must be greater than the time stamp specified for the STARTIME keyword. The last time stamp in the variable field of the RUNTIMES keyword must be equal to the stop time of the corresponding primary SLDS as specified in the record being changed. You cannot use this command to change the stop time of the primary SLDS.

SSID(*name*)
Optional keyword you use to specify the name of the IMS subsystem that created the SLDS for which the RECON record is to be changed.

The SSID is an eight-character string consisting of any alphanumeric characters that describe a valid IMS subsystem identification name. If you do not specify SSID, DBRC uses the default subsystem identifier in the RECON header record. Use the INIT.RECON or CHANGE.RECON command to set the default subsystem identifier in the RECON header record. If you have not specified a default in the RECON header record, you must specify SSID.

UNIT(3400 | *unittype*)
Optional keyword you use to change the unit type of the device on which the data set resides. The unit type can be up to 8 alphanumeric characters long.

VOLLIST(*volser*)
Optional keyword you use to change the record of the volume serial numbers of the volumes that contain the data set. This keyword is provided for compatibility with previous releases of DBRC. Use the NEWTIME, OLDVOL, and NEWVOL keywords to change the volume serial numbers of volumes in the data set.

If you specify the VOLLIST keyword, you must also specify the RUNTIMES keyword. See the above description of the RUNTIMES keyword for an explanation of how the two keywords interact. The following keyword sets are mutually exclusive:
- NEWTIME, NEWVOL, OLDVOL
- RUNTIMES, VOLLIST

Examples of using the CHANGE.PRILOG SLDS command

Here are some examples of using the CHANGE.PRILOG SLDS command.

Example of changing volume serial numbers and stop time

In this example, some volume serial numbers and a volume stop time for a log that contains a single data set are changed. The example PRISLD record in the RECON data set has a start time of 07233124329923456 and six volumes (VOL001, VOL002, VOL003, VOL004, VOL005, and VOL006). The fourth volume has been copied to new volumes VOL007 and VOL008, with the new volume stop time 07233124832523456 for VOL007. The PRISLD record can be updated with the following command:

```
//CHGPRILG JOB
.
.
.
//SYSIN    DD   *
     CHANGE.PRILOG  SLDS STARTIME(07233124329923456) -
                    OLDVOL(VOL004)              -
                    NEWVOL(VOL007,VOL008)       -
                    NEWTIME(07233124832523456)
/*
```

Example of marking primary SLDS as normal

In this example, the first data set of a primary SLDS is marked as normal.

```
//CHGPRILG JOB
.
.
.
//SYSIN    DD   *
     CHANGE.PRILOG  SLDS STARTIME(07054121212023456) -
                    DSSTART(07054121212023456) NORMAL
/*
```

Related reference:

"DBRC time stamps" on page 52

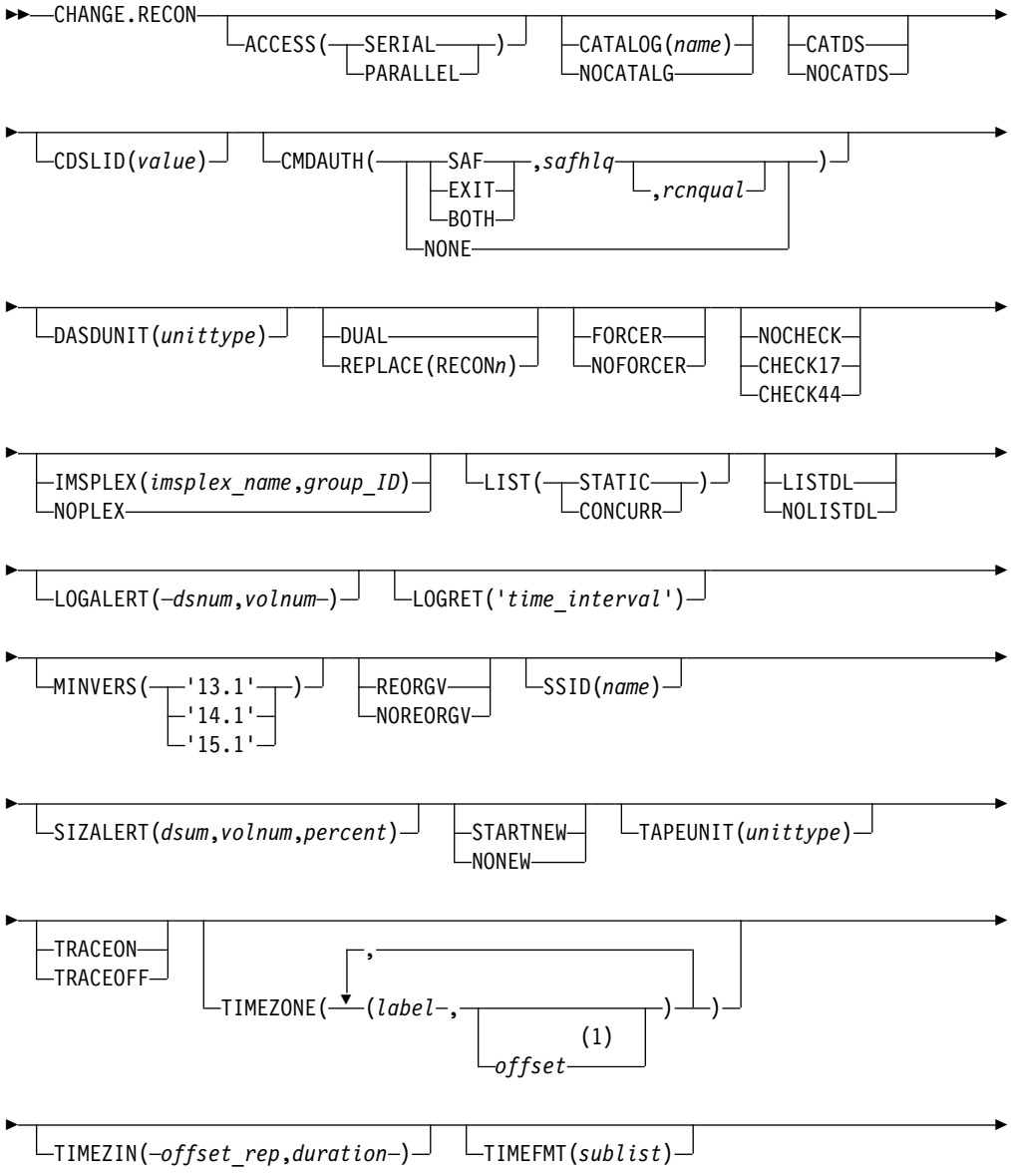
CHANGE.RECON command

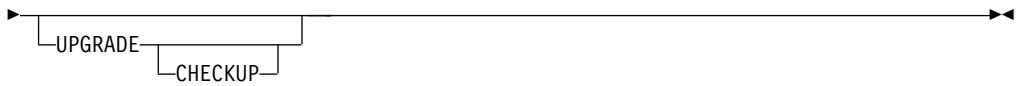
Use the CHANGE.RECON command to update options in the RECON status record.

Subsections:

- "Syntax"
- "Keywords" on page 119
- "Usage notes" on page 131
- "Example of updating the RECON header record" on page 132

Syntax





Notes:

1 The *offset* subparameter of the TIMEZONE keyword must be omitted in order to delete an entry.

Keywords

ACCESS(SERIAL | PARALLEL)
Optional keyword that you use to specify serial or parallel RECON access.

If the ACCESS keyword is specified, SERIAL or PARALLEL must be specified.

SERIAL
Specifies access to a RECON data set in serial mode. Only one DBRC request can be processed at a time. Access to the RECON data set is serialized at the data set level using hardware reserve or Global Resource Serialization (GRS).

PARALLEL
With parallel RECON access, DBRC requests from multiple IMS systems and batch/utility jobs can be processed concurrently. The RECON data set is accessed in Transactional VSAM mode. Access is serialized at the VSAM record level using coupling facility locking.

To use parallel RECON access, the following conditions must be met:

- All three RECON data sets must be available. There must be two active RECON data sets, COPY1 and COPY2, and a spare (or discarded) RECON data set.
- The RECON data set must be associated with an IMSplex.

If the ACCESS keyword specifies the access mode that is currently set for the RECON data set, the command sets return code 4.

Restriction: The ACCESS(SERIAL | PARALLEL) keyword cannot be changed in the same command when IMSPLEX() | NOPLEX keywords are being changed.

CATALOG | NOCATALG
Mutually exclusive, optional parameters that you use to indicate whether DBRC retrieves the database definitions from the IMS catalog or the DBDLIB data set when the following commands are run:

CHANGE.DBDS

CHANGE.PART

For HALDBs only, INIT.DB

INIT.DBDS

INIT.PART

NOTIFY.REORG

For databases that are defined using Database Definition Language (DDL), DBRC must access an IMS catalog.

CATALOG(*name*)
Specifies that an IMS catalog is used.

name is an 8-character alphanumeric string that identifies a HALDB database that is registered in the RECON data set. Any catalog name that is less than 8 characters is padded with blanks.

The catalog name is validated when the CHANGE.RECON command is run.

NOCATALG
Specifies that the DBDLIB data set is used.

CATDS | NOCATDS
Mutually exclusive, optional keywords you use to modify the status of whether image copy, change accumulation, and log data sets are cataloged.

CATDS
Specifies that these data sets are cataloged or SMS-managed.

DBRC does not verify the volume serial and file sequence numbers for the data set.

For the CATDS option to be effective, the data set must be cataloged, and VOLSER information for the data set must be omitted from the JCL. If the data set is cataloged, CATDS is specified, and VOLSER information is included in the JCL, DBRC ignores CATDS and allocates the data set by the JCL. Normal VOLSER and file sequence checking occurs.

If the data set is not cataloged, CATDS is not effective, and DBRC allocates the data set by the JCL, with VOLSER and file sequence checking.

If log data sets are SMS-managed, select the CATDS option and remove the %LOGVOLS keyword from skeletal JCL member CAJCL.

Attention: The CATDS option affects restart of IMS from SLDS data sets. Because the CATDS option indicates the SLDS are under the control of a catalog management system, the VOLSER is not passed back to IMS for data set allocation. If the SLDS data sets are not cataloged, IMS restart fails.

NOCATDS
Specifies that these data sets, regardless of their cataloged status, are not to be treated as cataloged.

DBRC checks the data set name and verifies that the volume serial and file sequence numbers specified in the JCL are the same as the information recorded in the RECON data set.

CDSLID(*value*)
Optional keyword that you use to set the cross-DBRC service level ID in the RECON data set. The service level ID indicates that the maintenance that corresponds to the given value has been applied to all systems that will be accessing this RECON data set.

Occasionally, maintenance must be applied to all systems that access the same RECON data set before you enable that maintenance. The ++HOLD card associated with the maintenance indicates the need to issue the CHANGE.RECON CDSLID() command.

This command is used to enable the function provided by the maintenance after the maintenance is applied to all the systems. The value to specify on the command is indicated in the ++HOLD card.

After the cross-DBRC service level ID is set, all maintenance related to any service level ID less than or equal to the value set will be activated. If, after enabling some maintenance, you find that a system does not yet have the

maintenance applied, to deactivate the enabled function specify a value less than the enabling service level ID. For example, if service level ID 5 is set in the RECON data set, issuing a CHANGE.RECON CDSLID(3) command will deactivate the functions associated with IDs 4 and 5.

CMDAUTH()
Optional keyword you use to specify whether DBRC command authorization is to be performed.

Specification of this keyword will result in command authorization being performed using the specified command authorization setting. However, in order to turn off command authorization (specifying NONE), you must be authorized under the current command authorization setting.

SAF | EXIT | BOTH | NONE
Mutually exclusive subparameters you use to specify your DBRC authorization option.

SAF
Specifies that a security product is to be called for command authorization.

EXIT
Specifies that the DBRC Command Authorization Exit (DSPDCAX0) is to be called for command authorization.

BOTH
Specifies that a security product and the DBRC Command Authorization Exit (DSPDCAX0) are to be called for command authorization. The security product is called first, then DSPDCAX0.

NONE
Specifies that command authorization is to be skipped.

safhlq
High level qualifier of the resource name defined to the z/OS System Authorization Facility (SAF) product (RACF) to define command authorization. *safhlq* must be specified with SAF, EXIT, or BOTH. *safhlq* is a name up to 8 alphanumeric characters in length.

Note: If you simply want to change your high level qualifier, specify your current command authorization option with the new *safhlq*. The profiles with the new qualifier must exist before entering the command. Create the new profiles or rename the old ones.

rcnqual
Optional parameter you use for setting or changing a string value in the RECON data set that will be used in DBRC command processing on the RECON to determine whether the data set is a copy and whether command authorization should be enforced. The *rcnqual* parameter can be specified only with SAF, EXIT, or BOTH.

When setting or changing the *rcnqual* value, the string must be a substring of the RECON COPY1 data set name in order to prevent inadvertently disabling security for the production RECON data set. If *rcnqual* is specified with an asterisk (*) at the end and within quotation marks, the RECON COPY1 data set name must start with *rcnqual* or DBRC command authorization will not be enforced.

For command authorization to be enforced the *rcnqual* must be a substring of the RECON COPY1 data set name. Ideally it should be a substring

common to all three RECON data set names in order to prevent any problems if RECON reconfiguration takes place. *rcnqual* can be up to 44 characters long and must follow DBRC data set naming conventions.

DASDUNIT(*unittype***)**
Optional keyword you use to change the unit type of the DASD device that holds the records for log data sets. The unit type can be up to 8 alphanumeric characters long.

DBRC stores the value specified for DASDUNIT in the unit field of various RECON records. Multiple utilities use this value as the default value for the output device type.

DUAL | REPLACE(RECON*n***)**
Mutually exclusive, optional keywords you use to reestablish dual mode or to replace an active RECON data set with the spare RECON data set:

DUAL
Causes DBRC to enter dual-RECON mode. If DBRC is already using two RECON data sets, the dual keyword is ignored. If DBRC is using one RECON data set, it attempts to use a spare RECON data set. If no spare RECON data set is available, dual mode is not entered; however, any other optional keywords are processed.

You are not required to use the DUAL keyword to cause DBRC to enter dual-RECON mode. If as a result of a permanent I/O error on a RECON data set, for example, DBRC is reduced to the use of a single RECON data set, it automatically reenters dual-RECON mode as soon as it becomes aware of the existence of a spare RECON data set. However, in installations that use DBRC for log control only, it can be some time before DBRC becomes aware of a recently created spare RECON data set. Use the CHANGE.RECON command with the DUAL keyword to cause DBRC to enter dual-RECON mode immediately.

REPLACE
Causes DBRC to replace an active RECON data set with a spare RECON data set. When you specify this keyword, you can reorganize the RECON data sets online.

For RECON*n*, specify the DD statement of the RECON data set you want replaced. For *n*, you can specify 1, 2, or 3. If you specify a RECON data set that is not active or if no spare RECON data set is available, the replace does not take place; however, any other optional keywords that are specified on the command are executed.

FORCER | NOFORCER
Mutually exclusive, optional keywords you use to specify whether all databases must be registered in the RECON data set.

FORCER
Specifies that all databases must be registered in the RECON data set. If a job tries to access an unregistered database, the database authorization call from IMS to DBRC fails.

NOFORCER
Specifies that databases do not have to be registered in the RECON data set.

DBRC checks this keyword during initialization and it remains in effect for as long as the subsystem runs. If you change this keyword while the control

region is active, the change does not take effect until restart or initialization, although the change appears in a listing of the RECON data set.

NOCHECK | CHECK17 | CHECK44
Mutually exclusive, optional keywords you use to change the type of comparison of log data set names that is done by DBRC.

NOCHECK
Specifies that the data set name you specify as input to DBRC has a new high-level qualifier and is longer than 17 characters. With NOCHECK, DBRC does not compare the log data set name that is recorded in the RECON data set with the name on the appropriate DD statement.

CHECK17
Verifies that the last 17 characters of a log data set name are consistent with the RECON data set. If the name in the RECON data set does not match the name on the appropriate DD statement, the utility stops.

CHECK44
Verifies that the 44-character log data set name is consistent with the RECON data set. If the name in the RECON data set does not match the name on the appropriate log DD statement, the utility stops.

IMSPLEX(*imsplex_name, group_ID*) | NOPLEX
Mutually exclusive, optional keywords you use to change the IMSplex name and the DBRC group ID associated with the RECON data set.

No other DBRC instances that are registered to SCI can access the RECON data set when this command is processed. The DBRC instance processing the command must use the IMSplex name that is currently saved in the RECON data set.

The IMSplex name in the RECON data set can be changed only when your SCI registration exit, DSPSCIX0, is changed to return the new IMSplex name.

The DBRC group ID can be used to uniquely identify different RECON data sets that are used in the same IMSplex. If you use parallel RECON access, each RECON in the same IMSplex must have a unique group ID.

Restrictions:

- CHANGE.RECON IMSPLEX and CHANGE.RECON NOPLEX can only be issued from the batch command utility. You cannot issue these commands from an online IMS.
- CHANGE.RECON IMSPLEX and CHANGE.RECON NOPLEX commands cannot be issued as /RMxxx commands. They can be issued embedded in the Database Recovery Control utility as batch commands or embedded in a FUNC=COMMAND DBRC API request.
- The ACCESS(SERIAL|PARALLEL) keyword cannot be changed in the same command when IMSPLEX()|NOPLEX keywords are being changed.
- CHANGE.RECON NOPLEX is not allowed if the current access mode for the RECON data set is parallel.

IMSPLEX(*imsplex_name*, *group_ID*)
Specifies the new IMSplex name. The IMSplex name can be up to 5 alphanumeric characters long.

imsplex_name
The *imsplex_name* can be up to 5 alphanumeric characters long.

group_ID
The *group_ID* can be up to 3 alphanumeric characters long. The

imsplex_name must be specified if *group_ID* is specified. The *group_ID* will be set to '001' if not specified and no value is already set.

NOPLEX
Specifies that no IMSplex name is to be associated with the RECON data set.

There is no default for this keyword. If neither is specified, the current setting is unchanged.

Note: If DSPSCIX0 is not found, DBRC behaves as though the sample exit were used. So, if the IMSPLEX keyword is not used, DBRC assumes that RECON Automatic Loss Notification is inactive. When the IMSPLEX keyword is used, DBRC will use its value to register with SCI.

LIST(STATIC | CONCURR)
Optional keyword to set a default of STATIC or CONCURR for the LIST.xxx commands

STATIC
Specifies that if the RECON data set is in parallel access mode and a LIST command is processed that did not specify the STATIC or CONCURR keyword, the default for list processing is STATIC.

CONCURR
Specifies that if the RECON data set is in parallel access mode and a LIST command is processed that did not specify the STATIC or CONCURR keyword, the default for list processing is CONCURR.

Recommendation: Because some LIST commands with the STATIC option are likely to be disruptive, set CONCURR as the default. STATIC can then be specified on specific commands, for example, when trying to resolve problems. (The default, if not explicitly set by a CHANGE.RECON command or the INIT.RECON command, is STATIC rather than CONCURR only because STATIC is what LIST command processing with serial RECON access produces.)

LISTDL | NOLISTDL
Mutually exclusive, optional keywords you use to specify whether data set names that are deleted from the RECON (by the DELETE.LOG command or by an archive job log compression) are listed in the job output. The setting specified on this command can be overridden by the DELETE.LOG command. There is no way to override the setting for log compression during an archive job.

LISTDL
Specifies that deleted data set names are to be listed in the job output.

NOLISTDL
Specifies that deleted data set names are not to be listed in the job output.

There is no default for this keyword. If neither is specified, the current setting is unchanged.

LOGALERT(*dsnum*,*volnum*)
Optional keyword you use to define the threshold that triggers the DSP0287W message. Message DSP0287W displays when you just have time to shut down an online IMS subsystem before it terminates abnormally because the PRILOG record size exceeds the 16M record size maximum.

Given the 16 MB maximum allowable size of a PRILOG record, for practical purposes, LOGALERT is obsolete.

dsnum,volnum
These values apply only to PRILOG-family records. The message is issued when both of the following conditions are true:

- A new OLDS data set opens.
- When there will no longer be room in the PRILOG record to successfully archive all OLDSs currently needing to be archived (including the new one) plus *dsnum* more, assuming each OLDS uses *volnum* volumes.

The values that you enter, based on your knowledge of the rate at which the subsystem normally fills OLDSs, should be calculated to give you sufficient time to effect a normal shutdown of the online IMS subsystem.

All values must be supplied. A zero (0) in any position means that the existing value in the RECON record is not to be changed.

The default values in a new RECON or one that has been upgraded from an earlier release are **(3,16)**, and are set during INIT.RECON command processing.

LOGRET(*time_interval*)
Optional keyword you use to change the retention period for log data sets.

Definitions:

- The *retention period* is the minimum amount of time in which a log becomes inactive after it is opened. (It is then eligible to be deleted.)
- The *time_interval* is a partial, punctuated time stamp representing a time interval (days, hours, minutes, seconds and tenths of a second) instead of date and time. The time stamp for this command follows the standard time stamp format except that the year subparameter element is omitted. Valid intervals range from a tenth of a second to 365 days.

  Because the time interval is treated as a time stamp, message DSP1048I can be issued for incorrect values. Some examples of valid time intervals include:

```
LOGRET(365)
LOGRET('030 12.00')
LOGRET('000 00:00:08.0')
LOGRET('000 00,00,00,1')
```

The following shows two different formats for equivalent time stamp specifications. Both are valid.

```
LOGRET(030)          LOGRET('030')        = 30 days
LOGRET('010 12,30')  LOGRET('010 12:30')  x= 10 days, 12 hours, 30 minutes
```

For information about inactive logs and how they are determined by use of the log retention period and other conditions, see the INACTIVE keyword under the DELETE.LOG (for RLDS and SLDS) command.

Recommendation: Ensure that an appropriate log retention period is specified so that log information is kept in the RECON data set for a long enough period of time. Certain IMS functions depend on log information in the RECON data set, and these functions might fail if that information is deleted too soon. For example, one or more of the following functions might fail:

- Dynamic backout
- Batch message processing (BMP) extended restart (XRST)
- Unit of work table entry (UOWE) resynchronization

MINVERS('13.1' | '14.1' | '15.1')
Optional keyword that specifies the lowest version of IMS that is allowed to access the RECON data sets. This keyword cannot be set to a higher IMS release if any subsystems with a lower version of IMS are currently signed on to DBRC.

Precision for all time stamp formats is set to the microsecond.

The format for the MINVERS keyword includes a period. DBRC considers this a special character and requires the value to be enclosed in single quotation marks.

Requirements:

- To run the DEDB Alter utility with the ALTERAREA function for a DEDB area with sequential dependent (SDEP) segments, the MINVERS value must be 14.1.
- To specify that an HALDB database can have up to 8 gigabytes of data in an OSAM PHDAM or PHIDAM database data set, the MINVERS value must be 14.1 or it must be 13.1 with a CDSLID value of 2. **MINVERS** cannot be lowered to '13.1' if an HALDB database is defined as using 8-GB OSAM data sets and the CDSLID value is less than 2. A CDSLID value of 2 indicates that the SPE APAR that enables the support of 8-GB OSAM data sets has been applied to all IMS Version 13 resident libraries that access the RECON data set.

REORGV | NOREORGV
Mutually exclusive, optional parameters you use to modify the HALDB partition reorganization number verification. A CHANGE command with these parameters is allowed while databases are currently authorized to other subsystems.

REORGV
Specifies that HALDB partition reorganization number verification is enabled. The RECON stores the reorganization number for each partition and prevents the reorganization number from being regressed. After a CHANGE.RECON REORGV command is issued, the partition reorganization number in each partition record is set to zeros.

NOREORGV
Specifies that HALDB partition reorganization number verification is disabled. Any requests to store, update or verify the reorganization number will be ignored.

SSID(*name*)
Optional keyword you use to change the name of the IMS subsystem to be used as the subsystem ID for the following commands:

- CHANGE.PRILOG
- CHANGE.SECLOG
- DELETE.LOG
- GENJCL.ARCHIVE
- GENJCL.CLOSE
- NOTIFY.PRILOG
- NOTIFY.SECLOG

The SSID is an 8-character string of any alphanumeric characters that comprise a valid IMS subsystem identification name.

SIZALERT(*dsnum,volnum,percent*)
Optional keyword you use to define thresholds that trigger messages to warn you that a record has grown unusually large. The decimal threshold values that you supply for SIZALERT are:

dsnum,volnum
These values apply only to PRILOG-family records. The message DSP0387W is issued when both of the following conditions are true:

- When a new OLDS data set opens.
- All currently open OLDSs, including the new one, have been archived, there will no longer be room in the record for *dsnum* data set entries of *volnum* volumes each, or the record size will exceed *percent* percent of the 16 MB record size maximum.

Given the 16 MB maximum allowable size of a PRILOG record, for practical purposes, *dsnum* and *volnum* are obsolete.

percent
This value applies to all records. The threshold is reached when a record exceeding *percent* percentage of the 16 MB record size maximum.

STARTNEW | NONEW
Mutually exclusive, optional keywords you use to specify whether new jobs are to be started when only one RECON data set is available.

STARTNEW
Specifies that new jobs are to be started. If parallel RECON access is enabled, new jobs cannot start with a single available RECON data set if another DBRC instance is running with two active RECON data sets.

NONEW
Specifies that new jobs are not to be started.

TAPEUNIT(*unittype*)
Optional keyword you use to change the unit type of the tape device that holds the records for log data sets. The unit type can be up to 8 alphanumeric characters long.

DBRC stores the value specified for TAPEUNIT in the unit field of various RECON records. Multiple utilities use this value as the default value for the output device type.

TRACEON | TRACEOFF
Mutually exclusive, optional keywords you use to specify whether to start or stop external tracing.

Note: If the CHANGE.RECON command is issued in a batch job, the RECON header record will be changed to show the requested status (TRACEON or TRACEOFF), however, the actual TRACE status will not be changed for any active IMS system.

If the CHANGE.RECON command is issued as a command in a control region (/RMCHANGE DBRC='RECON TRACEON | TRACEOFF'), the trace status and the RECON header record will be changed to show the requested status (TRACEON or TRACEOFF) for this control region. However, the actual TRACE status will not be changed for any other active IMS system (batch or online).

TRACEON
Starts external tracing. If you specify this keyword, the specified Generalized Trace Facility (GTF) must be active for USR-type records.

TRACEOFF
Stops external tracing. If you specify this keyword, DBRC only does internal tracing.

TIMEZONE((*label*,*offset*),(*label*,*offset*))
Optional keyword that alters the time zone label table. This keyword is used to define one or more symbolic time zone labels. Because most people do not readily associate a numeric offset with a time zone, TIMEZONE enables you to define symbolic labels, like PST (Pacific Standard Time), for numeric offsets, such as -8.

The time zone label table can contain up to 32 entries, each of which is composed of a label and an offset.

label
An alphanumeric value of up to 5 characters, the first of which must be alphabetic. Lowercase characters are translated to uppercase.

offset
A signed-decimal value in the form of ± [h]h[:mm] that meets the requirements of a valid time stamp offset. The offset is the value that, when added to UTC, gives local time. For example, the value to use for PST (Pacific Standard Time) is -8. The value for JST (Japan Standard Time) is +9.

Adding, replacing, and deleting entries from the stored list is supported as follows:

- Adding an entry to the stored table is accomplished when an input list entry contains both a label that does not exist in the RECON data set and a valid offset value.
- Replacing an entry to the stored table is accomplished when the input entry contains both a label that matches an existing label in the table and a valid offset value.
- Deleting an entry to the stored table is accomplished when the input entry is a label that matches an existing label in the table and no offset value was specified. If the offset is omitted, and the label is not found in the table, the table is not altered.

The labels in the table must be unique.

Suggestions for time zone label table management: The same offset should not be designated by more than one label because DBRC always uses the first occurrence in the table when outputting a time stamp.

The practicality of using the label format is affected by the scope of the IMS installation. For those operating solely in a single time zone, use of labels eliminates the need for the operator to know the exact offset to UTC at all times during the year. For multiple time zone operation, the use of offsets rather than labels, is suggested (though not mandatory). The time zone label table may not be practical if offsets are not unique from one zone to the next when daylight saving time adjustments are taken into account. Changing the table when daylight saving time switches are made would add to the confusion, so, in that case, use numeric offset values for cross time zone operation.

TIMEZIN(*offset_rep* **[,***duration***])**
Optional keyword you use to define a default time zone value for time stamps that are entered without time zone information on subsequent DBRC commands.

offset_rep
The default time zone value. It can be one of the following choices:

%SYS
A keyword used to designate that the offset is to be derived from the current offset found in the z/OS CVT control block. This is the initial default for DBRC.

label
A time zone label that has been previously defined using the TIMEZONE keyword.

offset
A numeric offset value in the same form as defined above for the TIMEZONE keyword.

duration
Specifies the duration of the *offset_rep* choice.

TEMP
Indicates that the label or offset default is in effect only for the job in which the command is entered.

PERM
Indicates that the label or offset default is to be in effect for any subsequent DBRC command running with the same RECON data set.

TIMEFMT(*sublist*)
Optional keyword you use to define the form in which time stamps appear from DBRC in messages, displays, and listings. The five values are positional. Each is optional and can be omitted by including only the comma.

UPGRADE
Optional keyword that you use to upgrade all RECON records to an IMS 15 level. You do not need to shut down other IMS subsystems that share the RECON data set, although they might have to wait for the command to complete.

Restriction: You can issue the CHANGE.RECON UPGRADE command using the Database Recovery Control utility (DSPURX00) or the DBRC application programming interface. You cannot issue it from an online IMS.

Attention: Jobs such as utility programs (for example, DSPURX00), which do not create subsystem records and which access the RECON data set, are not protected from having the RECON data set upgraded while they are running an IMS release without the appropriate IMS 15 SPE applied. Their next RECON access might have unpredictable results. Ensure that no such jobs are active when entering the CHANGE.RECON UPGRADE command.

Recommendations:

- Before running the CHANGE.RECON UPGRADE command against the production RECON data sets, upgrade a copy of the production RECON data set to verify that the upgrade will complete successfully. A

CHANGE.RECON UPGRADE CHECKUP command can also be run against a copy of the production RECON data set or against the production RECON data set.

- Large numbers of CA/CI splits in the RECON data sets impact performance with so many RECON records increasing in size. Maintain the RECON data sets to delete records that are no longer needed. Ensure that secondary extents are available so that the RECON data sets do not run out of space during the upgrade. Upgrade the RECON at a time when there is minimal activity against the RECON data sets. Upgrade a backup copy of the production RECON as a test to determine how long the upgrade will take at your site.

The CHANGE.RECON UPGRADE command upgrades the RECON data set in two stages: first, Copy 1, and then Copy 2.

Message DSP0251I indicates the beginning of the first stage. A failure during this stage for RECON data sets that are using serial access causes DBRC to reconfigure the RECON data sets, discarding Copy 1. If a failure occurs during the first stage with serial access, you need to rerun the CHANGE.RECON UPGRADE command. A failure during the first stage for RECON data sets that are using parallel access causes DBRC to back out any changes, and the command is retried.

After successful completion of the first stage, message DSP0252I indicates that the second stage is beginning. Any failure during this stage causes DBRC to reconfigure the RECON data sets, discarding Copy 2. However, if there is a failure during the second stage, you do not need to rerun the CHANGE.RECON UPGRADE command.

You can use the CHANGE.RECON UPGRADE command in either nonconcurrent or concurrent mode. The mode is determined by the environment at the time you issue the command.

Nonconcurrent mode
No other jobs are currently accessing the RECON data set.

Before issuing the CHANGE.RECON UPGRADE command, create a backup copy of the RECON data set in case the upgrade and the RECON recovery procedures fail. The procedures can fail if you have not allocated two RECON data sets and a spare.

Concurrent mode
Other jobs are currently accessing the RECON data set. These jobs can be either and must have the respective IMS 15 SPE applied. Message DSP1113A is issued for any subsystem that does not have the appropriate coexistence SPE applied. Two RECON data sets plus a spare must be allocated for a concurrent upgrade.

CHECKUP
Optional keyword that you use with the UPGRADE keyword to check the status of the RECON data set and records to verify if the RECON data set is in a state that allows upgrade. No RECON data set records are changed as a result of executing the CHANGE.RECON UPGRADE CHECKUP command. After upgrading a backup copy of the production RECON data sets, this command can be used prior to actually upgrading the production RECON data sets to ensure conditions have not changed that would prevent the RECON data set from being upgraded. You do not need to shut down IMS subsystems that share the RECON data set, although they might have to wait for the command to complete.

Recommendation: Run a CHANGE.RECON UPGRADE CHECKUP command against a copy of the production RECON data set prior to issuing the command against the production RECON data set to understand any possible performance considerations.

When a CHANGE.RECON UPGRADE CHECKUP command is started, message DSP1238I is issued. If the command completes successfully with no errors encountered that would prevent upgrade, message DSP1239I is issued and return code 0 is set. If the command completes successfully and errors are encountered that would prevent an upgrade, message DSP1240E is issued and return code 12 is set.

Additional messages are issued to indicate any errors found, such as DSP1236E or DSP1113A.

Since RECON data set records are not upgraded when the CHECKUP keyword is included in the CHANGE.RECON UPGRADE command, upgrading the RECON data sets later could fail for other reasons, such as running out of space in the RECON data set.

A CHANGE.RECON UPGRADE CHECKUP command should be run against a copy of the production RECON data set prior to issuing the command against the production RECON data set to understand any possible performance considerations.

Do not specify the UPGRADE and CHECKUP keywords with any other keywords on this command. If a CHANGE.RECON UPGRADE CHECKUP command includes additional keywords, the UPGRADE and CHECKUP keywords are processed first. The RECON data sets are not upgraded. If any additional keywords are included, the attempt to open the RECON data sets that are not upgraded fails.

Usage notes

If parallel RECON access is used, RECON activity for all DBRC instances is quiesced. A check for the existence of retained locks related to the Copy1 or Copy2 of the RECON data sets is completed before RECON upgrade is invoked. If any retained locks exist, message DSP0383A is issued and the RECON upgrade fails.

In parallel RECON access mode, any retained locks that exist for the RECON data sets must be resolved. Ensure that the SMS server (SMSVSAM) is active and Transactional VSAM (TVS) is active on all processors and resubmit the command.

DBRC accesses the database definitions stored in the DBDLIB while processing the NOTFY.REORG, CHANGE.PART, CHANGE.DBDS, INIT.PART, INIT.DB and INIT.DBDS commands. DBRC needs to access the IMS catalog when the directory is enabled instead of the DBDLIB because it may not contain the database definitions for a database defined with DDL.

The RECON contains a new catalog name. This name will be defined indicating the DBRC is using the directory instead of the DBDLIB for commands that need to reference the database definition. This catalog name can be set on the CHANGE.RECON command as the default. The catalog name is not required; if it is not present then DBRC will continue to expect the IMS DD to contain the DBDLIB.

Example of updating the RECON header record

Here are some examples of using the CHANGE.RECON command.

Example 1 for CHANGE.RECON command

```
//CHGRECON JOB
⋮
//SYSIN    DD   *
     CHANGE.RECON  SSID(IMSB) FORCER LOGRET('007')
/*
```

In this example, you are forcing all databases to be registered, changing the default subsystem ID to IMSB, and changing the log retention period to 7 days.

Example 2 for CHANGE.RECON command

```
//CHGRECON JOB
⋮
//SYSIN    DD   *
     CHANGE.RECON CMDAUTH(SAF,SAFHLQ1,IMSTESTS.DSHR)
/*
```

In this example, if the RECON COPY1 data set name is IMSTESTS.DSHR.RECON1, DBRC command authorization security will remain enforced for any future access attempts on the RECON data set, because the *rcnqual* value of "IMSTESTS.DSHR" is a substring of the RECON COPY1 data set name IMSTESTS.DSHR.RECON1.

If you copy this RECON data set into a data set whose name has COPYDSHR as the second level qualifier, DBRC command authorization security will be disabled for any future access attempts on the RECON data set, because when the copy is made, the RECON COPY1 data set name is changed to IMSTESTS.COPYDSHR.RECON1, which no longer contains the *rcnqual* string value of "IMSTESTS.DSHR".

Example 3 for CHANGE.RECON command

```
//CHNGRCON JOB
⋮
//RECON1   DD  DSN=ILOVE.IMS.SAIDPAT.RECON1,DISP=SHR
//RECON2   DD  DSN=ILOVE.IMS.SAIDPAT.RECON2,DISP=SHR
//SYSIN    DD   *
     CHANGE.RECON  CATALOG(DFSCD000)
```

Related concepts:

Repairing the RECON data sets (System Administration)

Related reference:

"DELETE.LOG command (for RLDS and SLDS)" on page 169

"DBRC time stamps" on page 52

CHANGE.SECLOG OLDS command

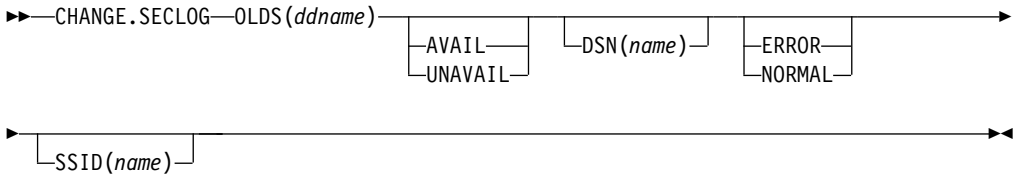
Use the CHANGE.SECLOG OLDS command to change information about a secondary online data set (SECOLDS) in the RECON data set.

Subsections:

• "Syntax" on page 133

- "Keywords"
- "Example showing a SECOLDS error" on page 134

Syntax



Attention: For this command you must specify the OLDS keyword. If OLDS is not specified, DBRC will default to processing an RLDS, which will not provide the desired results.

Keywords

OLDS(*ddname***)**
Required keyword you use to specify the OLDS for which the RECON record is to be changed. Failure to specify this keyword results in an RLDS being changed.

AVAIL | UNAVAIL
Mutually exclusive, optional keywords you use to change the SECOLDS to indicate its availability.

AVAIL
Indicates that the OLDS contains valid data and that it can be used as input to the Log Archive utility.

UNAVAIL
Indicates that the OLDS contains invalid data and it should not be used as input to the Log Archive utility.

DSN(*name***)**
Optional keyword you use to change the name of a secondary OLDS. The name you substitute in the variable field can be up to 44 characters long.

ERROR | NORMAL
Mutually exclusive, optional keywords you use to change the specified SECOLDS record to indicate whether it contains errors.

ERROR
Changes the RECON record to indicate that a specified OLDS contains errors, so IMS is unable to close the OLDS properly. The OLDS must be closed before it can be used as input to the Log Archive utility.

When you use dual logging, you use ERROR to change a specified SECOLDS record to indicate that it contains errors. The subsystem uses the data in the error-free OLDS to close the OLDS that is marked ERROR.

NORMAL
Changes the SECOLDS record, which was previously marked as containing errors, to indicate that the data set is now available for use as input to any log utility. When you specify NORMAL for a secondary OLDS, the record immediately indicates that the next primary OLDS is no longer needed in order to close the corresponding primary OLDS.

SSID(*name*)
Optional keyword you use to specify the name of the IMS subsystem that created the OLDS for which the RECON record is to be changed.

The SSID is an 8-character string consisting of any alphanumeric characters that comprise a valid IMS subsystem identification name. If you do not specify SSID, DBRC uses the default subsystem identifier in the RECON header record. Use the INIT.RECON or CHANGE.RECON command to set the default subsystem identifier in the RECON header record. If you have not specified a default in the RECON header record, you must specify SSID.

Example showing a SECOLDS error

In this example, a SECOLDS that IMS online subsystem IMSA created is known to be in error.

```
//CHGSECLG JOB
⋮
//SYSIN    DD   *
     CHANGE.SECLOG  OLDS(DFSOLS02)          -
                    SSID(IMSA) ERROR
/*
```

CHANGE.SECLOG RLDS command

Use the CHANGE.SECLOG RLDS command to change information about a primary RLDS (or an SLDS that a batch subsystem created) in the RECON data set.

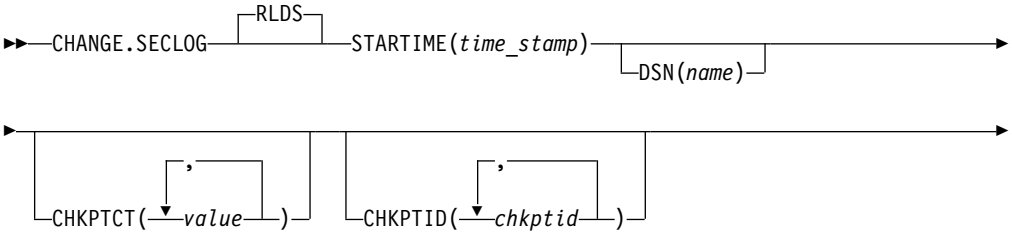
Use the NOTIFY.SECLOG command (for RLDS) to add a SECLOG record or to add data set entries to an existing SECLOG record.

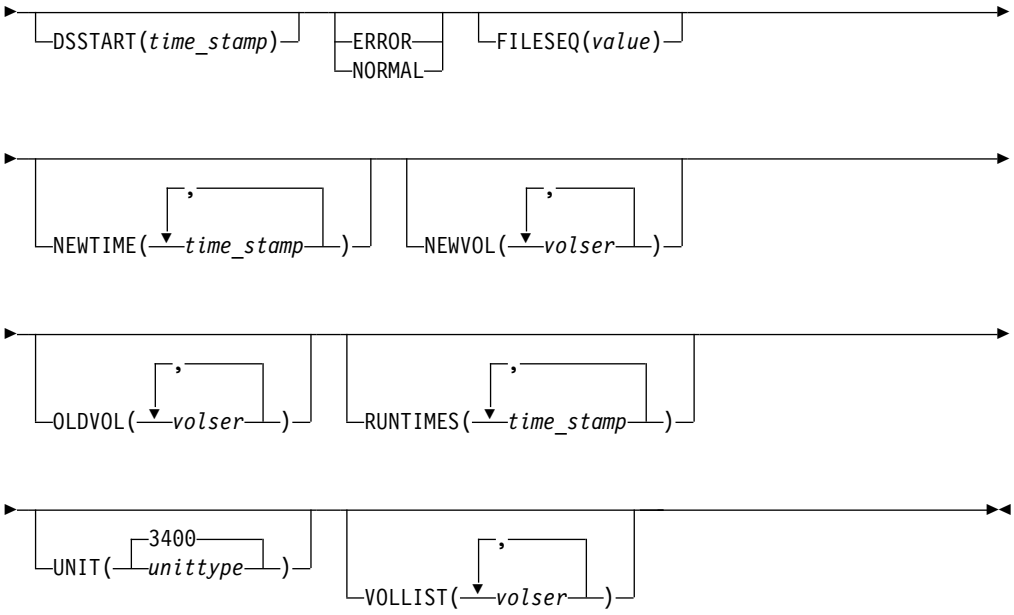
All the information you can change resides in a data set entry of the SECLOG record. Each CHANGE.SECLOG command you issue changes only one data set entry. If the log has multiple data sets, you must use the DSSTART keyword to identify the data set entry to be changed.

Subsections:

- "Syntax"
- "Keywords" on page 135
- "Examples of using the CHANGE.SECLOG RLDS command" on page 138

Syntax





Keywords

RLDS
Is the keyword you can use to specify that a SECLOG record is to be changed. Since RLDS is the default, if you do not specify a record type as the first keyword for CHANGE.SECLOG, RLDS is assumed.

STARTIME(*time_stamp*)
Required keyword you use to specify the starting time stamp of the SECLOG record that is to be changed. The time stamp must be in standard form.

DSN(*name*)
Optional keyword you use to change data set name. *name* can be up to 44 characters.

CHKPTCT(*value*)
Optional keyword you use to change the number of checkpoints completed on each volume of the data set. Specify a value for each volume designated in the OLDVOL or NEWVOL keywords. If OLDVOL is specified without NEWVOL, the number of values for CHKPTCT equals the number of volume serial numbers that appear with OLDVOL. If NEWVOL is specified, the number of values for CHKPTCT equals the number of volume serial numbers that appear in NEWVOL.

The values for CHKPTCT are:

0 No checkpoints on the volume

1 A single checkpoint on the volume

2 More than one checkpoint on the volume

CHKPTID(*chkptid*)
Optional keyword you use to change the oldest checkpoint ID for any active PST on each volume of the data set. Specify one checkpoint ID for each volume listed in OLDVOL or NEWVOL. If OLDVOL is specified without

NEWVOL, the number of checkpoint IDs equals the number of volumes listed in OLDVOL. If NEWVOL is specified, the number of checkpoint IDs equals the number of volumes listed in NEWVOL.

The checkpoint ID must be in standard form for a time stamp. You can specify a zero time value.

DSSTART(*time_stamp***)**
Is a keyword you use to specify the starting time of the data set entry to be changed. The DSSTART keyword is required if the SECLOG has multiple data set entries; it is optional if the SECLOG has only one data set entry. The time stamp must be in standard form.

ERROR | NORMAL
Mutually exclusive, optional keywords you use to change the data set entry to indicate whether it contains errors.

ERROR
Optional keyword you use to change the data set to indicate that it contains errors and should not be used as input to any DBRC-controlled run of a recovery utility.

NORMAL
Optional keyword you use to change a data set which was previously marked as containing errors to indicate that it is now available for use as input to any recovery utility.

DBRC selects the required log data sets from the PRILOG (or SECLOG) records. These can contain RLDS entries, SLDS entries, or both. If you issue a CHANGE.PRILOG RLDS ERROR command, DBRC automatically uses the corresponding SECLOG entry, if one exists. If a SECLOG entry does not exist, or if it is marked in error, the GENJCL commands that require log data for this time frame fail.

FILESEQ(*value***)**
Optional keyword you use to specify the file sequence number on the volume. Specify this keyword only if you specify a VOLLIST keyword. The value you substitute in the variable field must be a decimal number from 1 to 9999.

NEWTIME(*time_stamp***)**
Optional keyword you use to change the stop times of any but the last volume of the data set. If you specify NEWTIME, you must also specify OLDVOL and NEWVOL. The following keyword sets are mutually exclusive:
- NEWTIME, NEWVOL, OLDVOL
- RUNTIMES, VOLLIST

If you specify NEWTIME, you must specify one less time stamp than the number of volume serial numbers specified in NEWVOL. This is because the stop time of the last volume specified in NEWVOL cannot be changed with this command. Each time stamp is used as the volume stop time of the corresponding volume serial number specified by NEWVOL. If not specified, the stop time of the new volume is the same as the stop time of the last-specified old volume.

Each time stamp you specify must be greater than the previous time stamp. The first time stamp in NEWTIME must be greater than or equal to the stop time of the volume immediately preceding the changed volumes. Each time stamp must be in standard form.

NEWVOL(*volser***)**
Optional keyword you use to change the volume serial number of one or more

volumes of the data set. If you specify NEWVOL, you must also specify OLDVOL. The following keyword sets are mutually exclusive:
- NEWTIME, NEWVOL, OLDVOL
- RUNTIMES, VOLLIST

The volume serial numbers you specify in NEWVOL replace the corresponding volume serial numbers specified in the OLDVOL keyword. You do not need to specify the same number of volume serial numbers in NEWVOL and OLDVOL. You cannot specify a volume serial number in NEWVOL that is the same as one which already exists in the SECLOG record.

You can specify from 1 to 255 volume serial numbers.

Use the NEWTIME keyword to change the time stamps as well as the serial numbers of the volumes.

OLDVOL(*volser***)**
Optional keyword you use to change the volume serial number of one or more volumes of the data set. If you specify OLDVOL, you must also specify one of the following: NEWVOL, CHKPTCT, or CHKPTID.

The volume serial numbers you specify are those of the volumes being changed. Each volume serial number specified in OLDVOL must match a volume serial number in the SECLOG record.

You can specify from 1 to 255 volume serial numbers.

RUNTIMES(*time_stamp***)**
Optional keyword you use to change the stop times of any but the last volume of the data set. This keyword is provided for compatibility with previous releases of DBRC. Use the NEWTIME, OLDVOL, and NEWVOL keyword set to change the stop times of log volumes. If you do specify RUNTIMES, you must also specify VOLLIST. The following keyword sets are mutually exclusive:
- NEWTIME, NEWVOL, OLDVOL
- RUNTIMES, VOLLIST

You can specify up to 255 time stamps on the RUNTIMES keyword. Each time stamp must be in standard form.

Each time stamp in the variable field must correspond to a volume in the variable field of the VOLLIST keyword. The variable fields of the RUNTIMES and VOLLIST keywords must each contain the same number of entries. Each time stamp in the variable field of the RUNTIMES keyword must be greater than the previous time stamp.

The first time stamp in the variable field of the RUNTIMES keyword must be greater than the time stamp specified for the STARTIME keyword. The last time stamp in the variable field of the RUNTIMES keyword must be equal to the stop time of the corresponding secondary RLDS as specified in the record being changed. You cannot use this command to change the stop time of the secondary RLDS.

UNIT(3400 | *unittype*)
Optional keyword you use to change the unit type of the device on which the data set resides. The unit type can be up to 8 alphanumeric characters long.

VOLLIST(*volser***)**
Optional keyword you use to change the record of the volume serial numbers of the volumes that contain the data set. This keyword is provided for

compatibility with previous releases of DBRC. Use the NEWTIME, OLDVOL, and NEWVOL keyword set to change the volume serial numbers of volumes in the data set.

If you specify the VOLLIST keyword, you must also specify the RUNTIMES keyword. See the description of the RUNTIMES keyword for an explanation of how the two keywords interact. The following keyword sets are mutually exclusive:

- NEWTIME, NEWVOL, OLDVOL
- RUNTIMES, VOLLIST

Examples of using the CHANGE.SECLOG RLDS command

Here are some examples of using the CHANGE.SECLOG RLDS command.

Example of changing volume serial numbers

In this example, some volume serial numbers are changed. The example SECLOG record in the RECON data set has one data set with six volumes—VOL001, VOL002, VOL003, VOL004, VOL005, and VOL006—and a start time of 07233124329923456. The serial numbers of the third and fourth volumes are replaced with three others by the following command:

```
//CHGSECLG JOB
.
.
.
//SYSIN    DD   *
     CHANGE.SECLOG  RLDS STARTIME(07233124329923456)    -
                    OLDVOL(VOL003,VOL004)          -
                    NEWVOL(VOL007,VOL008,VOL009)
/*
```

Example of changing volume stop times

In this example, STARTIME identifies the SECLOG record and DSSTART identifies its first data set entry, of which the data set name and the stop times of three volumes are to be changed.

```
//CHGSECLG JOB
.
.
.
//SYSIN    DD   *
     CHANGE.SECLOG  RLDS STARTIME(07054121212023456) -
                    DSSTART(07054121212023456) -
                    DSN(IMS.SECLOG.SEC001.DSN) -
                    VOLLIST(VOL001,VOL002,VOL993) -
     RUNTIMES(070541212122,070541313133,070541515150)
/*
```

Related reference:

"DBRC time stamps" on page 52

"NOTIFY.PRILOG command (for RLDS)" on page 359

CHANGE.SECLOG SLDS command

Use the CHANGE.SECLOG SLDS command to change information in the RECON data set about a secondary SLDS for an online system.

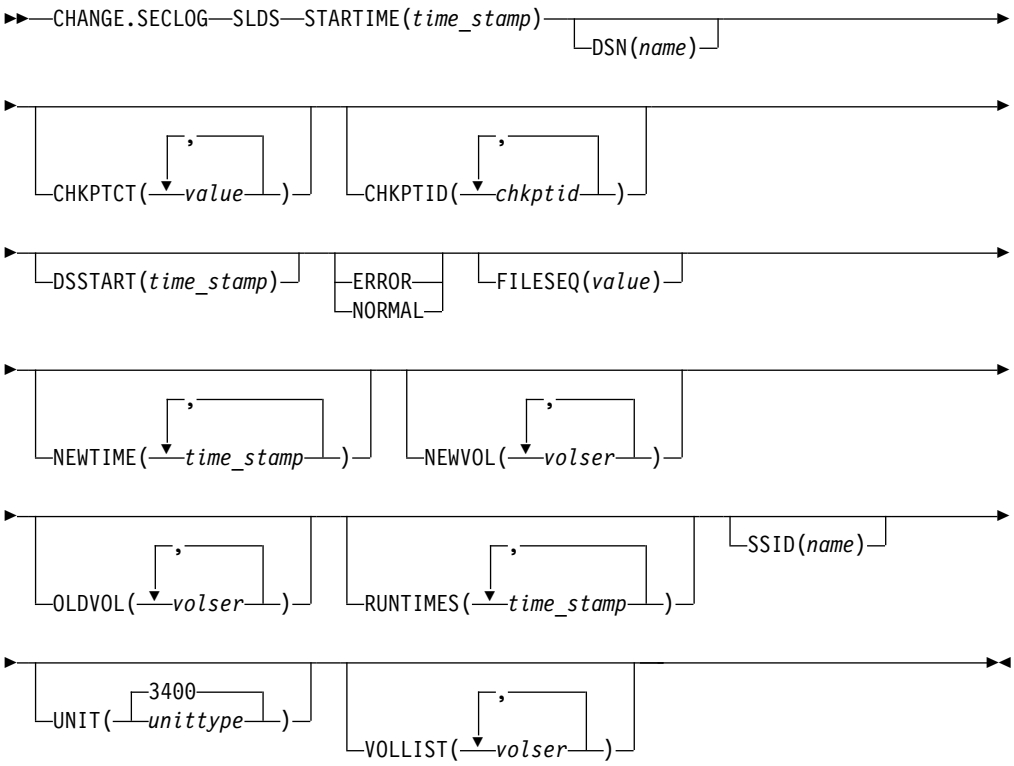
Use the CHANGE.SECLOG RLDS command to change information about an SLDS that a batch subsystem created, because DBRC considers such data to be an RLDS. Use the NOTIFY.SECLOG SLDS command to add a SECSLD record or to add data set entries to an existing SECSLD record.

All the information you can change resides in a data set entry of the SECSLD record. Each CHANGE.SECLOG command you issue changes only one data set entry. If the log has multiple data sets, you must use the DSSTART keyword to identify the data set entry to be changed.

Subsections:

- "Syntax"
- "Keywords"
- "Examples of using the CHANGE.SECLOG SLDS command" on page 142

Syntax



Attention: For this command you must specify the SLDS keyword. If SLDS is not specified, DBRC will default to processing an RLDS, which will not provide the desired results.

Keywords

SLDS
Required parameter you use to specify that a SECSLD record is to be changed.

STARTIME(*time_stamp*)
Required keyword you use to specify the starting time stamp of the SECSLD record that is to be changed. The time stamp must be in standard form.

DSN(*name*)
Optional keyword you use to change data set name. *name* can be up to 44 characters.

CHKPTCT(*value*)
Optional keyword you use to change the number of checkpoints completed on each volume of the data set. Specify a value for each volume designated in the OLDVOL or NEWVOL keywords. If OLDVOL is specified without NEWVOL, the number of values for CHKPTCT equals the number of volume serial numbers that appear with OLDVOL. If NEWVOL is specified, the number of values for CHKPTCT equals the number of volume serial numbers that appear in NEWVOL.

The values for CHKPTCT are:

0 No checkpoints on the volume

1 A single checkpoint on the volume

2 More than one checkpoint on the volume

CHKPTID(*chkptid*)
Optional keyword you use to change the oldest checkpoint ID for any active PST on each volume of the data set. Specify one checkpoint ID for each volume listed in OLDVOL or NEWVOL. If OLDVOL is specified without NEWVOL, the number of checkpoint IDs equals the number of volumes listed in OLDVOL. If NEWVOL is specified, the number of checkpoint IDs equals the number of volumes listed in NEWVOL.

The checkpoint ID must be in standard form for a time stamp. You can specify a zero time value.

DSSTART(*time_stamp*)
Is a keyword you use to specify the starting time of the data set entry to be changed.

The DSSTART keyword is required if the SECSLD has multiple data set entries. The keyword is optional if the SECSLD has only one data set entry.

The time stamp must be in standard form.

ERROR | NORMAL
Mutually exclusive, optional keywords you use to change the data set entry to indicate whether it contains errors.

ERROR
Is used to change the data set entry to indicate that it contains errors.

NORMAL
Is used to change a data set entry which was previously marked as containing errors to indicate that it is normal.

DBRC selects the required log data sets from the PRILOG (or SECLOG) records. These can contain RLDS entries, SLDS entries, or both. If you issue a CHANGE.PRILOG SLDS ERROR command, DBRC automatically uses the corresponding SECLOG entry, if one exists. If a SECLOG entry does not exist, or if it is marked in error, the GENJCL commands that require log data for this time frame fail.

FILESEQ(*value*)
Optional keyword you use to specify the file sequence number on the volume. Specify this keyword only if you specify a VOLLIST keyword. The value you substitute in the variable field must be a decimal number from 1 to 9999.

NEWTIME(*time_stamp*)
Optional keyword you use to change the stop times of any but the last volume of the data set. When you specify NEWTIME, you must also specify OLDVOL and NEWVOL. The following keyword sets are mutually exclusive:
- NEWTIME, NEWVOL, OLDVOL
- RUNTIMES, VOLLIST

When you specify NEWTIME, you must specify one less time stamp than the number of volume serial numbers specified in NEWVOL. This is because the stop time of the last volume specified in NEWVOL cannot be changed with this command. Each time stamp is used as the volume stop time of the corresponding volume serial number specified by NEWVOL. If not specified, the stop time of the new volume is the same as the stop time of the last—specified old volume.

Each time stamp you specify must be greater than the previous time stamp. The first time stamp in NEWTIME must be greater than or equal to the stop time of the volume prior to the changed volumes. Each time stamp must be in standard form.

NEWVOL(*volser*)
Optional keyword you use to change the volume serial number of one or more volumes of the data set. When you specify NEWVOL, you must also specify OLDVOL. The following keyword sets are mutually exclusive:
- NEWTIME, NEWVOL, OLDVOL
- RUNTIMES, VOLLIST

The volume serial numbers you specify in NEWVOL replace the corresponding volume serial numbers specified in the OLDVOL keyword. You do not need to specify the same number of volume serial numbers in NEWVOL and OLDVOL.

You cannot specify a volume serial number in NEWVOL that is the same as one that already exists in the SECSLD record.

You can specify from 1 to 255 volume serial numbers.

Use the NEWTIME keyword if you want to change the time stamps as well as the serial numbers of the volumes.

OLDVOL(*volser*)
Optional keyword you use to change the volume serial number of one or more volumes of the secondary SLDS.

The volume serial numbers you specify are those of the volumes to be changed. Each volume serial number specified must match a volume serial number in the SECSLD record.

When you specify OLDVOL, you must also specify one of the following: NEWVOL, CHKPTCT, or CHKPTID.
You can specify from 1 to 255 volume serial numbers.

RUNTIMES(*time_stamp*)
Optional keyword you use to change the stop times of any but the last volume of the data set. This keyword is provided for compatibility with previous releases of DBRC. Use the NEWTIME, OLDVOL, and NEWVOL keyword set to change the stop times of log volumes. If you do specify RUNTIMES, you must also specify VOLLIST. The following keyword sets are mutually exclusive:
- NEWTIME, NEWVOL, OLDVOL
- RUNTIMES, VOLLIST

You can specify up to 255 time stamps on the RUNTIMES keyword. Each time stamp must be in standard form.

Each time stamp in the variable field must correspond to a volume in the variable field of the VOLLIST keyword. The variable fields of the RUNTIMES and VOLLIST keywords must each contain the same number of entries. Each time stamp in the variable field of the RUNTIMES keyword must be greater than the previous time stamp.

The first time stamp in the variable field of the RUNTIMES keyword must be greater than the time stamp specified for the STARTIME keyword. The last time stamp in the variable field of the RUNTIMES keyword must be equal to the stop time of the corresponding secondary SLDS as specified in the record being changed. You cannot use this command to change the stop time of the secondary SLDS.

SSID(*name*)
Optional keyword you use to specify the name of the IMS subsystem that created the SLDS for which the RECON record is to be changed.

The SSID is an eight-character string consisting of any alphanumeric characters that describe a valid IMS subsystem identification name. If you do not specify SSID, DBRC uses the default subsystem identifier in the RECON header record. Use the INIT.SECLOG or CHANGE.SECLOG command to set the default subsystem identifier in the RECON header record. If you have not specified a default in the RECON header record, you must specify SSID.

UNIT(3400 | *unittype*)
Optional keyword you use to change the unit type of the device on which the data set resides. The unit type can be up to 8 alphanumeric characters long.

VOLLIST(*volser*)
Optional keyword you use to change the record of the volume serial numbers of the volumes that contain the data set. This keyword is provided for compatibility with previous releases of DBRC. You should use the new keyword set, NEWTIME OLDVOL NEWVOL, to change the volume serial numbers of volumes in the data set.

If you specify the VOLLIST keyword, you must also specify the RUNTIMES keyword. See the above description of the RUNTIMES keyword for an explanation of how the two keywords interact. The following keyword sets are mutually exclusive:
- NEWTIME, NEWVOL, OLDVOL
- RUNTIMES, VOLLIST

Examples of using the CHANGE.SECLOG SLDS command

Here are some examples of using the CHANGE.SECLOG SLDS command.

Example of changing volume serial numbers and stop time

In this example, some volume serial numbers and a volume stop time of an SECSLD are changed. The example SECSLD record in the RECON data set has six volumes (VOL001, VOL002, VOL003, VOL004, VOL005, and VOL006) and a start time of 07233124329923456. The fourth volume has been copied to new volumes VOL007 and VOL008 with the new volume stop time 07233124832523456 for VOL007. The SECSLD record is updated with the following command:

```
//CHGPRILG JOB
⋮
//SYSIN    DD   *
     CHANGE.SECLOG  SLDS STARTIME(07233124329923456)       -
                    OLDVOL(VOL004)                    -
                    NEWVOL(VOL007,VOL008)             -
                    NEWTIME(07233124832523456)
/*
```

Example of marking the secondary SLDS as normal

In this example, the first and only data set of a secondary SLDS is being marked as normal.

```
//CHGPRILG JOB
⋮
//SYSIN    DD   *
     CHANGE.SECLOG  SLDS STARTIME(07054121212023456) -
                    DSSTART(07054121212023456) NORMAL
/*
```

Related reference:

"DBRC time stamps" on page 52

"NOTIFY.PRILOG command (for SLDS)" on page 363

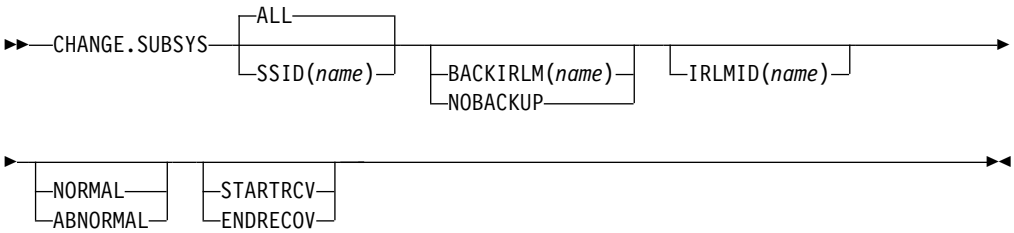
CHANGE.SUBSYS command

Use a CHANGE.SUBSYS command to modify information that is contained in a subsystem record in the RECON data set.

Subsections:

- "Syntax"
- "Keywords"
- "Example of identifying the IRLM" on page 145

Syntax



Keywords

ALL | SSID(*name*)
Optional keywords you use to specify the subsystem you are using. The SSID is an eight-character string consisting of any alphanumeric characters that comprise a valid z/OS or IMS subsystem identification name. ALL is the default, which will process this command for every subsystem that communicates with the specified Internal Resource Lock Manager (IRLM).

BACKIRLM(*name*) | NOBACKUP
Mutually exclusive, optional keywords you use to change the specification of the alternate subsystem.

BACKIRLM
Adds an alternate subsystem IRLM ID to the active subsystem record. When you specify BACKIRLM, you must also specify IRLMID. DBRC locates the specified subsystem record and adds or changes the IRLM ID of the alternate subsystem.

NOBACKUP
Deletes the IRLM ID of the alternate subsystem from the active subsystem record. This also resets the flags in the subsystem record to indicate that the alternate subsystem is signed on. This command might be required prior to restarting the alternate subsystem.

Restriction: You cannot use NORMAL or ABNORMAL, or STARTRCV or ENDRECOV with either of these keywords.

IRLMID(*name*)
Optional keyword you use to specify the name of the IRLM with which the subsystem is communicating. The IRLMID is a five-character string consisting of any alphanumeric characters.

Restriction: You cannot change the IRLM ID. Specify the IRLM ID in order to change processing mode of a subsystem.

NORMAL | ABNORMAL
Mutually exclusive, optional keywords you use to specify the status of the subsystem.

NORMAL
Specifies that normal processing is to continue for the subsystem.

ABNORMAL
Indicates that the subsystem has abnormally ended. When ABNORMAL is specified, DBRC does the following:

- Removes authorization for any databases that have not been updated but are authorized for the specified subsystem.
- Flags the identified subsystem entry as having been abnormally ended.
- Turns off the recovery-processing-started flag.
- If the subsystem is batch and no databases were updated, then the subsystem record is deleted.

Do not specify ABNORMAL unless the subsystem terminated abnormally and was unable to communicate the failure to DBRC.

Restriction: If you specify STARTRCV or ENDRECOV, you cannot specify ABNORMAL.

STARTRCV | ENDRECOV
Mutually exclusive, optional keywords you use to specify whether a signon recovery has completed successfully.

STARTRCV
Indicates a signon recovery start.

ENDRECOV
Removes authorization for all databases that the specified subsystem authorized.

If you want to delete all database authorizations from a subsystem, you must issue the CHANGE.SUBSYS STARTRCV command and then issue the CHANGE.SUBSYS ENDRECOV command. These two commands simulate the signon recovery start and signon recovery complete calls.

Recommendation: Do not use this sequence of commands unless an abnormal end occurred. Otherwise, you remove authorization for the databases that an active subsystem is currently using.

If, after using STARTRCV | ENDRECOV and DELETE.SUBSYS commands, subsystem information is still associated with the database, a CHANGE.DB command with the NOBACK keyword is required in order to clear the remaining subsystem ID from the database record.

Example of identifying the IRLM

In this example, IRLMID identifies the IRLM that is communicating with the subsystem identified in the SSID keyword. In addition, ABNORMAL indicates that this subsystem abnormally ended.

```
//CHGSBSYS   JOB
⋮
//SYSIN      DD    *
    CHANGE.SUBSYS  IRLMID(IRLM2)  SSID(ISM34)  ABNORMAL
/*
```

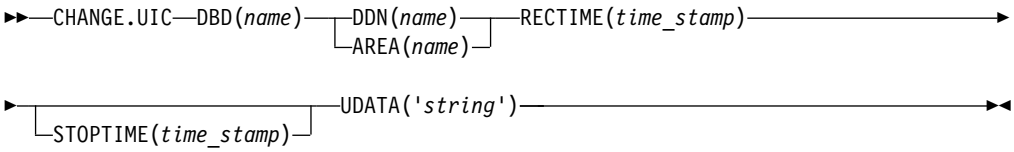
CHANGE.UIC command

Use a CHANGE.UIC command to modify information in the image copy record in the RECON data set that corresponds to a nonstandard image copy data set.

Subsections:

- "Syntax"
- "Keywords"
- "Example of changing the nonstandard ICDSN in the RECON data set" on page 146

Syntax



Keywords

DBD(*name*)

Required keyword you use to identify the database name of the DBDS for which a nonstandard image copy data set exists.

Restrictions for HALDB databases: For HALDB databases, *name* specifies the name of a HALDB partition.

DDN(*name*) | AREA(*name*)
Mutually exclusive, required keywords you use to identify the name of the DBDS or area for which the nonstandard image copy data set exists.

RECTIME(*time_stamp*)
Required keyword you use to identify the specific image copy record of the nonstandard image copy data set that is to be modified. Use the time stamp with an adjacent asterisk (*) in a listing of the IMAGE record. The time stamp must be in standard form.

STOPTIME(*time_stamp*)
Optional keyword that specifies the stop time of a concurrent image copy. The time stamp must be in standard form.

UDATA('*string*')
Required keyword you use to change the user data in the identified image copy record. *string* can be up to 80 characters. The *string* value must be enclosed in single quotation marks if blanks or any other special characters are included.

Example of changing the nonstandard ICDSN in the RECON data set

In this example, information in the RECON data set about the nonstandard image copy data set identified by the RECTIME keyword is to be changed. The UDATA keyword specifies the new information that is to be recorded for the specified image copy data set.

```
 //CHGUIC   JOB
⋮
 //SYSIN    DD   *
      CHANGE.UIC   DBD(DBDKSDS1)  AREA(AREA003)  -
        RECTIME(07065101010023456)  -
        UDATA('DUMP OF VOLUME VOL001 AT 07065101010023456')
 /*
```

Related reference:
"DBRC time stamps" on page 52

Chapter 11. CLEANUP.RECON command

Use the CLEANUP.RECON command to delete old or expired recovery-related information, change accumulation execution (CA) records, and log information from the RECON data set. Recovery-related information includes image copy, allocation, reorganization, and recovery records.

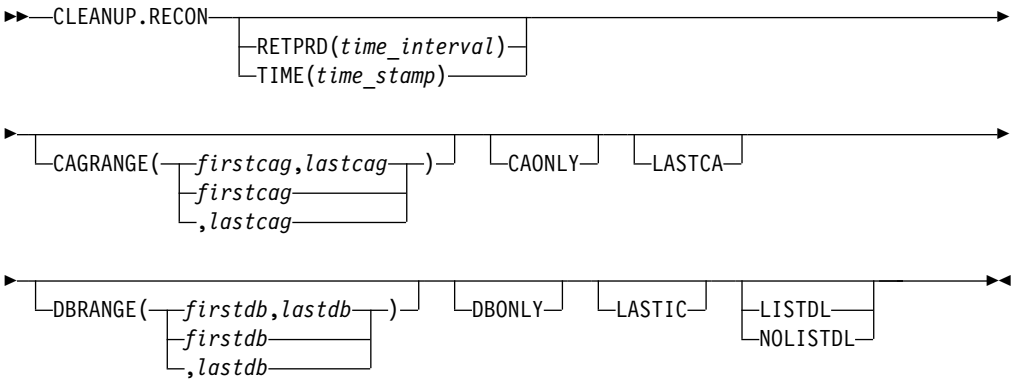
This command specifies the point in time up to which DBRC will delete database recovery-related information, change accumulation execution information, and log information from the RECON data set. To control the amount of time that is required to process this command, you can select the range of databases and change accumulation groups to be processed. This can be done with or without processing log information.

The CLEANUP.RECON command is supported only as a batch command.

Subsections:

- "Syntax"
- "Keywords"
- "Usage notes" on page 150
- "Examples of cleaning up information from the RECON data set" on page 150

Syntax



Keywords

The following keywords are valid for the CLEANUP.RECON command:

RETPRD(*time_interval*) | TIME(*time_stamp*)
Mutually exclusive optional parameters that specify either a retention period or an absolute time at which recovery-related information and log information in the RECON data set is to be deleted. If neither of these options is specified, the default log retention period is the LOGRET parameter value that is specified in the INIT.RECON command or the CHANGE.RECON command.

If the DBONLY option is not specified, PRILOG information will also be deleted. Unlike the DELETE.LOG INACTIVE command, which deletes an inactive PRILOG and its associated log records, log data will be deleted from

© Copyright IBM Corp. 1974, 2017

all PRILOGs, open and closed, up to each log's earliest database allocation record (ALLOC) time or the cleanup time, whichever is the oldest.

RETPRD(*time_interval*)
Specifies that a time period or interval will be used to determine what is to be deleted. This could be thought of as an aging value where data older than this "age" will be deleted and data younger than this "age" will be retained.

The *time_interval* is a partial, punctuated time stamp representing a time interval (days, hours, minutes, and seconds) rather than a date and time. The time stamp has the following format:

ddd|*hh*|*mm*|*ss*|*t*

where:

ddd
The number of days (000 to 365)

hh The number of hours (0 to 23)

mm The number of minutes (0 to 59)

ss The number of seconds (0 to 59)

t The tenths of a second (0 to 9)

The punctuation for the time stamp, shown in the above format as a vertical bar (|), can be any non-numeric character, such as a period (.) or a comma (,). The time stamp must be enclosed in single quotes (') if it contains any blanks or special characters. The number of days must include any leading zeros. Valid intervals range from a tenth of a second to 365 days.

Because the time interval is treated as a time stamp, message DSP1048I might be issued for incorrect values. Some examples of valid time intervals include:

```
CLEANUP.RECON RETPRD(365)
CLEANUP.RECON RETPRD('030 12.00')
CLEANUP.RECON RETPRD('000 00:00:08.0')
CLEANUP.RECON RETPRD('000 00,00,00,1')
```

Two different valid formats for equivalent time stamp specifications are shown:

```
CLEANUP.RECON RETPRD('010 12,30') = 10 days, 12 hours, 30 minutes
CLEANUP.RECON RETPRD('010 12:30') = 10 days, 12 hours, 30 minutes
```

Recommendation: Ensure that an appropriate log retention period is specified so that log information is kept in the RECON data set for a long enough period of time. Certain IMS functions depend on log information in the RECON data set, and these functions might fail if that information is deleted too soon. For example, one or more of the following functions might fail:

- Dynamic backout
- Batch message processing (BMP) extended restart (XRST)
- Unit of work table entry (UOWE) resynchronization

TIME(*time_stamp*)
Specifies that an absolute time stamp will be used to determine what is to be deleted. Only the data that is older than the time specified will be deleted.

CAGRANGE(*firstcag,lastcag* | *firstcag* |,*lastcag*)
Optional keyword that specifies that change accumulation data sets for a range of change accumulation (CA) groups are to have cleanup done. If this keyword is not specified, all CA groups are processed. Use this keyword to clean up change accumulation data sets for multiple change accumulation groups or to resume cleanup after a CLEANUP.RECON CAGRANGE command is stopped.

firstcag,lastcag
The upper and lower limits of the range of CA groups desired. These parameters do not need to match the name of a CA group in the RECON data set. They are used as the beginning and ending arguments in an alphabetic search.

firstcag
Name of the CA group from which RECON cleanup will begin processing change accumulation execution data sets. This CA group and all subsequent CA groups will be processed until the last group in the RECON data set is reached or until the CA group specified as the *lastcag* value is reached.

lastcag
Name of the CA group up to which RECON cleanup will continue to process CA groups. RECON cleanup will start with the first CA group defined in the RECON data set and will stop after this CA group is processed. If *lastcag* is specified, it must be preceded by a comma.

CAONLY
Optional keyword you use to specify that you want to delete only change accumulation execution records (no database recovery-related information or log information will be processed). However, if the DBONLY keyword is also included in the command, database recovery-related information will also be processed and only the log information will remain unprocessed.

LASTCA
Optional keyword you use to specify that the last CA execution record for a CA group can be deleted as part of the RECON cleanup process if it meets the criteria for deletion. If this keyword is not specified, the last CA execution record for a CA group will be retained.

DBRANGE(*firstdb,lastdb* | *firstdb* |,*lastdb*)
Optional keyword you use to specify a range of database names for which the cleanup is done. These are names of HALDB partitions and non-HALDB databases including DEDBs. HALDB master databases within the range are not processed. HALDB partitions within the range are processed. This option can be used to resume after a prior CLEANUP.RECON command was stopped.

firstdb,lastdb
Parameters you use to specify the upper or lower limits of the database range desired. The user must specify at least one of these parameters when using the DBRANGE keyword. Neither of these parameters needs to necessarily correspond to a name of a database registered in the RECON data set; they will simply be used as the beginning and ending arguments in an alphabetic search.

firstdb
Name of the database from which RECON cleanup will start processing databases. The RECON cleanup will process the database specified and then continue to process databases until it reaches the last database in the RECON data set or until it reaches the database specified as the *lastdb* in the DBRANGE keyword.

lastdb
Name of the database up to which RECON cleanup will continue to process databases. RECON cleanup will stop once this database has been processed. If *lastdb* is specified, it must be preceded by a comma.

DBONLY
Optional keyword you use to specify that you want to delete only database recovery-related information (no change accumulation or log information will be processed). However, if the CAONLY keyword is also specified, change accumulation information will also be processed and only the log information will remain unprocessed.

LASTIC
Optional keyword you use to specify that the last available IC for a database can be deleted as part of the RECON cleanup process if it meets the criteria for deletion. If this keyword is not specified, the last available IC for a database will be kept in order to avoid leaving a potentially useful database without a valid recovery point.

LISTDL | NOLISTDL
Mutually exclusive optional keywords you use to specify whether you want to list information about the data that was deleted from the RECON data set. If this option is not specified, the current LISTDL option specified in the RECON header will be used.

LISTDL
Specifies that information about the data that was deleted from the RECON data set is to be listed.

NOLISTDL
Specifies that information about the data that was deleted from the RECON data set is not to be listed.

Usage notes

Attention: To avoid deleting necessary recovery and log information from the RECON data set, run the CLEANUP.RECON command on a copy of the RECON data set first and verify the results, which indicate the length of time that is needed by the job. You can then use this information to determine when to run this command on your active RECON data set.

Recommendation: After using the CLEANUP.RECON command, reorganize your RECON data sets to reclaim space that was created as a result of issuing the command.

Examples of cleaning up information from the RECON data set

The following are some examples of the CLEANUP.RECON command.

Example 1 for CLEANUP.RECON command

Command input:

```
//CLEANUP  JOB
⋮
//SYSIN DD *
       CLEANUP.RECON RETPRD('010 12:30') LASTIC LISTDL
/*
```

Command output:

```
DSP1214I RECON INFORMATION WAS DELETED FOR DBDS DBNAME=DBXYZN DDN=DDXYZN
DSP1214I RECORD             TIME
DSP1214I    ALLOC             06.221 13:05:37.0
DSP1214I    IMAGE             06.221 13:05:22.0
DSP1214I                                 IC1 DSN=IMSVS. DBXYZN. DDXYZN.IC.IC130520
DSP1214I    IMAGE             06.221 13:45:47.0
DSP1214I                                 IC1 DSN=IMSVS. DBXYZN. DDXYZN.IC.IC134545
DSP1214I    RECOV       06.221 13:59:42.0
DSP1212W ALL EXISTING IMAGE COPIES FOR DBNAME= DBXYZN DDN= DDXYZN WERE DELETED
DSP1216I THE PRILOG FAMILY WITH TIME=06.221 13:04:17.0 AND SSID=IMS1 WAS DELETED
DSP1047I DELETED DSN= IMSVS.RLDSP.IMS1.D06221.T1304170.V00,FILESEQ= 0001,VOLSER= 000000
```

Explanation: In this example, the user issued a CLEANUP.RECON command specifying the LASTIC and LISTDL keywords along with a retention time of 10 days, 12 hours, and 30 minutes. There is only one database (DBXYZN) that has ALLOC, IC, and RECOV information. It has only two ICs, and both are older than the retention time specified.

As a result of the command, message DSP1214I is issued listing the records that were deleted, message DSP1212W is issued notifying the user that all existing image copies for the database have been deleted, and messages DSP1216I and DSP1047I are issued listing the PRILOG families and log data sets that were deleted.

Example 2 for CLEANUP.RECON command

Command input:

```
//CLEANUP  JOB
⋮
//SYSIN DD *
       CLEANUP.RECON RETPRD('010 12:30') LISTDL
/*
```

Command output:

```
DSP1214I RECON INFORMATION WAS DELETED FOR DBDS DBNAME=DBXYZN DDN=DDXYZN
DSP1214I RECORD             TIME
DSP1214I    ALLOC             06.221 13:05:37.0
DSP1214I    IMAGE             06.221 13:05:22.0
DSP1214I                                 IC1 DSN=IMSVS. DBXYZN. DDXYZN.IC.IC130520
DSP1214I    RECOV       06.221 13:59:42.0
DSP1213I THE LAST IMAGE COPY FOR DBNAME = DBXYZN DDN= DDXYZN WAS NOT DELETED
DSP1216I THE PRILOG FAMILY WITH TIME=06.221 13:04:17.0 AND SSID=IMS1 WAS DELETED
DSP1047I DELETED DSN= IMSVS.RLDSP.IMS1.D06221.T1304170.V00,FILESEQ= 0001,VOLSER= 000000
```

Explanation: In this example, the user issued a CLEANUP.RECON command specifying the LISTDL keyword along with a retention time of 10 days, 12 hours, and 30 minutes. There is only one database (DBXYZN) that has ALLOC, IC, and RECOV information. It has only two ICs, and both are older than the retention time specified.

As a result of the command, message DSP1214I is issued listing the records that were deleted, message DSP1213I is issued notifying the user that the last image copy for the database was not deleted even though it fell within the deletion period (because LASTIC was not specified), and messages DSP1216I and DSP1047I are issued listing the PRILOG families and log data sets that were deleted.

Example 3 for CLEANUP.RECON command

Command input:

```
//CLEANUP  JOB
:
//SYSIN DD *
       CLEANUP.RECON RETPRD('010 12:30') NOLISTDL
/*
```

Command output:

```
DSP1213I THE LAST IMAGE COPY FOR DBNAME = DBXYZN DDN= DDXYZN WAS NOT DELETED
DSP1216I THE PRILOG FAMILY WITH TIME=06.221 13:04:17.0 AND SSID=IMS1 WAS DELETED
```

Explanation: In this example, the user issued a CLEANUP.RECON command specifying the NOLISTDL keyword along with a retention time of 10 days, 12 hours, and 30 minutes. There is only one database (DBXYZN) that has ALLOC, IC, and RECOV information. It has only two ICs, and both are older than the retention time specified.

As a result of the NOLISTDL option, message DSP1214I is not issued, message DSP1213I is still issued notifying the user that the last image copy for the database was not deleted even though it fell within the deletion period (because LASTIC was not specified), and message DSP1216I is issued listing the PRILOG families that were deleted. Because the NOLISTDL option was specified, message DSP1047I is not issued.

Example 4 for CLEANUP.RECON command

Command input:

```
//CLEANUP  JOB
:
//SYSIN DD *
       CLEANUP.RECON RETPRD('010 12:30') DBRANGE(DBXYZN) LASTIC LISTDL
/*
```

Command output:

```
DSP1214I RECON INFORMATION WAS DELETED FOR DBDS DBNAME=DBXYZZ DDN=DDXYZZ
DSP1214I RECORD             TIME
DSP1214I    ALLOC             06.221 14:25:37.0
DSP1214I    IMAGE             06.221 14:25:22.0
DSP1214I                                IC1 DSN=IMSVS. DBXYZZ. DDXYZZ.IC.IC142520
DSP1214I    IMAGE             06.221 14:45:47.0
DSP1214I                                IC1 DSN=IMSVS. DBXYZZ. DDXYZZ.IC.IC144545
DSP1214I    RECOV       06.221 14:59:42.0
DSP1212W ALL EXISTING IMAGE COPIES FOR DBNAME= DBXYZZ DDN= DDXYZZ WERE DELETED
DSP1216I THE PRILOG FAMILY WITH TIME=06.221 14:24:17.0 AND SSID=IMS1 WAS DELETED
DSP1047I DELETED DSN= IMSVS.RLDSP.IMS1.D06221.T1424170.V00,FILESEQ= 0001,VOLSER= 000000
```

Explanation: In this example, the user decides to resume the cleanup processing after a prior CLEANUP.RECON RETPRD('010 12:30') LASTIC command was stopped. The user searched for message DSP1214I to find out that the last database processed is DBXYZN and issues a CLEANUP.RECON RETPRD('010 12:30') DBRANGE(DBXYZN) LASTIC LISTDL command. As a result, the command will begin processing databases starting with DBXYZN and continue until it processes the last database (which in this case is DBXYZZ). Database DBXYZZ has two ICs and both ICs are older than the retention time specified of 10 days, 12 hours and 30 minutes.

As a result of the command, message DSP1214I is issued listing the records that were deleted, message DSP1212W is issued to notify the user that all existing

image copies for the database have been deleted, and messages DSP1216I and DSP1047I are issued listing the PRILOG families and log data sets that were deleted.

Example 5 for CLEANUP.RECON command

Command input:

```
//CLEANUP  JOB
⋮
//SYSIN DD *
       CLEANUP.RECON RETPRD('010 12:30') DBONLY LASTIC LISTDL
/*
```

Command output:

```
DSP1214I RECON INFORMATION WAS DELETED FOR DBDS DBNAME=DBXYZN DDN=DDXYZN
DSP1214I RECORD             TIME
DSP1214I    ALLOC             06.221 13:25:37.0
DSP1214I    IMAGE             06.221 13:00:22.0
DSP1214I                                IC1 DSN=IMSVS. DBXYZN. DDXYZN.IC.IC130020
DSP1214I    IMAGE             06.221 13:45:47.0
DSP1214I                                IC1 DSN=IMSVS. DBXYZN. DDXYZN.IC.IC134545
DSP1214I    RECOV       06.221 13:35:47.0
DSP1212W ALL EXISTING IMAGE COPIES FOR DBNAME= DBXYZN DDN= DDXYZN WERE DELETED
```

Explanation: In this example, the user issued a CLEANUP.RECON command specifying the DBONLY, LASTIC, and LISTDL keywords along with a retention time of 10 days, 12 hours, and 30 minutes. There is only one database (DBXYZN) that has ALLOC, IC, and RECOV information, however, it only has two ICs and they both are older than the retention time specified.

As a result of the command, message DSP1214I is issued listing the records that were deleted, and message DSP1212W is issued notifying the user that all existing image copies for the database have been deleted. However, messages DSP1216I and DSP1047I are not issued in this example, because the DBONLY option was specified, which indicates that only the database portion of the RECON cleanup process is to be performed.

Example 6 for CLEANUP.RECON command

Command input:

```
//CLEANUP  JOB
⋮
//SYSIN DD *
       CLEANUP.RECON TIME(06221145536000000) LASTIC LISTDL
/*
```

Command output:

```
DSP1214I RECON INFORMATION WAS DELETED FOR DBDS DBNAME=DBXYZN DDN=DDXYZN
DSP1214I RECORD             TIME
DSP1214I    ALLOC             06.221 13:05:37.0
DSP1214I    IMAGE             06.221 13:05:22.0
DSP1214I                                IC1 DSN=IMSVS. DBXYZN. DDXYZN.IC.IC130520
DSP1214I    IMAGE             06.221 13:45:47.0
DSP1214I                                IC1 DSN=IMSVS. DBXYZN. DDXYZN.IC.IC134545
DSP1214I    RECOV       06.221 13:59:42.0
DSP1212W ALL EXISTING IMAGE COPIES FOR DBNAME= DBXYZN DDN= DDXYZN WERE DELETED
DSP1216I THE PRILOG FAMILY WITH TIME=06.221 13:04:17.0 AND SSID=IMS1 WAS DELETED
DSP1047I DELETED DSN= IMSVS.RLDSP.IMS1.D06221.T1304170.V00,FILESEQ= 0001,VOLSER= 000000
```

Explanation: In this example, the user issued a CLEANUP.RECON command specifying the LASTIC and LISTDL keywords along with a time stamp of '06221145536000000'. There is only one database (DBXYZN) that has ALLOC, IC, and RECOV information, however, it only has two ICs and they both are older than the time stamp specified.

As a result of the command, message DSP1214I is issued listing the records that were deleted, message DSP1212W is issued notifying the user that all existing image copies for the database have been deleted, and messages DSP1216I and DSP1047I are issued listing the PRILOG families and log data sets that were deleted.

Example 7 for CLEANUP.RECON command

Command input:

```
//CLEANUP  JOB
⋮
//SYSIN DD *
       CLEANUP.RECON TIME(06221145536000000) LISTDL
/*
```

Command output:

```
DSP1214I RECON INFORMATION WAS DELETED FOR DBDS DBNAME=DBXYZN DDN=DDXYZN
DSP1214I RECORD             TIME
DSP1214I    ALLOC             06.221 13:05:37.0
DSP1214I    IMAGE             06.221 13:05:22.0
DSP1214I                                IC1 DSN=IMSVS. DBXYZN. DDXYZN.IC.IC130520
DSP1214I    RECOV       06.221 13:59:42.0
DSP1213I THE LAST IMAGE COPY FOR DBNAME = DBXYZN DDN= DDXYZN WAS NOT DELETED
DSP1216I THE PRILOG FAMILY WITH TIME=06.221 13:04:17.0 AND SSID=IMS1 WAS DELETED
DSP1047I DELETED DSN= IMSVS.RLDSP.IMS1.D06221.T1304170.V00,FILESEQ= 0001,VOLSER= 000000
```

Explanation: In this example, the user issued a CLEANUP.RECON command specifying the LISTDL keyword along with a time stamp of '06221203236000000'. There is only one database (DBXYZN) that has ALLOC, IC, and RECOV information, however, it only has two ICs and they both are older than the time stamp specified.

As a result of the command, message DSP1214I is issued listing the records that were deleted, message DSP1213I is issued notifying the user that the last image copy for the database was not deleted even though it fell within the deletion period (because LASTIC was not specified), and messages DSP1216I and DSP1047I are issued listing the PRILOG families and log data sets that were deleted.

Example 8 for CLEANUP.RECON command

Command input:

```
//CLEANUP  JOB
⋮
//SYSIN DD *
       CLEANUP.RECON DBRANGE(,DBXYZZ) LASTIC LISTDL
/*
```

Command output:

```
DSP1214I RECON INFORMATION WAS DELETED FOR DBDS DBNAME=DBXYZN DDN=DDXYZN
DSP1214I RECORD             TIME
DSP1214I    ALLOC             06.221 13:05:37.0
DSP1214I    IMAGE             06.221 13:05:22.0
DSP1214I                                IC1 DSN=IMSVS. DBXYZN. DDXYZN.IC.IC130520
```

```
DSP1214I    IMAGE             06.221 13:45:47.0
DSP1214I                                IC1 DSN=IMSVS. DBXYZN. DDXYZN.IC.IC134545
DSP1214I    RECOV       06.221 13:59:42.0
DSP1212W ALL EXISTING IMAGE COPIES FOR DBNAME= DBXYZN DDN= DDXYZN WERE DELETED
DSP1216I THE PRILOG FAMILY WITH TIME=06.221 13:04:17.0 AND SSID=IMS1 WAS DELETED
DSP1047I DELETED DSN= IMSVS.RLDSP.IMS1.D06221.T1304170.V00,FILESEQ= 0001,VOLSER= 000000

DSP1214I RECON INFORMATION WAS DELETED FOR DBDS DBNAME=DBXYZZ DDN=DDXYZZ
DSP1214I RECORD             TIME
DSP1214I    ALLOC             06.221 14:25:37.0
DSP1214I    IMAGE             06.221 14:25:22.0
DSP1214I                                IC1 DSN=IMSVS. DBXYZZ. DDXYZZ.IC.IC142520
DSP1214I    IMAGE             06.221 14:45:47.0
DSP1214I                                IC1 DSN=IMSVS. DBXYZZ. DDXYZZ.IC.IC144545
DSP1214I    RECOV       06.221 14:59:42.0
DSP1212W ALL EXISTING IMAGE COPIES FOR DBNAME= DBXYZZ DDN= DDXYZZ WERE DELETED
DSP1216I THE PRILOG FAMILY WITH TIME=06.221 14:24:17.0 AND SSID=IMS1 WAS DELETED
DSP1047I DELETED DSN= IMSVS.RLDSP.IMS1.D06221.T1424170.V00,FILESEQ= 0001,VOLSER= 000000
```

Explanation: In this example, the user who has three databases (DBXYZN, DBXYZZ, and DBXZZZ) decides to issue a CLEANUP.RECON command to clean up the first two databases. In this case, the user issues a CLEANUP.RECON DBRANGE(,DBXYZZ) LASTIC LISTDL command, which will use the log retention period (LOGRET) as the default deletion time. As a result, the command begins by processing the first database (DBXYZN) and continues until it reaches the database specified in the 'lastdb' position of the DBRANGE keyword (DBXYZZ), using the log retention period (LOGRET) as the default deletion time. Both databases DBXYZN and DBXYZZ have two ICs that are older than the log retention period of 10 days, 12 hours, and 30 minutes.

As a result of the command, message DSP1214I is issued listing the records that were deleted, message DSP1212W is issued notifying the user that all existing image copies for the database have been deleted, and messages DSP1216I and DSP1047I are issued listing the PRILOG families and log data sets that were deleted.

Example 9 for CLEANUP.RECON command

Command input:

```
//CLEANUP  JOB
⋮
//SYSIN DD *
       CLEANUP.RECON RETPRD('010 12:30') LASTIC LASTCA LISTDL
/*
```

Command output:

```
DSP1214I RECON INFORMATION WAS DELETED FOR DBDS DBNAME=DBXYZN DDN=DDXYZN
DSP1214I RECORD             TIME
DSP1214I    ALLOC             09.021 13:05:37.012345
DSP1214I    IMAGE             09.021 13:05:22.012345
DSP1214I                          IC1 DSN=IMSVS.DBXYZN.DDXYZN.IC.IC130520
DSP1214I    IMAGE             09.021 13:45:47.012345
DSP1214I                          IC1 DSN=IMSVS.DBXYZN.DDXYZN.IC.IC134545
DSP1214I    RECOV       09.021 13:59:42.012345
DSP1212W ALL EXISTING IMAGE COPIES FOR DBNAME= DBXYZN DDN= DDXYZN WERE DELETED
DSP1225I CHANGE ACCUMULATION EXECUTION RECORDS WERE DELETED FOR CA GROUP GRPNAME=GRPDBX
DSP1225I RECORD            TIME
DSP1225I    CA               09.021 15:05:37.012345
DSP1225I                         DSN=IMSVS.GRPDBX.CA2.CA182601
DSP1216I THE PRILOG FAMILY WITH TIME=09.021 13:04:17.012345 AND SSID=IMS1 WAS DELETED
DSP1047I DELETED DSN= IMSVS.RLDSP.IMS1.D09021.T1304170.V00,FILESEQ= 0001,VOLSER=0000009
```

Explanation: In this example, the user issued a CLEANUP.RECON command specifying the LASTIC, LASTCA, and LISTDL keywords along with a retention time of 10 days, 12 hours, and 30 minutes. There is only one database (DBXYZN) that has ALLOC, IC, and RECOV information. It has only two ICs, and both are older than the retention time specified. There is one change accumulation group (GRPDBX) with reusable change accumulation execution records with five change accumulation execution records. Only one change accumulation execution record that is older than the retention period was deleted.

As a result of the command, message DSP1214I is issued listing the records that were deleted, message DSP1212W is issued notifying the user that all existing image copies for the database have been deleted, messages DSP1216I and DSP1047I are issued listing the PRILOG families and log data sets that were deleted, and message DSP1225I is issued listing the change accumulation execution record that was deleted.

Example 10 for CLEANUP.RECON command

Command input:

```
//CLEANUP  JOB
:
//SYSIN DD *
       CLEANUP.RECON RETPRD('010 12:30') CAONLY LISTDL
/*
```

Command output:

```
DSP1225I CHANGE ACCUMULATION EXECUTION RECORDS WERE DELETED FOR CA GROUP GRPNAME=CAGRP1
DSP1225I RECORD            TIME
DSP1225I    CA               09.021 13:05:37.012345
DSP1225I                         DSN=IMSVS.CAGRP1.CA2.CA182601
DSP1225I    CA               09.021 13:45:47.012345
DSP1225I                         DSN=IMSVS.CAGRP1.CA3.CA182602
```

Explanation: In this example, the user issued a CLEANUP.RECON command specifying the CAONLY keyword along with a retention time of 10 days, 12 hours, and 30 minutes. There are two change accumulation data sets that are older than 10 days.

As a result of the command, message DSP1225I is issued when change accumulation execution records are deleted for a change accumulation group. Because the LISTDL option is specified, each change accumulation execution record deleted is also listed.

Example 11 for CLEANUP.RECON command

Command input:

```
//CLEANUP  JOB
:
//SYSIN DD *
       CLEANUP.RECON RETPRD('010 12:30') CAGRANGE(CAGRP1) CAONLY NOLISTDL
/*
```

Command output:

```
DSP1225I CHANGE ACCUMULATION EXECUTION RECORDS WERE DELETED FOR CA GROUP GRPNAME=CAGRP2
DSP1225I CHANGE ACCUMULATION EXECUTION RECORDS WERE DELETED FOR CA GROUP GRPNAME=CAGRP3
DSP1225I CHANGE ACCUMULATION EXECUTION RECORDS WERE DELETED FOR CA GROUP GRPNAME=CAGRP4
```

```
DSP1225I CHANGE ACCUMULATION EXECUTION RECORDS WERE DELETED FOR CA GROUP GRPNAME=CAGRP5
DSP1225I CHANGE ACCUMULATION EXECUTION RECORDS WERE DELETED FOR CA GROUP GRPNAME=MYGRP1
DSP1225I CHANGE ACCUMULATION EXECUTION RECORDS WERE DELETED FOR CA GROUP GRPNAME=MYGRP2
```

Explanation: In this example, the user decides to clean up only change accumulation records for a range of change accumulation groups that begin with CAGRP1. The command will begin processing change accumulation group CAGRP1 and continue until it processes the last change accumulation group (which, in this case, is MYGRP2). Because change accumulation group CAGRP1 had already been processed by a previous CLEANUP.RECON command, no change accumulation executions records were deleted. Message DSP1225I is issued for each change accumulation group that had change accumulation records deleted. The change accumulation records deleted are not listed because of the NOLISTDL option.

Example 12 for CLEANUP.RECON command

Command input:

```
//CLEANUP  JOB
⋮
//SYSIN DD *
       CLEANUP.RECON RETPRD('010 12:30') CAONLY NOLISTDL
/*
```

Command output:

```
DSP1228I NO CHANGE ACCUMULATION INFORMATION WAS DELETED
```

Explanation: Because all the change accumulation groups had already been processed by a previous CLEANUP.RECON command, the only message that is issued for this command is DSP01228I.

Example 13 for CLEANUP.RECON command

Command input:

```
//CLEANUP  JOB
⋮
//SYSIN DD *
       CLEANUP.RECON RETPRD('010 12:30') CAGRANGE(,CAGRP4) CAONLY NOLISTDL
/*
```

Command output:

```
DSP1228I NO CHANGE ACCUMULATION INFORMATION WAS DELETED
```

Explanation: In this example, the user decides to resume the cleanup processing after a prior CLEANUP.RECON RETPRD('010 12:30') CAGRANGE(CAGRP1) command completed. As a result, this command will begin processing the first change accumulation group (CAGRP1) because the CAGRANGE keyword omitted the first change accumulation group and included CAGRP4 as the last change accumulation group to process. Because all the change accumulation groups had already been processed by the previous CLEANUP.RECON command, the only message issued for this command is DSP1228I.

Example 14 for CLEANUP.RECON command

Command input:

```
//CLEANUP  JOB
:
//SYSIN DD *
       CLEANUP.RECON DBRANGE(,DBXYZZ) LASTIC LISTDL
/*
```

Command output:

```
DSP1214I RECON INFORMATION WAS DELETED FOR DBDS DBNAME=DBXYZN DDN=DDXYZN
DSP1214I RECORD             TIME
DSP1214I    ALLOC             06.221 13:05:37.012345
DSP1214I    IMAGE             06.221 13:05:22.012345
DSP1214I                          IC1 DSN=IMSVS.DBXYZN.DDXYZN.IC.IC130520
DSP1214I    IMAGE             06.221 13:45:47.012345
DSP1214I                          IC1 DSN=IMSVS.DBXYZN.DDXYZN.IC.IC134545
DSP1214I    RECOV       06.221 13:59:42.0
DSP1212W ALL EXISTING IMAGE COPIES FOR DBNAME= DBXYZN DDN= DDXYZN WERE DELETED
DSP1216I THE PRILOG FAMILY WITH TIME=06.221 13:04:17.012345 AND SSID=IMS1 WAS DELETED
DSP1047I DELETED DSN= IMSVS.RLDSP.IMS1.D06221.T1304170.V00,FILESEQ= 0001,VOLSER= 000000

DSP1214I RECON INFORMATION WAS DELETED FOR DBDS DBNAME=DBXYZZ DDN=DDXYZZ
DSP1214I RECORD             TIME
DSP1214I    ALLOC             08.221 14:25:37.012345
DSP1214I    IMAGE             08.221 14:25:22.012345
DSP1214I                          IC1 DSN=IMSVS.DBXYZZ.DDXYZZ.IC.IC142520
DSP1214I    IMAGE             08.221 14:45:47.012345
DSP1214I                          IC1 DSN=IMSVS.DBXYZZ.DDXYZZ.IC.IC144545
DSP1214I    RECOV       08.221 14:59:42.012345
DSP1212W ALL EXISTING IMAGE COPIES FOR DBNAME= DBXYZZ DDN= DDXYZZ WERE DELETED
DSP1225I CHANGE ACCUMULATION EXECUTION RECORDS WERE DELETED FOR CA GROUP GRPNAME=CAGRP33
DSP1225I RECORD            TIME
DSP1225I    CA               09.021 13:05:37.012345
DSP1225I                         DSN=IMSVS.CAGRP33.CA2.CA182601
DSP1225I    CA               09.021 13:45:47.012345
DSP1225I                         DSN=IMSVS.CAGRP33.CA3.CA182602
DSP1216I THE PRILOG FAMILY WITH TIME=08.221 14:24:17.012345 AND SSID=IMS1 WAS DELETED
DSP1047I DELETED DSN= IMSVS.RLDSP.IMS1.D06221.T1424170.V00,FILESEQ= 0001,VOLSER= 000000
```

Explanation: In this example, the user who has three databases (DBXYZN, DBXYZZ, and DBXZZZ) decides to issue a CLEANUP.RECON command to clean up the first two databases. In this case, the user issues a CLEANUP.RECON DBRANGE(,DBXYZZ) LASTIC LISTDL command, which will use the log retention period (LOGRET) as the default deletion time. As a result, the command begins by processing the first database (DBXYZN) and continues until it reaches the database specified in the *lastdb* position of the DBRANGE keyword (DBXYZZ). Both databases DBXYZN and DBXYZZ have two ICs that are older than the log retention period of 10 days, 12 hours, and 30 minutes. Also, there are two change accumulation data sets for CAGRP33 that are older than the log retention period. They are deleted.

As a result of the command, message DSP1214I is issued listing the recovery related records that were deleted, message DSP1212W is issued notifying the user that all existing image copies for the database have been deleted, messages DSP1216I and DSP1047I are issued listing the PRILOG families and log data sets that were deleted, and message DSP1225I is issued notifying the user that the change accumulation execution records were deleted for CA group CAGRP33. Because of the LISTDL option, message DSP1225I also lists each change accumulation execution record that was deleted.

Example 15 for CLEANUP.RECON command

Command input:

```
//CLEANUP  JOB
:
//SYSIN DD *
       CLEANUP.RECON RETPRD('010 12:30') DBONLY CAONLY LASTIC LASTCA LISTDL
/*
```

Command output:

```
DSP1214I RECON INFORMATION WAS DELETED FOR DBDS DBNAME=DBXYZN DDN=DDXYZN
DSP1214I RECORD             TIME
DSP1214I    ALLOC             09.021 13:05:37.012345
DSP1214I    IMAGE             09.021 13:05:22.012345
DSP1214I                          IC1 DSN=IMSVS. DBXYZN.DDXYZN.IC.IC130520
DSP1214I    IMAGE             09.021 13:45:47.012345
DSP1214I                          IC1 DSN=IMSVS. DBXYZN.DDXYZN.IC.IC134545
DSP1214I    RECOV       09.021 13:59:42.012345
DSP1212W ALL EXISTING IMAGE COPIES FOR DBNAME= DBXYZN DDN= DDXYZN WERE DELETED
DSP1225I CHANGE ACCUMULATION EXECUTION RECORDS WERE DELETED FOR CA GROUP GRPNAME=GRPDBX
DSP1225I RECORD            TIME
DSP1225I    CA               09.021 15:05:37.012345
DSP1225I                         DSN=IMSVS.GRPDBX.CA2.CA182601
```

Explanation: In this example, the user issued a CLEANUP.RECON command specifying the DBONLY, CAONLY, LASTIC, LASTCA, and LISTDL keywords along with a retention time of 10 days, 12 hours, and 30 minutes. There is only one database (DBXYZN) that has ALLOC, IC, and RECOV information. It has only two ICs, and both are older than the retention time specified. There is one change accumulation group (GRPDBX) with five change accumulation execution records and only one change accumulation execution record that is older than the retention period. When either DBONLY or CAONLY is specified, log information is not processed.

As a result of the command, message DSP1214I is issued listing the records that were deleted, message DSP1212W is issued notifying the user that all existing image copies for the database have been deleted, and message DSP1225I is issued listing the change accumulation execution record deleted for CA group GRPDBX.

Chapter 12. DELETE commands

Use the DELETE commands to delete information from a RECON data set.

DELETE.ADS command

Use the DELETE.ADS command to delete an ADS from its associated area, which can consist of a maximum of seven ADSs, in the RECON record structure.

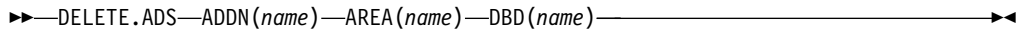
The ADS that is to be deleted must have been registered by the INIT.ADS command.

The DELETE.ADS command fails if you issue it while the area is authorized and the ADS is in AVAILABLE status. The command can be used if the ADS is in UNAVAILABLE status, provided that the ADS Create utility is not running.

Subsections:

- "Syntax"
- "Keywords"
- "Example of deleting an ADS record"

Syntax



Keywords

ADDN(*name***)**
Required keyword you use to identify the area name of the ADS to be deleted.

AREA(*name***)**
Required keyword you use to identify the name of the area that contains the ADS to be deleted.

DBD(*name***)**
Required keyword you use to identify the database name of the area that is to be deleted.

Example of deleting an ADS record

In this example, an ADS record is deleted from RECON for the DEDB area that is identified by the DBD, AREA, and ADDN keywords.

```
//DELADS   JOB  ('LEOPARD,IMS'),'LEOPARD',REGION=880K,
.
.
.
//SYSIN    DD   *
     DELETE.ADS  DBD(DBD00001)  AREA(AREA0001)  -
                 ADDN(AREA0002)
/*
```

© Copyright IBM Corp. 1974, 2017

DELETE.ALLOC command

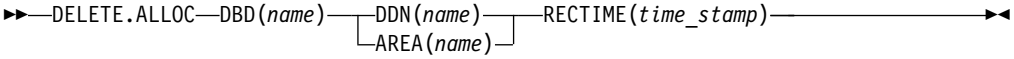
Use the DELETE.ALLOC command to delete from the RECON data set an allocation record that is related to a specified DBDS or DEDB area.

An allocation record can be deleted only when it contains a deallocation time or when its associated log has a stop time. Except for deleting allocation records that precede the oldest image copy data set for a DBDS or DEDB area, deleting an allocation record should be done with caution, and is not normally required. Deleting an allocation record that represents a period of time during which you changed the specified DBDS or area can cause a future recovery to be incorrect.

Subsections:

- "Syntax"
- "Keywords"
- "Example of deleting an allocation record"

Syntax



Keywords

DBD(*name***)**
Required keyword you use to identify the database name of the DBDS or area for which the allocation record is to be deleted.

Restriction for HALDB databases: For the DELETE.ALLOC command, *name* specifies the name of a HALDB partition.

DDN(*name***) | AREA(***name***)**
Mutually exclusive, required keywords you use to identify the data set ddname of the DBDS or DEDB area for which the allocation record is to be deleted.

RECTIME(*time_stamp***)**
Required keyword you use to identify the specific allocation record to be deleted for a specified DBDS or DEDB area. Use the time stamp with an adjacent asterisk (*) in a listing of the ALLOC record. The time stamp must be in standard form.

Example of deleting an allocation record

In this example, an allocation record is deleted from RECON for the DBDS that is identified by the DBD and DDN keywords. The RECTIME keyword identifies the specific allocation record that is to be deleted.

```
 //DELALLOC JOB
⋮
 //SYSIN    DD   *
      DELETE.ALLOC  DBD(DBDKSDS1) DDN(DDNKSDS1)  -
        RECTIME(07023110223423456)
 /*
```

Related reference:

"DBRC time stamps" on page 52

DELETE.BKOUT command

Use a DELETE.BKOUT command to delete backout records from the RECON data set.

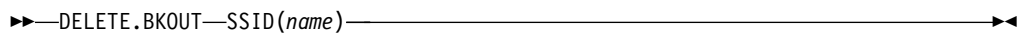
Use this command, for example, following the successful restore of a recent image copy. The backout information held in the RECON data set at the time of the copy is meaningless, but DBRC is not aware of this fact, and DBRC does not delete the backout records automatically.

Attention: Use the DELETE.BKOUT command with extreme caution. It deletes all backout information for a subsystem from the RECON; this is information that DBRC uses to help IMS maintain database integrity.

Subsections:

- " Syntax"
- "Keywords"
- "Example of using the DELETE.BKOUT command"

Syntax



Keywords

SSID(*name*)
Required keyword you use to identify the subsystem for which a backout record is to be deleted. The subsystem name is an eight-character, alphanumeric string that represents any valid subsystem. You can specify one subsystem each time you issue the command.

For each database entry in the record that is marked as backout required, the backout count in its associated database header record for this subsystem (SSID) is reduced by one. If this results in a zero backout count for this SSID, the SSID entry is removed from the database header record.

Example of using the DELETE.BKOUT command

This example uses the DELETE.BKOUT command to backout subsystem IMS3.

```
//DELBKOUT JOB
⋮
//SYSIN    DD   *
     DELETE.BKOUT  SSID(IMS3)
/*
```

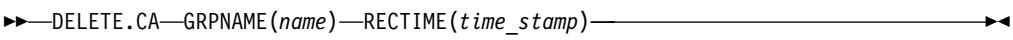
DELETE.CA command

Use a DELETE.CA command to delete from RECON a change accumulation run record for a specified CA group.

Subsections:

- "Syntax" on page 164
- "Keywords" on page 164
- "Example of deleting a run record" on page 164

Syntax



Keywords

GRPNAME(*name*)
Required keyword you use to specify the CA group. The CA run record that is to be deleted is a member of this CA group.

RECTIME(*time_stamp*)
Required keyword you use to specify the change accumulation run record that is to be deleted.

Use the RECTIME marked with an asterisk (*) from the listing of the CA record. The time stamp must be in standard form.

Example of deleting a run record

In this example, a run record is deleted from RECON for the CA group identified by the GRPNAME keyword. The RECTIME keyword identifies the record to be deleted.

```
 //DELCA    JOB
⋮
 //SYSIN    DD   *
      DELETE.CA   GRPNAME(CAGRP1)  RECTIME(07122090954023456)
 /*
```

Related reference:

"DBRC time stamps" on page 52

DELETE.CAGRP command

Use a DELETE.CAGRP command to delete a CA group record and all associated CA run records from the RECON data set.

Subsections:

- "Syntax"
- "Keywords"
- "Example of deleting CA group records" on page 165

Syntax

Keywords

GRPNAME(*name*)
Required keyword you use to specify the name of the CA group whose records that are to be deleted.

Example of deleting CA group records

In this example, CA group records are deleted from the RECON data set. The CA group for which the record is being deleted is identified by the GRPNAME keyword.

```
//DELCAGRP JOB
.
.
.
//SYSIN    DD   *
     DELETE.CAGRP   GRPNAME(CAGRP2)
/*
```

DELETE.DB command

Use the DELETE.DB command to delete from the RECON data set a database and all information related to the database that has been recorded in the RECON data set.

If the database is a HALDB, the database and all its partitions are deleted. If the named database is a HALDB partition, the command fails. Use the DELETE.PART command to delete a HALDB partition. If the database or any partition of a HALDB database is in use, the command fails and none of the RECON records are deleted.

Subsections:

- "Syntax"
- "Keywords"
- "Example of deleting records from the RECON data set"

Syntax

Keywords

DBD(*name*)

Required keyword you use to identify the name of the database to be deleted.

All database, DBDS, allocation, image copy, recovery, and reorganization records that have the same database name as *name* are deleted. In addition, all CA group and DBDS group records are scanned in order to delete any entries for which the corresponding DBDS records have been deleted. All log allocation records are also scanned in order to delete any entries in the allocation list for which the corresponding DBDS records have been deleted.

Restriction for HALDB databases: For the DELETE.DB command, *name* specifies the name of a HALDB master database.

Example of deleting records from the RECON data set

In this example, records are deleted from RECON for the database and its corresponding DBDSs identified by the DBD keyword.

```
//DELDB    JOB
⋮
//SYSIN    DD    *
     DELETE.DB  DBD(THISDB)
/*
```

Related concepts:

Partitions in PHDAM, PHIDAM, and PSINDEX databases (Database Administration)

DELETE.DBDS command

Use the DELETE.DBDS command to delete from the RECON data set all records that are related to a specified DBDS or DEDB area.

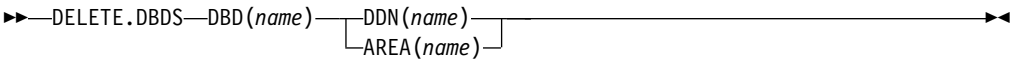
If the DBDS for which the records are to be deleted belongs to a CA group or to a DBDS group, its name is removed from the group record. The DELETE.DBDS command fails if the DL/I database or Fast Path DEDB area is in use.

Restriction for HALDB databases: The DELETE.DBDS command cannot be used to delete any DBDSs from a HALDB partition. Use the DELETE.PART command or the HALDB Partition Definition utility to delete a HALDB partition and all of its DBDSs.

Subsections:

- "Syntax"
- "Keywords"
- "Example of deleting records for the DBDS"

Syntax



Keywords

DBD(*name*)
Required keyword you use to specify the database name of the DBDS or DEDB area for which all records to be deleted.

DDN(*name*) | AREA(*name*)
Mutually exclusive, required keywords you use to specify the ddname of the DBDS or area for which all records are to be deleted from the RECON data set.

Example of deleting records for the DBDS

In this example, records are deleted from RECON for the DBDS identified by the DBD and DDN keywords.

```
//DELDBDS  JOB
⋮
//SYSIN    DD   *
     DELETE.DBDS   DBD(DBDESDSA)    DDN(DDNESDSA)
/*
```

DELETE.DBDSGRP command

Use a DELETE.DBDSGRP command to delete the record of a specified DBDS group from the RECON data set.

Subsections:

- "Syntax"
- "Keywords"
- "Example of deleting a DBDS group record"

Syntax

Keywords

GRPNAME(*name*)
Required keyword you use to specify the name of the DBDS group that is being deleted. The specified name must be that of a group that is identified in the RECON data set.

Example of deleting a DBDS group record

In this example, a DBDS group record is deleted from the RECON data set.

```
//DELDBDGP   JOB
⋮
//SYSIN      DD   *
     DELETE.DBDSGRP   GRPNAME(DBDSGRP1)
/*
```

DELETE.IC command

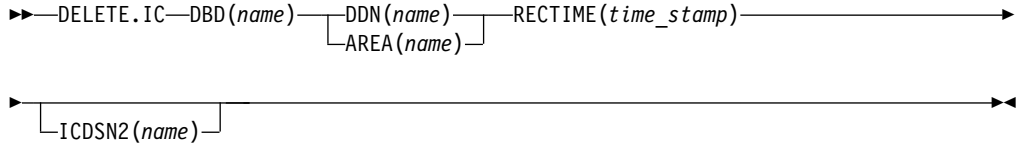
Use the DELETE.IC command to delete an image copy record or the information about a second image copy data set.

If you specify the ICDSN2 keyword, only the information about a second image copy data set is deleted; otherwise, both the entire image copy record and the information about the second image copy data set are deleted.

Subsections:

- "Syntax"
- "Keywords" on page 168
- "Example of deleting information from an image copy record" on page 168

Syntax



Keywords

DBD(*name*)
Required keyword you use to identify the image copy record to be deleted. *name* is the database name of the DBDS or DEDB area to which it is related.

Restriction for HALDB databases: For the DELETE.IC command, *name* specifies the name of a HALDB partition.

DDN(*name*) | AREA(*name*)
Mutually exclusive, required keywords you use to identify the image copy record to be deleted. *name* is the name of the DBDS or DEDB area to which it is related.

RECTIME(*time_stamp*)
Required keyword you use to identify the specific image copy record that is to be deleted. Use the RUN time marked with an asterisk (*) in a listing of the IMAGE record. The time stamp must be in standard form.

ICDSN2(*name*)
Optional keyword you use to specify the name of a duplicate image copy data set for which information is to be deleted from an image copy record. (The record of the first image copy data set remains in the RECON data set.)

Example of deleting information from an image copy record

In this example, information about a duplicate image copy data set is deleted from an image copy record in the RECON data set. The keywords DBD, AREA, ICDSN2, and RECTIME identify the information to be deleted. The asterisk (*) in the ICDSN2 keyword is to be expanded by DBRC according to the default-naming convention for image copy data sets.

```
//DELIC    JOB
.
.
.
//SYSIN    DD   *
     DELETE.IC   DBD(DBDKSDS1)     AREA(AREA006)  -
       RECTIME(07123122322123456)  ICDSN2(IMS.*.ICDSN5)
/*
```

Related reference:

"DBRC time stamps" on page 52

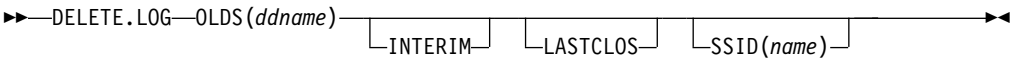
DELETE.LOG command (for OLDS)

Use the DELETE.LOG OLDS command to delete a data set entry from an OLDS record.

Subsections:

- "Syntax"
- "Keywords" on page 169
- "Example of deleting an interim OLDS record" on page 169

Syntax



Keywords

OLDS(*ddname*)
Required keyword you use to specify the ddname of the primary OLDS. DBRC deletes the RECON records of the primary and secondary OLDS for the specified subsystem with the specified *ddname*. You can delete the record of an OLDS only if the OLDS has been archived.

INTERIM
Optional keyword you use to specify that an interim OLDS record is to be deleted.

LASTCLOS
Optional keyword you use to specify that the OLDS specified in the OLDS keyword is the last OLDS in the PRIOLDS record and should be deleted. Use this keyword with caution. Normally, the last OLDS in the PRIOLDS record is the last OLDS that was closed, and you should not delete it. Close the first OLDS in a subsequent restart if the first OLDS is empty because of an error.

SSID(*name*)
Optional keyword you use to specify the name of the IMS subsystem that created the log data set for which the RECON record is to be deleted.

The SSID is an eight-character string of any alphanumeric characters that comprise a valid IMS subsystem identification name. If you do not specify SSID, DBRC uses the default subsystem identifier in the RECON header record. Use the INIT.RECON or CHANGE.RECON command to set the default subsystem identifier in the RECON header record. If you have not set a default in the RECON header record, you must specify SSID.

Example of deleting an interim OLDS record

In this example, an interim OLDS record is deleted from the RECON data set.

```
//DELLOG   JOB
.
.
.
//SYSIN    DD   *
     DELETE.LOG  SSID(IMSA) OLDS(DFSOLP03) -
       INTERIM
/*
```

DELETE.LOG command (for RLDS and SLDS)

Use the DELETE.LOG command to delete PRILOG and SLDS records, to delete single data set entries from those records, and to prevent PRILOG records from exceeding the maximum RECON record size.

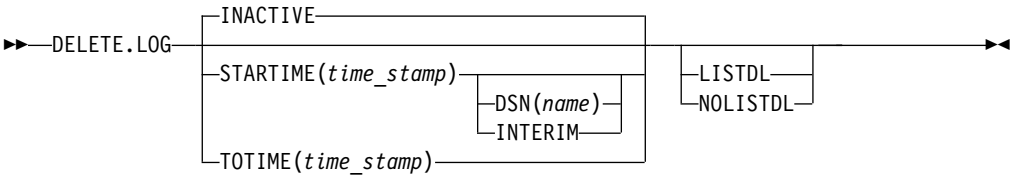
You can delete the following records:

- A PRILOG family of records
- One data set entry from a PRILOG family
- Some or all inactive PRILOG families

Subsections:

- "Syntax" on page 170
- "Keywords" on page 170
- "Example of deleting the record of an RLDS and SLDS" on page 172

Syntax



Keywords

INACTIVE | STARTIME(*time_stamp*) | TOTIME(*time_stamp*)

Mutually exclusive, optional keywords you use to specify the records that are to be deleted.

For a log to be considered inactive, all of the following conditions must be met:

- The log does not contain any DBDS change records more recent than the oldest image copy data set that is known to DBRC (empty LOGALL record).
- The log is older than the log retention period that was specified in the INIT.RECON or CHANGE.RECON command.
- The log has either been terminated (nonzero stop time) or has the ERROR flag in the PRILOG or SECLOG record set on.
- If the log has been terminated, the stop time in the PRILOG or SECLOG record must be older than the log retention period.

INACTIVE

Deletes an inactive PRILOG and its associated log records. INACTIVE is the default.

When the DELETE.LOG INACTIVE command is issued, DBRC also examines active PRILOG records to determine if they should be compressed (meaning that any inactive data set entries in the records will be deleted). A data set entry is defined as inactive if all of the following three conditions are met:

- It is older than the log retention period that was specified in the INIT.RECON or CHANGE.RECON command.
- It is older than the earliest log volume required for recovery of any database that is registered in the RECON data set.
- It is older than the earliest checkpoint that is required for system restart.

INACTIVE does not delete PRILOG and SECLOG records for logs that do not satisfy all three conditions. Most logs satisfy the conditions. Use DELETE.LOG STARTIME(*time_stamp*) command for those logs that do not satisfy all three conditions.

Recommendation: Ensure that an appropriate log retention period is specified so that log information is kept in the RECON data set for a long enough period of time. Certain IMS functions depend on log information in the RECON data set, and these functions might fail if that information is deleted too soon. For example, one or more of the following functions might fail:

- Dynamic backout
- Batch message processing (BMP) extended restart (XRST)
- Unit of work table entry (UOWE) resynchronization

STARTIME(*time_stamp*)
Specifies the START time of the log records to be deleted. Use the time stamp with an adjacent asterisk (*) in a list of the PRILOG or SECLOG record.

DSN(*name*)
Optional keyword (used with STARTIME) to specify the data set name of a particular log data set (located at the tracking site) whose entry in the RLDS or SLDS record is to be deleted. The specified data set name can exist in one or more of the primary and secondary RLDS and SLDS records. All entries having the same log sequence number range as the specified data set are deleted.

Only tracking log data at the tracking site (the data that is a partial copy of the log data at the active site) is eligible for deletion.

If the data set to be deleted is the last in the log record and if it is closed, the log stop time is set to zeros to indicate that a data set gap exists at the end of the record.

If DSN is not specified, the entire RLDS or SLDS record is deleted.

Note: STARTIME is required with DSN. You cannot specify TOTIME, INTERIM, or TRACKING with DSN.

INTERIM
Optional keyword you use to specify the interim RLDS record and interim SLDS record to be deleted.

Note: You cannot specify INTERIM if DSN is specified. If INTERIM is specified, you must also specify STARTIME.

TOTIME
Specifies that the records of all inactive RLDSs and SLDSs that have a stop time older than the time specified with the TOTIME keyword are to be deleted. You must specify a time that is older than the current time, minus the log retention period. The time stamp must be in standard form. If the PRILOG is older than TOTIME specified, but it is still active, the PRILOG will be examined and all inactive data set entries in the record will be deleted.

Restriction: The TOTIME and DSN parameters are mutually exclusive.

LISTDL | NOLISTDL
Mutually exclusive, optional keywords you use to specify whether the names of the data sets deleted from the RECON are to be listed in the job output. These keywords override the default. The default is the value specified by the INIT.RECON or CHANGE.RECON command.

LISTDL
Specifies that the names of deleted data sets are to be listed in the job output.

NOLISTDL
Specifies that the names of deleted data sets are not to be listed in the job output.

Example of deleting the record of an RLDS and SLDS

In this example, the interim RLDS and interim SLDS records that are identified by the STARTIME keyword are deleted from the RECON data set.

```
//DELLOG   JOB
.
.
.
//SYSIN    DD   *
     DELETE.LOG  STARTIME(07054121212023456)-
       INTERIM
/*
```

Related concepts:

Deleting unnecessary RECON records (System Administration)

Related reference:

"DBRC time stamps" on page 52

DELETE.PART command

Use the DELETE.PART command to delete from the RECON data set a HALDB partition and all information related to the partition that has been recorded in the RECON data set.

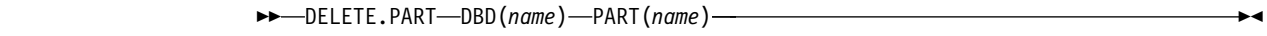
If the partition, or any other partition affected by the deletion, is in use, the command fails and none of the RECON records are deleted.

Restriction: You cannot delete individual DBDSs belonging to HALDB partitions with the DELETE.DBDS command. When a DELETE.PART command is issued, it removes all the DBDSs associated with this partition as well as all of their recovery related records (for example, ALLOC, IMAGE, RECOV, and REORG records).

Subsections:

- "Syntax"
- "Keywords"
- "Example of deleting a partition from RECON"

Syntax



Keywords

DBD(*name*)
Required keyword you use to identify the HALDB for which the partition is to be deleted.

PART(*name*)
Required keyword you use to identify the HALDB partition name to be deleted.

Example of deleting a partition from RECON

In this example, records are deleted from the RECON data set for the partition and its corresponding DB and DBDSs identified by the DBD and NAME keyword.

```
//DELPART JOB
//SYSIN DD *
DELETE.PART  DBD(DB3)  PART(PART3)
/*
```

Related concepts:

Partitions in PHDAM, PHIDAM, and PSINDEX databases (Database Administration)

DELETE.RECOV command

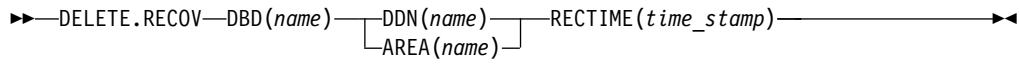
Use the DELETE.RECOV command to delete a specified recovery run record from the RECON data set.

Specifying DELETE.RECOV for the recovery run record of a time-stamp recovery implies that the DBDS or DEDB area that is related to the record has been restored. It has been restored to the state it was in just before the time-stamp recovery that created the recovery run record that you are deleting. Such a deletion also implies that no allocations of the DBDS or DEDB area took place that generated change records on IMS log data sets after the time-stamp recovery occurred.

Subsections:

- "Syntax"
- "Keywords"
- "Example of deleting a recovery record of the DBDS"

Syntax



Keywords

DBD(*name*)
Required keyword you use to identify the recovery record to be deleted; *name* is the database name of the related DBDS or DEDB area.

Restriction for HALDB databases: For the DELETE.RECOV command, *name* specifies the name of a HALDB partition.

DDN(*name*) | AREA(*name*)
Mutually exclusive, required keywords you use to identify the recovery record to be deleted; *name* is the name of the related DBDS or DEDB area.

RECTIME(*time_stamp*)
Required keyword you use to specify the time stamp of the recovery run record to be deleted. Use the time stamp with an adjacent asterisk (*) in a list of the RECOV record. The time stamp must be in standard form.

Example of deleting a recovery record of the DBDS

This example shows the deletion from the RECON data set of the record of a recovery of the DBDS identified by the DBD and DDN keywords. The record to be deleted is identified by the RECTIME keyword.

```
//DELRECOV JOB
⋮
//SYSIN    DD   *
     DELETE.RECOV  DBD(DBDESDSB)     DDN(DDNESDSB)  -
       RECTIME(07089191919023456)
/*
```

Related reference:

"DBRC time stamps" on page 52

DELETE.REORG command

Use a DELETE.REORG command to delete a database reorganization record for a specified DBDS from the RECON data set.

When you specify the DELETE.REORG command, you are implying that the DBDS and the IMS DBD library have been restored to the state they were in before the reorganization that created the database reorganization record. By using the DELETE.REORG command, you are also implying that no allocations of the reorganized database that generated records in IMS log data sets took place.

Subsections:

- "Syntax"
- "Keywords"
- "Example of deleting a reorganization record of a DBDS"

Syntax

Keywords

DBD(*name*)
Required keyword you use to identify the reorganization record to be deleted; *name* is the database name of the related DBDS.

Restriction for HALDB databases: For the DELETE.REORG command, *name* specifies the name of a HALDB partition.

DDN(*name*)
Required keyword you use to identify the reorganization record to be deleted; *name* is the data set ddname of the related DBDS.

RECTIME(*time_stamp*)
Required keyword you use to identify the specific database reorganization record to be deleted. Use the time stamp with an adjacent asterisk (*) in a list of the REORG record. The time stamp must be in standard form.

Example of deleting a reorganization record of a DBDS

In this example, a record of the reorganization of a DBDS is deleted from the RECON data set.

```
//DELREORG JOB
⋮
//SYSIN    DD   *
     DELETE.REORG  DBD(DBDESDSB)     DDN(DDNESDSB)  -
       RECTIME(07023110223423456)
/*
```

Related reference:

"DBRC time stamps" on page 52

DELETE.SUBSYS command

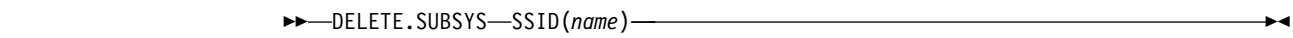
Use a DELETE.SUBSYS command to delete the subsystem entry in the RECON data set after it is verified that the specified subsystem is not authorized to use any database.

To close the subsystem log, issue the NOTIFY.PRILOG command, and then issue the DELETE.SUBSYS command.

Subsections:

- "Syntax"
- "Keywords"
- "Example of deleting a specified SUBSYS record"

Syntax



Keywords

SSID(*name*)
Required keyword you use to identify the subsystem for which the entry is deleted from RECON if no database is authorized by the subsystem.

When you issue this command online, the IMS control region under which the command was issued cannot be the subsystem being deleted.

Example of deleting a specified SUBSYS record

In this example, the specified SUBSYS record is deleted if no database is authorized by the subsystem.

```
//DELSBSYS   JOB
⋮
//SYSIN      DD    *
    DELETE.SUBSYS  SSID(IMS34)
/*
```

Related reference:

"CHANGE.SUBSYS command" on page 143

Related information:

DSP0045I (Messages and Codes)

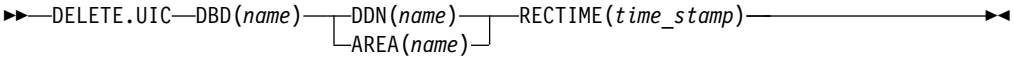
DELETE.UIC command

Use a DELETE.UIC command to delete the record of a nonstandard image copy data set from the RECON data set.

Subsections:

- "Syntax"
- "Keywords"
- "Example of deleting a nonstandard image copy data set record"

Syntax



Keywords

DBD(*name*)
Required keyword you use to identify the nonstandard image copy record to be deleted; *name* is the database name of the related DBDS or area.

Restriction for HALDB databases: For the DELETE.UIC command, *name* specifies the name of a HALDB partition.

DDN(*name*) | AREA(*name*)
Mutually exclusive, required keywords you use to identify the nonstandard image copy record to be deleted; *name* is the name of the related DBDS or DEDB area.

RECTIME(*time_stamp*)
Required keyword you use to specify the time stamp of the nonstandard image copy record to be deleted. Use the time stamp with an adjacent asterisk (*) in a list of the IMAGE record. The time stamp must be in standard form.

Example of deleting a nonstandard image copy data set record

This example shows the deletion of a record of a nonstandard image copy data set from the RECON data set.

```
//DELUIC   JOB
.
.
.
//SYSIN    DD   *
     DELETE.UIC    DBD(DBDESDSB)   AREA(AREAESD2) -
       RECTIME(07087121212023456)
/*
```

Related reference:

"DBRC time stamps" on page 52

Chapter 13. GENJCL commands

Use the GENJCL commands to generate JCL and utility control statements that run the various IMS recovery utilities.

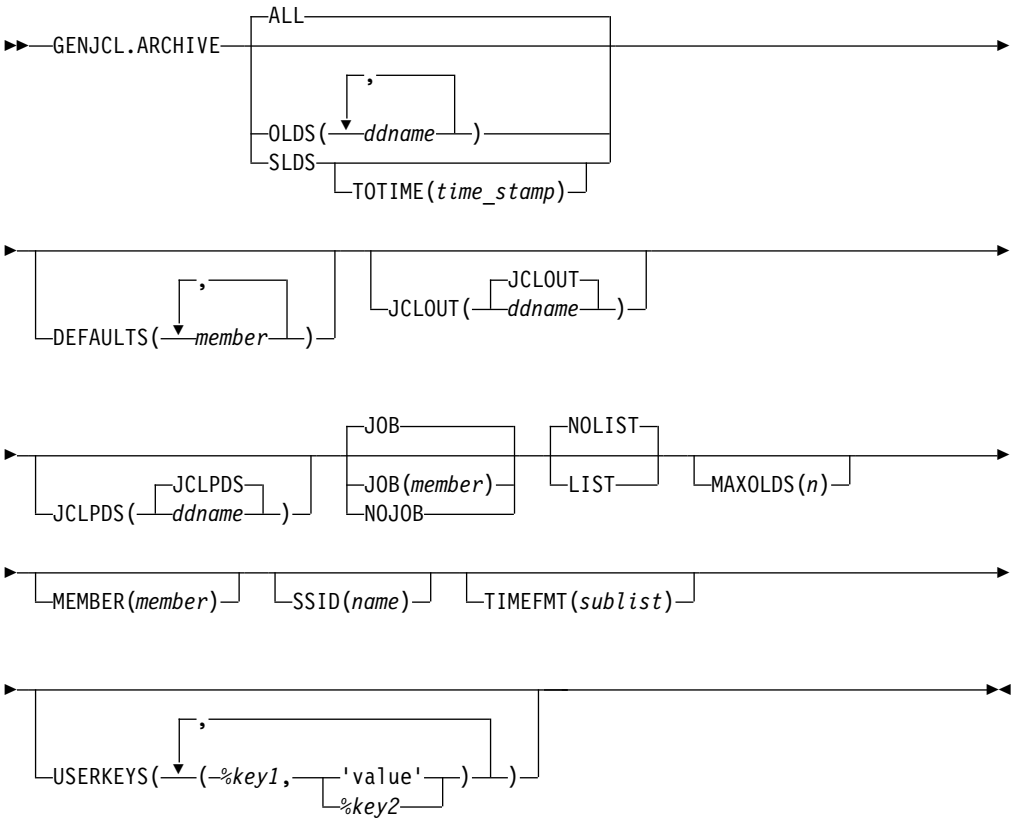
GENJCL.ARCHIVE command

Use the GENJCL.ARCHIVE command to generate the JCL and utility control statements that run the Log Archive utility.

Subsections:

- "Syntax"
- "Keywords"
- "Examples" on page 180

Syntax



Keywords

ALL | OLDS(*ddname*) | SLDS
Mutually exclusive, optional keywords that you use to specify which OLDS is being archived or to request the archive of tracking SLDSs.

© Copyright IBM Corp. 1974, 2017

ALL
Generates JCL to archive all OLDSs that have not been archived. A multiple-step job can be produced if either of the following conditions exist:

- The specified subsystem has non-contiguous OLDSs.
- A force-EOV condition occurred after you entered /DBRECOVERY.

OLDS
Specifies the ddname of the primary OLDS you are archiving.

SLDS
Generates JCL to archive all tracking SLDSs which are associated with the specified subsystem that have not been archived. A multiple-step job can be produced if the PRISLD or SECSLD (or both) have non-contiguous data set entries that need to be archived, or if they have more unarchived DSNs than the specified MAXOLDS value.

TOTIME(*time_stamp***)**
Specifies that only tracking log data sets with start times older than or equal to *time_stamp* are to be archived. This keyword is optional and is valid only when SLDS is also specified. Otherwise it is ignored. The time stamp must be in standard format.

DEFAULTS(*member***)**
Optional keyword you use to specify up to 10 skeletal JCL default members to be used when generating JCL. Default members are searched to resolve keywords in the order in which the members are specified on this keyword.

If a keyword is assigned a value in both the DEFAULTS and USERKEYS keywords, the value specified in USERKEYS is used.

JCLOUT(JCLOUT | *ddname*)
Optional keyword you use to specify the output data set for the generated JCL. The data set is specified by *ddname*. A JCL DD statement with this ddname must be included in the job step containing the GENJCL command. The specified data set can be a member of a partitioned data set (PDS) as long as it is not the same data set used for the default JCLOUT.

JCLPDS(JCLPDS | *ddname*)
Optional keyword you use to specify the skeletal JCL data set that is to be used for input when generating JCL. The data set that is specified by *ddname*. A JCL DD statement with this ddname must be included in the job step containing the GENJCL command.

JOB | JOB(*member*) | NOJOB
Mutually exclusive, optional keywords that you use to specify whether to produce the job statement in the generated JCL.

JOB
Specifies that the job statement is to be produced. When JOB is specified without a member name, the IBM-supplied execution member JOBJCL produces the job statement. When JOB(*member*) is specified, the specified execution member produces the job statement.

NOJOB
Specifies that the job statement is not produced in the generated JCL.

NOLIST | LIST
Mutually exclusive, optional keywords that you use to specify whether to print the generated JCL using the SYSPRINT data set.

NOLIST
Suppresses printing of the generated JCL.

LIST
Prints the generated JCL.

MAXOLDS(*n*)
Optional keyword you use to specify the maximum number of OLDSs or SLDSs to be archived in a single job. *n* can be any decimal number from 1 to 100.

If MAXOLDS is specified and more OLDSs need archiving than are specified in *n*, multiple jobs are generated. Each generated job archives no more than *n* OLDSs.

This keyword functions somewhat differently for SLDSs than for OLDSs. If MAXOLDS is specified and more SLDSs need archiving than are specified in *n*, multiple job steps are generated. Each generated job step archives no more than *n* SLDSs.

MAXOLDS applies only to the primary data sets. If dual logging is in effect, each job can have DD statements for the secondary and primary data sets (that is, DD statements for 2 x *n* data sets).

If you do not specify MAXOLDS, a single job is generated for all OLDSs or SLDSs.

MEMBER(*member*)
Optional keyword you use to specify the name of the skeletal JCL execution member to be used. If this keyword is not specified, the IBM-supplied execution member for the GENJCL.ARCHIVE command is used.

SSID(*name*)
Optional keyword you use to specify the name of the IMS subsystem that created the OLDSs or SLDSs that are to be archived.

The SSID is an eight-character string of any alphanumeric characters that comprise a valid IMS subsystem identification name. If you do not specify SSID, DBRC uses the default subsystem identifier in the RECON header record. Use the INIT.RECON or CHANGE.RECON command to set the default system identifier in the RECON header record. If you have not set a default in the RECON header record, you must specify SSID.

TIMEFMT(*sublist*)
Optional keyword you use to define the form in which time stamps appear in user-defined output. This specification overrides both the `GENJCL` default values and any values set on `%SET` statements in the input skeletal JCL.

The default for the `GENJCL` output time format is compressed, with a two-digit year, and the offset in numeric form: 960021315001 +0700. If you want the output time stamps to appear without offsets, for example, you can override the default with TIMEFMT(,N).

The override is good only for the duration of a single GENJCL command.

USERKEYS(*%key1*,*'value'* | *%key2*)

Optional keyword you use to set the value of keywords you have defined. Up to 64 keywords can be specified.

%key1
User-defined keyword being assigned a value. The maximum length of the

keyword is 8 characters, including the percent sign. The first character after the percent sign must be alphabetic (A-Z). The remaining characters must be alphanumeric (A-Z, 0-9).

'value'
Value assigned to the user-defined keyword. *value* can be any character string enclosed in single quotation marks. The maximum length of *value* is 132 characters (excluding the quotation marks). If *value* itself contains a quotation mark, specify two single quotation marks. (For example, if *value* is `TRK'S`, specify `'TRK''S'`). *value* can be a null string (''). If *value* is a time stamp, you can set it to 0.

%key2
Any simple keyword that was previously assigned a value, including DBRC-defined and user-defined keywords.

Any keyword can be assigned a value with the USERKEYS keyword. However, if you assign a value to DBRC-defined keywords, the value is ignored.

If a keyword is assigned a value in both the USERKEYS and DEFAULTS keyword, the value specified in USERKEYS is used.

Examples

Here are some examples of using the GENJCL.ARCHIVE command.

Example with primary OLDS defined by the OLDS keyword

In this example, a GENJCL.ARCHIVE command generates the JCL and control statements required to run the Log Archive utility for the primary OLDSs that are defined by the OLDS keyword. When this command is issued, the PRIOLDS record in the RECON data set is updated to indicate that an archive has been scheduled for the OLDS. Default skeletal member ARCHJCL is taken from the data set that is identified in the JCLPDS DD statement. The generated JCL goes to the data set identified in the JCLOUT DD statement. Skeletal member JOBJCL produces a job statement.

```
//GENJAR   JOB
//JCLOUT   DD   . . .
//JCLPDS   DD   . . .

.
.
.
//SYSIN    DD   *
     GENJCL.ARCHIVE SSID(IMSA) -
                    OLDS(DFSOLP01,DFSOLP02)
/*
```

As part of the archive process, the PRIOLDS record in the RECON data set is updated to indicate that the archive has completed. The RECON data set is updated with the PRISLD and SECSLD records that identify the created SLDSs. In addition, the RECON data set is updated with the PRILOG and SECLOG records that identify the created RLDSs.

Example of the SSID IMSB OLDS keyword defining the primary OLDS

In this example, a GENJCL.ARCHIVE command generates JCL to archive the primary OLDS that is defined in the OLDS keyword for SSID IMSB. JCL execution member ARCHJCLA is taken from the JCLPDS data set that is identified in the

PDSJCL DD statement. The generated JCL goes to SYSOUT=A, which is identified in the OUTJCL DD statement. Skeletal member JOBJCL produces a job statement.

```
//GENJAR1 JOB
//OUTJCL  DD   SYSOUT=A
//PDSJCL  DD   DSN=dsname
//SYSIN   DD   *
GENJCL.ARCHIVE SSID(IMSB) OLDS(DFSOLP01) MEMBER(ARCHJCLA) -
JCLPDS(PDSJCL) JCLOUT(OUTJCL)
```

Example for unarchived default subsystem OLDSs

In this example, the GENJCL.ARCHIVE command generates JCL and control statements to archive all unarchived OLDSs for the default subsystem ID.

JCL execution member ARCHJCLB is taken from the JCLPDS data set that is identified by the JCLPDS DD statement. Member DEFARC01 from the JCLPDS data set (identified in the JCLPDS DD statement) contains values to resolve user-defined keywords in ARCHJCLB. `%SSPACE` is a user-defined keyword in member ARCHJCLB which is assigned a value of 'CYL,1'. `%RSPACE` is a user-defined keyword in member ARCHJCLB, which is assigned a value of 'TRK,4'.

The values specified in the USERKEYS keyword for a keyword overrides the values found in the DEFAULTS member. JOB1 is a member in the JCLPDS that produces a job statement.

```
//GENJAR2 JOB
//JCLPDS  DD   . . .
//JCLOUT  DD   . . .
//SYSIN   DD   *
GENJCL.ARCHIVE MEMBER(ARCHJCLB) DEFAULTS(DEFARC01) -
USERKEYS((%SSPACE,'CYL,1'),(%RSPACE,'TRK,4')) JOB(JOB1)
```

Related reference:

"Generating JCL and user-defined output" on page 223

"DBRC time stamps" on page 52

"Skeletal JCL syntax" on page 225

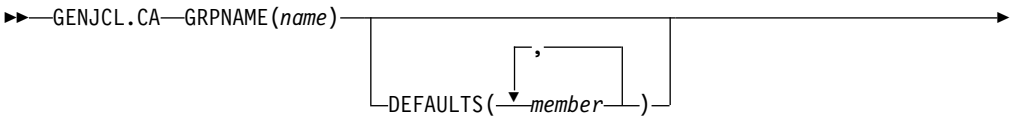
GENJCL.CA command

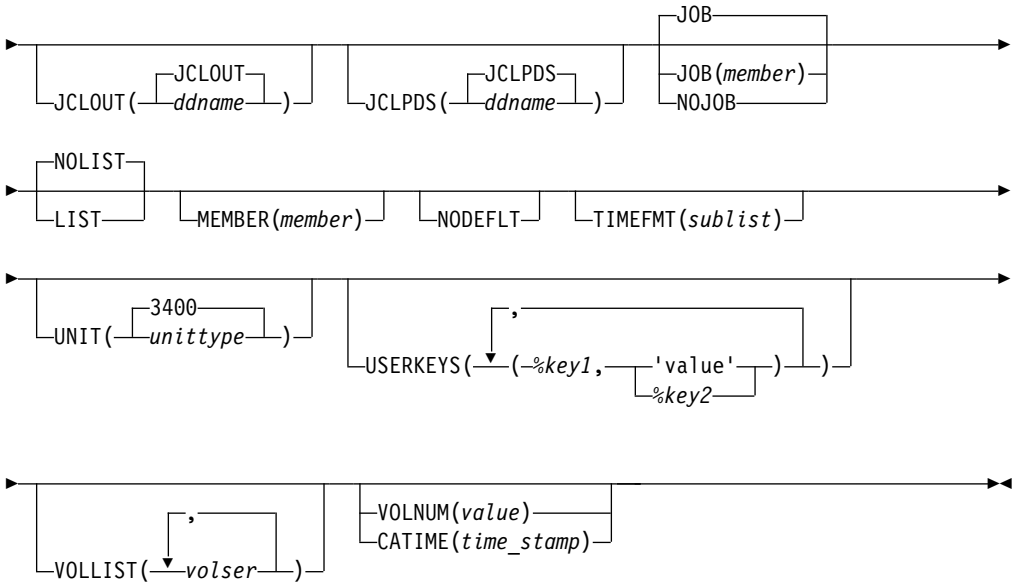
Use the GENJCL.CA command to generate the JCL and utility control statements to run the Change Accumulation utility for a specified CA group.

Subsections:

- "Syntax"
- "Keywords " on page 182
- "Examples" on page 184

Syntax





Keywords

GRPNAME(*name***)**
Required keyword you use to specify the name of the CA group for which you are running the Change Accumulation utility.

DEFAULTS(*member***)**
Optional keyword you use to specify the names of up to 10 skeletal JCL default members that are used when generating JCL. Default members are searched to resolve keywords in the order in which the members are specified on this keyword.

If a keyword is assigned a value in both the DEFAULTS and USERKEYS keywords, the value specified in USERKEYS is used.

JCLOUT(JCLOUT | *ddname*)
Optional keyword you use to specify the output data set for the generated JCL. The data set is specified by *ddname*. A JCL DD statement with this ddname must be included in the job step containing the GENJCL command. The specified data set can be a member of a partitioned data set as, but only if it is not the same data set that is used for the default, JCLOUT.

JCLPDS(JCLPDS | *ddname*)
Optional keyword you use to specify the skeletal JCL data set that is to be used for input when generating JCL. The data set is specified by *ddname*. A JCL DD statement with this ddname must be included in the job step containing the GENJCL command.

JOB | JOB(*member*) | NOJOB
Mutually exclusive, optional keywords that you use to specify whether to produce the first job statement in the generated JCL.

JOB
Specifies that the job statement is to be produced. When JOB is specified without a member name, the IBM-supplied execution member JOBJCL produces the job statement.

JOB(*member*)
Specified execution member produces the job statement.

NOJOB
Specifies that the job statement is not produced in the generated JCL.

NOLIST | LIST
Mutually exclusive, optional keywords that you use to specify whether you want the generated JCL to be written to the SYSPRINT data set.

NOLIST
Suppresses the printing of the generated JCL.

LIST
Prints the generated JCL.

MEMBER(*member*)
Optional keyword you use to specify the name of the skeletal JCL execution member that is to be used. If this keyword is not specified, the default specified for the CA group is used.

NODEFLT
Optional keyword you use to specify that the implicit skeletal JCL default member, if any, for the CA group is not to be used.

TIMEFMT(*sublist*)
Optional keyword you use to define the form in which time stamps appear in user-defined output. This specification overrides both the GENJCL default values and any values set on %SET statements in the input skeletal JCL.

The default for the GENJCL output time format is compressed, with a two-digit year, and the offset in numeric form: 960021315001 +0700. If you want the output time stamps to appear without offsets, for example, you can override the default with TIMEFMT(,N).

The override is good only for the duration of a single GENJCL command.

UNIT(3400 | *unittype*)
Optional keyword you use to specify the unit type of the output change accumulation data set. This keyword is valid only when both of the following conditions are true:

- The VOLLIST keyword is specified.
- The CA group for which the JCL is being generated is defined with the NOREUSE keyword.

USERKEYS(*%key1*,*'value'* | *%key1*, *%key2*)

Optional keyword you use to set the value of keywords you have defined. Up to 64 keywords can be specified.

%key1
User-defined keyword being assigned a value. The maximum length of the keyword is 8 characters, including the percent sign. The first character after the percent sign must be alphabetic (A-Z). The remaining characters must be alphanumeric (A-Z, 0-9).

'value'
Value assigned to the user-defined keyword when it is encountered. *value* can be any character string enclosed in single quotation marks. The maximum length of *value* is 132 characters (excluding the quotation marks). If *value* contains a quotation mark, use two single quotation marks. *value* can be a null string (''). If *value* is a time stamp, it can be 0.

%key2
Any simple keyword that was previously assigned a value, including DBRC-defined and user-defined keywords.

Any keyword can be assigned a value with the USERKEYS keyword. However, if you assign a value to DBRC-defined keywords, the value is ignored.

If a keyword is assigned a value in both the USERKEYS and DEFAULTS keyword, the value specified in USERKEYS is used.

VOLLIST(*volser***)**
Optional keyword you use to specify the volumes that are to contain the change accumulation data set. This keyword is valid only if the CA group for which the JCL is being generated was defined with the NOREUSE keyword.

VOLNUM(*value***) | CATIME(***time***)**
Mutually exclusive, optional keywords that you use to specify the log volumes that are used by the Change Accumulation utility.

VOLNUM(*value***)**
Specifies the number of log volumes that are to be used in each change accumulation job step. DBRC generates a multiple-step job that invokes the Change Accumulation utility in each step (unless you specify VOLLIST), and limits the number of log volumes in each step to the specified number. If another volume is needed to complete subset processing, VOLNUM may be overridden by DBRC. DBRC may also override VOLNUM for the following reasons:

- CATDS is specified and a data set entry spans multiple volumes.
- Log volumes have identical start times.
- Log volumes have identical start times and stop times.

For *value*, specify the number of log volumes. You can specify a decimal number from 1 to 255.

Each job step except the first one uses the change accumulation data set (that was generated in the previous step) as the beginning point of the accumulation in that step.

CATIME(*time_stamp***)**
Specifies the time after which no log volumes for the specified CA group are to be included. The time stamp does not need to be the stop time of any log volume. DBRC uses the time stamp as the ending delimiter for the log volume subset. Therefore, all log volumes that have start times less than or equal to the specified time stamp are included in the subset of volumes. The time stamp must be in standard form.

Examples

Here are some examples of using the GENJCL.CA command.

Example for the GRPNAME CA group

In this example, a GENJCL.CA command generates the JCL and control statements required to run the Change Accumulation utility for the CA group identified in the GRPNAME keyword. CAGRP1 is defined as REUSE. If the INIT.CAGRP command for CAGRP1 is specified without a CAJCL(*member*) keyword, default skeletal member CAJCL from the data set identified in the JCLPDS DD statement is used. If INIT.CAGRP is specified with the CAJCL(*member*) keyword, that member is

used. The generated JCL goes to the data set that is identified in the JCLOUT DD statement. Skeletal member JOBJCL produces a job statement.

```
//GENJCA   JOB
//JCLPDS   DD   . . .
//JCLOUT   DD   . . .

⋮
//SYSIN    DD   *
     GENJCL.CA     GRPNAME(CAGRP1)
/*
```

Example of CAJCLA generated skeletal JCL

In this example, the GENJCL.CA command is generated with skeletal JCL execution member CAJCLA, which was taken from the JCLPDS data set identified by the PDSJCL DD statement. Output from the generated JCL goes to SYSOUT=A, identified in the OUTJCL DD statement. CAGRP2 is defined with the NOREUSE keyword. Skeletal member JOBJCL produces a job statement.

```
//GENJCA1 JOB
//OUTJCL  DD   SYSOUT=A
//PDSJCL  DD   DSN=dsname
//SYSIN   DD   *
GENJCL.CA GRPNAME(CAGRP2) VOLLIST(VOL001) MEMBER(CAJCLA) -
JCLPDS(PDSJCL) JCLOUT(OUTJCL)
```

Example of CAJCLB generated skeletal JCL

In this example, the GENJCL.CA command generates JCL and control statements to run the Change Accumulation utility for CAGRP3, which is defined as REUSE. JCL execution member CAJCLB is taken from the JCLPDS data set identified by the JCLPDS DD statement.

DEFAULTS(DEFCA01) is a member in the JCLPDS data set which contains values to resolve user defined keywords in member CAJCLB. The default member for the CAGRP, if initialized in the INIT.CAGRP DEFLTJCL(MEMBER) command, is also used to resolve keywords. `%DISP` is a user-defined keyword in member CAJCLB which is assigned a value of 'SHR'. `%OUTCLS` is a user-defined keyword in member CAJCLB which is assigned a value of 'B'.

The values in the explicitly defined DEFAULTS member overrides values in the predefined DEFLTJCL member. The values specified in the USERKEYS keyword for a keyword overrides the values found in the DEFAULTS member. JCL is generated with no job statement. All volumes that have stop times less than or equal to the specified time stamp are included in the subset of volumes that is used as input to the Change Accumulation utility. Generated JCL is listed.

```
//GENJCA3 JOB
//JCLPDS  DD
//JCLOUT  DD
//SYSIN   DD   *
GENJCL.CA GRPNAME(CAGRP3) MEMBER(CAJCLB) DEFAULTS(DEFCA01) -
USERKEYS((%DISP,'SHR'),(%OUTCLS,'B')) NOJOB LIST -
CATIME(071020202111)
```

Related reference:

"Generating JCL and user-defined output" on page 223

"DBRC time stamps" on page 52

"Skeletal JCL syntax" on page 225

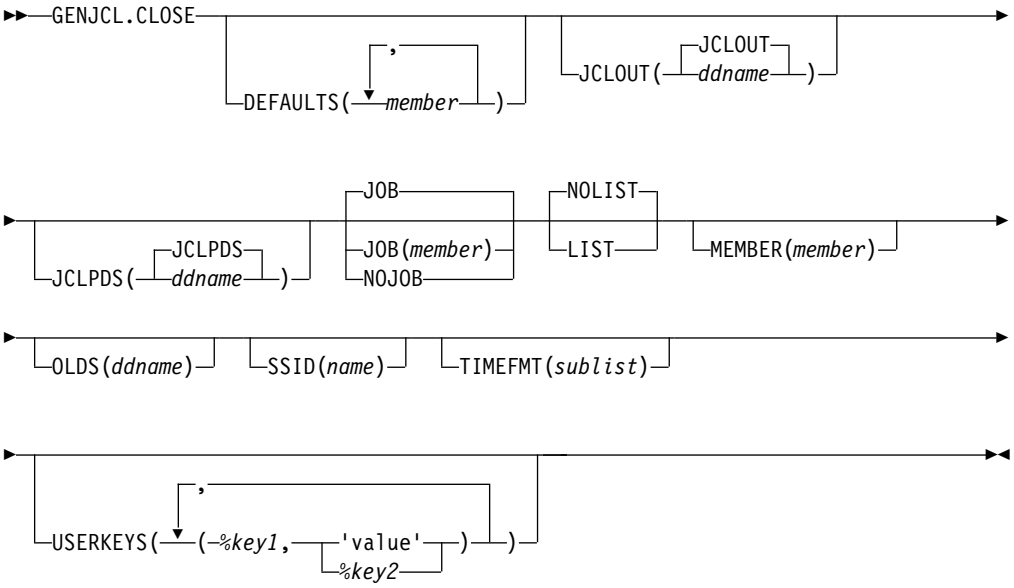
GENJCL.CLOSE command

Use the GENJCL.CLOSE command to generate the JCL and utility control statements that run the Log Recovery utility to close an OLDS using the WADS.

Subsections:

- "Syntax"
- "Keywords"
- "Examples" on page 188

Syntax



Keywords

DEFAULTS(*member*)
Optional keyword you use to specify up to 10 names of skeletal JCL default members used when generating JCL. Default members are searched to resolve keywords in the order in which the members are specified on this keyword.

If a keyword is assigned a value in both the DEFAULTS and USERKEYS keywords, the value specified in USERKEYS is used.

JCLOUT(JCLOUT | *ddname*)
Optional keyword you use to specify the output data set for the generated JCL. The data set is specified by *ddname*. A JCL DD statement with this ddname must be included in the job step containing the GENJCL command. The specified data set can be a member of a partitioned data set, but only if it is not the same data set that is used for the default, JCLOUT.

JCLPDS(JCLPDS | *ddname*)
Optional keyword you use to specify the skeletal JCL data set to be used for input when generating JCL. The data set is specified by *ddname*. A JCL DD statement with this ddname must be included in the job step containing the GENJCL command.

JOB | JOB(*member*) | NOJOB
Mutually exclusive, optional keywords that you use to specify whether to produce the job statement in the generated JCL.

JOB
Specifies that the job statement is produced. When JOB is specified without a member name, the IBM-supplied execution member JOBJCL produces the job statement.

JOB(*member*)
Specified execution member produces the job statement.

NOJOB
Specifies that no job statement is to be produced in the generated JCL.

NOLIST | LIST
Mutually exclusive, optional keywords that you use to specify whether you want the generated JCL to be written to the SYSPRINT data set.

NOLIST
Suppresses the printing of the generated JCL.

LIST
Prints the generated JCL.

MEMBER(*member*)
Optional keyword you use to specify the name of the skeletal JCL execution member to be used. If this keyword is not specified, the IBM-supplied execution member for the GENJCL.CLOSE command is used.

OLDS(*ddname*)
Optional keyword you use to specify which OLDS is to be closed. You specify the name of the DD statement that was used when the online IMS subsystem created the log data. The ddname of the primary OLDS must be specified. If you do not specify the OLDS, DBRC generates JCL to close the OLDS that was most recently opened.

SSID(*name*)
Optional keyword you use to specify the name of the IMS subsystem that created the OLDSs being closed.

The SSID is an eight-character alphanumeric string that comprises a valid IMS subsystem identification name. If you do not specify SSID, DBRC uses the default subsystem identifier in the RECON header record. Use the INIT.RECON or CHANGE.RECON command to set the default system identifier in the RECON header record. If you have not set a default in the RECON header record, you must specify SSID.

TIMEFMT(*sublist*)
Optional keyword you use to define the form in which time stamps appear in user-defined output. This specification overrides both the GENJCL default values and any values set on %SET statements in the input skeletal JCL.

The default for the GENJCL output time format is compressed, with a two-digit year, and the offset in numeric form: 960021315001 +0700. If you want the output time stamps to appear without offsets, for example, you can override the default with TIMEFMT(,N).

The override is good only for the duration of a single GENJCL command.

USERKEYS(*%key1,'value'* | *%key2*)

Optional keyword you use to set the value of keywords you have defined. Up to 64 keywords can be specified.

%key1
User-defined keyword that is being assigned a value. The maximum length of the keyword is 8 characters, including the percent sign. The first character after the percent sign must be alphabetic (A-Z). The remaining characters must be alphanumeric (A-Z, 0-9).

'value'
Value assigned to the user-defined keyword when it is encountered. *value* can be any character string enclosed in single quotation marks. The maximum length of *value* is 132 characters (excluding the quotation marks). If *value* contains a quotation mark, use two single quotation marks. *value* can be a null string (''). If *value* is a time stamp, it can be 0.

%key2
Any simple keyword that was previously assigned a value, including DBRC-defined and user-defined keywords.

Any keyword can be assigned a value with the USERKEYS keyword. However, if you assign a value to DBRC-defined keywords, the value is ignored.

If a keyword is assigned a value in both the USERKEYS and DEFAULTS keyword, the value specified in USERKEYS is used.

Examples

Here are some examples of using the GENJCL.CLOSE command.

Example when a host operating system failed and /ERE is not possible

In this example, a GENJCL.CLOSE command generates the JCL and control statements that are required to run the Log Recovery utility for the IMS online subsystem with subsystem ID IMSA, which was using a primary OLDS when a host operating system failed and /ERE could not be performed. Default skeletal member LOGCLJCL is taken from the data set identified in the JCLPDS DD statement. Output from the generated JCL goes to the data set identified in the JCLOUT DD statement. Skeletal member JOBJCL produces a job statement.

```
//GENJCL   JOB
//JCLOUT   DD
//JCLPDS   DD

.
.
.
//SYSIN    DD   *
     GENJCL.CLOSE SSID(IMSA)
/*
```

After the close job runs, the PRIOLDS record in the RECON data set that corresponds to the OLDS is updated to indicate the successful close.

Example using the CLOSE1 JCLPDS member

In this example, the GENJCL.CLOSE command is generated with a skeletal JCL execution member CLOSE1, which is taken from the JCLPDS data set identified in the PDS DD statement. Output from the generated JCL goes to the data set identified in the OUT DD statement. MEMBER DEFCL1 from the JCLPDS DD statement contains values to resolve user-defined keywords in member CLOSE1. Skeletal member JOBJCL produces a job statement. Generated JCL is listed.

```
//GENJCL1 JOB
//OUT     DD   . . .
//PDS     DD   . . .
//SYSIN   DD   *
GENJCL.CLOSE MEMBER(CLOSE1) OLDS(DFSOLP01) -
JCLPDS(PDS) JCLOUT(OUT) DEFAULTS(DEFCL1) LIST
```

Related reference:

"Generating JCL and user-defined output" on page 223

"DBRC time stamps" on page 52

"Skeletal JCL syntax" on page 225

GENJCL.IC command

Use the GENJCL.IC command to generate the JCL and utility control statements needed to run the Database Image Copy utility or the Database Image Copy 2 utility.

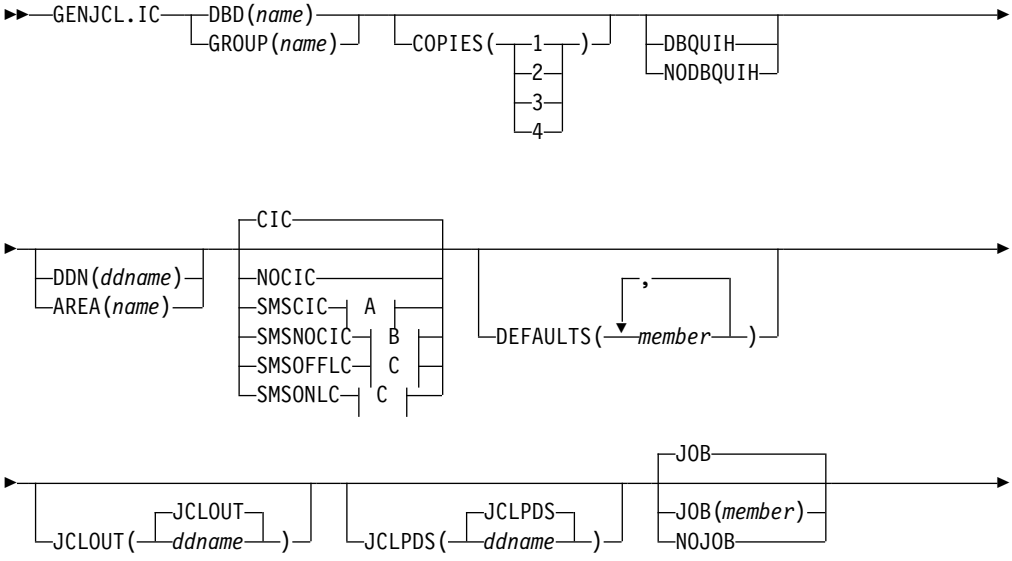
Important:

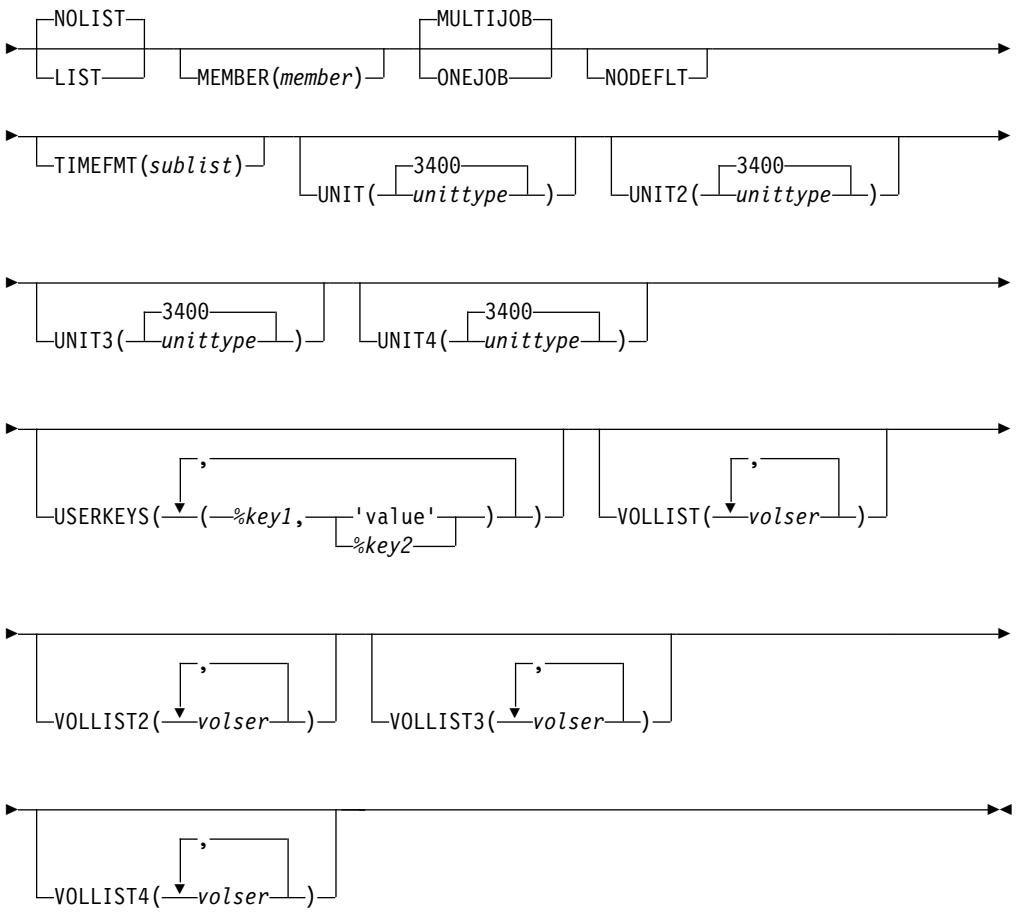
- For HALDB partitions, the GENJCL.IC command treats ILDS and index data sets differently than data DBDSs. The GENJCL.IC command skips these data sets in groups, regardless of whether the groups are explicit or implicit. If you explicitly specify one of these data sets, the GENJCL.IC command fails.
- For HALDB DBDSs, the GENJCL.IC command does not generate DD statements for the DBDSs to be copied. The SYSIN control statement identifies which active DBDSs to copy. These can be either the A-through-J data set or the M-through-V data sets.

Subsections:

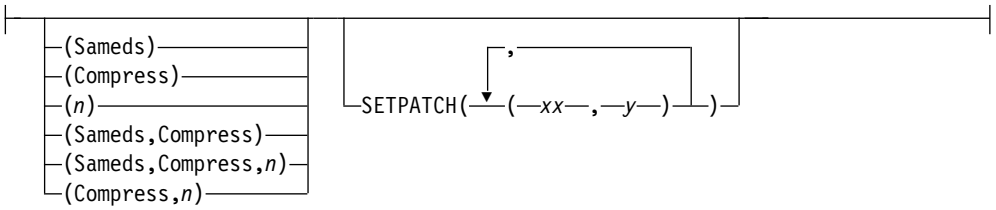
- "Syntax"
- "Keywords" on page 191
- "Examples" on page 201

Syntax

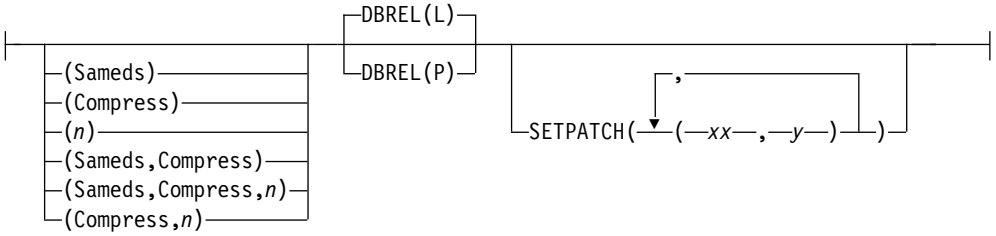



NOLIST
LIST
MEMBER(member)
MULTIJOB
ONEJOB
NODEFLT
TIMEFMT(sublist)
UNIT(3400 unittype)
UNIT2(3400 unittype)
UNIT3(3400 unittype)
UNIT4(3400 unittype)
USERKEYS(, (%key1, 'value' %key2))
VOLLIST(, volser)
VOLLIST2(, volser)
VOLLIST3(, volser)
VOLLIST4(, volser)

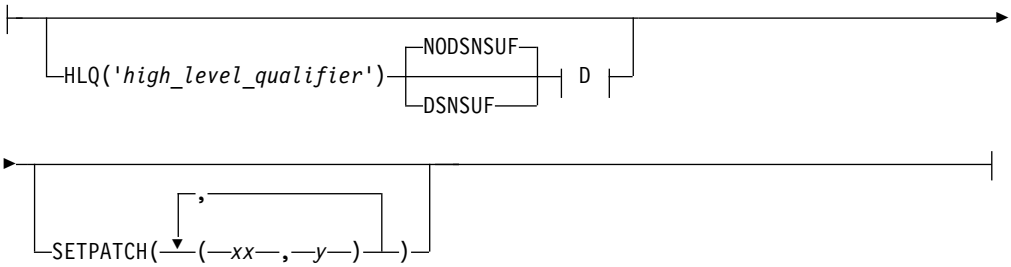
A:


(Sameds)
(Compress)
(n)
(Sameds,Compress)
(Sameds,Compress,n)
(Compress,n)
SETPATCH(, (xx, y))

B:


DBREL(L)
DBREL(P)
(Sameds)
(Compress)
(n)
(Sameds,Compress)
(Sameds,Compress,n)
(Compress,n)
SETPATCH(, (xx, y))

C:



D:

Keywords

DBD(*name*) | GROUP(*name*)
Mutually exclusive, required keywords that you use to specify the database or DBDS group for which DBDSs are to be copied.

DBD(*name*)
Specifies the name of the database that contains the DBDS or area to be copied.

If DDN is omitted and ONEJOB is specified for a SMSCIC or SMSNOCIC image copy, a utility control statement specifying the DBD name as the group name is generated in the JCL.

Restriction for HALDB databases: For the GENJCL.IC command, *name* specifies the name of either a HALDB master database name or a HALDB partition name.

GROUP(*name*)
Specifies that all DBDSs of a DBDS group are to be copied. If GROUP is specified, the GENJCL.IC command executes repeatedly for each DBDS of the named DBDS group.

If GROUP and ONEJOB are specified for an SMSCIC or SMSNOCIC image copy, a utility control statement specifying the group name is generated in the JCL.

COPIES(1 | 2 | 3 | 4)
Optional keyword you use to specify how many image copy data sets are to be produced for the specified DBDS.

If the specified DBDS is identified in the RECON data set with the NOREUSE attribute, you can specify the COPIES keyword if you want two or more image copy data sets; otherwise, one image copy data set is produced. COPIES(3 | 4) can only be specified if either SMSCIC or SMSNOCIC is also specified. The third and fourth copies are not recorded in the RECON data set.

If the specified DBDS is identified in the RECON data set with the REUSE attribute, the COPIES keyword is not allowed. The number of copies is determined by the number of pre-defined image copy data sets specified on the INIT.IC command.

DFSMS fast replication does not support multiple output copies. If you specify either SMSOFFLC or SMSONLC with the GENJCL.IC command, you cannot specify values 2, 3, or 4 for COPIES.

DBQUIH | NODBQUIH
Mutually exclusive, optional keywords that you use to override the current database quiesce settings in the RECON data set.

The JCL generated for image copy or hardcoded by the user must specify DISP=SHR, because the online IMS subsystems can have the database data sets allocated during a database quiesce. DBRC generates DISP=SHR for all image copies except batch image copies. For batch image copies, the DISP is set to DISP=OLD. When the QUIESCE HELD flag is set on in the RECON data set, DBRC will generate JCL with DISP=SHR instead of DISP=OLD. The following keywords can be used to override the current settings in the RECON data set.

DBQUIH
Specifies that the database will be quiesced when the JCL is executed. DISP=SHR is set in the generated JCL.

NODBQUIH
Specifies that the database will not be quiesced when the JCL is executed. DISP=OLD is set in the generated JCL.

If neither of these keywords are specified, DBRC will set the DISP value for batch image copies based on the status of the QUIESCE HELD flag at the time the JCL is generated. If the QUIESCE HELD flag is on in the DB or AREA record in the RECON data set, DISP=SHR is set. If the QUIESCE HELD flag is off, DISP=OLD is set.

DDN(*ddname*) | AREA(*name*)
Mutually exclusive, optional keywords that you use to identify the DBDS ddname or DEDB area name that is to be copied.

Specify the DDN or AREA keyword only if you specify the DBD keyword. If you do not specify DDN or AREA, the GENJCL.IC command executes repeatedly, once for each DBDS or area of the specified database. If you specify a HALDB master name, the GENJCL.IC command is performed for all data DBDSs for each partition in the HALDB master database. If you specify a HALDB partition name, the GENJCL.IC command is performed for all data DBDSs of the identified partition.

For HALDB databases, you must specify a partition database name with the DBD keyword in order to use the DDN keyword. The DDN keyword value is the partition DDN. The GENJCL.IC command is performed for the identified DBDS of the partition. The GENJCL.IC command fails if DDN does not identify a data DBDS in the partition.

CIC | NOCIC | SMSCIC(Sameds | Compress | *n* | any combination) | SMSNOCIC(Sameds | Compress | *n* | any combination) DBREL(L | P) | SMSOFFLC | SMSONLC
Optional keywords that you use to indicate how the image copy is to be taken.

Restriction: CIC and SMSCIC cannot be used for nonrecoverable or user-recoverable databases.

CIC
Specifies that the Database Image Copy (DFSUDMP0) utility is to be used to take an image copy. If CIC is specified, the copy is concurrent with update processing.

NOCIC
Specifies that the Database Image Copy (DFSUDMP0) utility is to be used to take an image copy. If NOCIC is specified, the copy is made while the database is unavailable for update processing.

SMSCIC(Sameds | Compress | *n* | any combination) | SETPATCH
Indicates that the Database Image Copy 2 utility is to be used to take an image copy concurrent with update processing. The Database Image Copy 2 utility invokes DFSMSdss to dump the database data set or sets using the Concurrent Copy option.

Sameds
An optional keyword you use to specify that all the image copies generated by the job are to be written to a single output data set. This option is not valid if any of the DBDs or areas have the REUSE attribute, if ONEJOB is not specified, or if more than 255 data sets one or the other or both areas are to be copied. You can abbreviate this option when specifying it. When a DBDS group (explicit or implicit) is specified, this option applies to all members of the group.

Compress
An optional keyword that directs the utility to use compression. The compress option enables you to reduce the storage space required to hold the image copy; however, using the compress option increases the CPU time that is required to perform the copy operation. You can abbreviate this option when specifying it. When a DBDS group (explicit or implicit) is specified, this option applies to all members of the group.

The utilities request compression services from DFSMSdss. When the Image Copy 2 utility requests compression services, it issues the DFSMSdss DUMP command with the COMPRESS ZCOMPRESS(PREFERRED) options, so that if z/Enterprise Data Compression (zEDC) services are enabled in your z/OS system, zEDC is used for compression. zEDC uses less CPU and improved compression algorithms.

n An optional keyword you use to specify the value to be used for the DFSMSdss OPTIMIZE keyword. *n* is a number from 1 to 4 where:

- If *n* is 1, DFSMSdss reads one track at a time.
- If *n* is 2, DFSMSdss reads two tracks at a time.
- If *n* is 3, DFSMSdss reads five tracks at a time.
- If *n* is 4, DFSMSdss reads one cylinder at a time.

If *n* is not specified, the default is 1 for SMSCIC and 4 for SMSNOCIC. Note that the higher values use more real and virtual storage and keeps the channel busier for longer blocks of time. You can abbreviate this option when specifying it. When a DBDS group (explicit or implicit) is specified, this option applies to all members of the group.

SETPATCH((*xx1*,*y1*),(*xx2*,*y2*) ... (*xxn*,*yn*))
A keyword that specifies DFSMSdss processing options by defining the SET PATCH offsets that the Image Copy 2 utility passes to DFSMSdss. SETPATCH is applicable only when SMSCIC, SMSNOCIC, SMSONLC or SMSOFFLC is specified.

Offsets *xx1* through *xxn* are patch bytes, and *y1* through *yn* are the values to which the corresponding patch byte is to be set. Up to 32 pairs of patch bytes and values can be supplied.

SMSNOCIC(Sameds | Compress | *n* | any combination) | DBREL(L | P) | SETPATCH

Indicates that the Database Image Copy 2 utility is to be used to take an image copy while the database is unavailable for update processing. The Database Image Copy 2 utility invokes DFSMSdss to dump the database data set(s) using the Concurrent Copy option.

Sameds

An optional keyword you use to specify that all the image copies generated by the job are to be written to a single output data set. This option is not valid if any of the DBDs or areas have the REUSE attribute, if ONEJOB is not specified, or if more than 255 data sets one or the other or both areas are to be copied. You can abbreviate this option when specifying it. When a DBDS group (explicit or implicit) is specified, this option applies to all members of the group.

Compress

An optional keyword that directs the utility to use compression. The compress option enables you to reduce the storage space required to hold the image copy; however, using the compress option increases the CPU time that is required to perform the copy operation. You can abbreviate this option when specifying it. When a DBDS group (explicit or implicit) is specified, this option applies to all members of the group.

The utilities request compression services from DFSMSdss. When the Image Copy 2 utility requests compression services, it issues the DFSMSdss DUMP command with the COMPRESS ZCOMPRESS(PREFERRED) options, so that if z/Enterprise Data Compression (zEDC) services are enabled in your z/OS system, zEDC is used for compression. zEDC uses less CPU and improved compression algorithms.

n An optional keyword you use to specify the value to be used for the DFSMSdss OPTIMIZE keyword. *n* is a number from 1 to 4 where:

- If *n* is 1, DFSMSdss reads one track at a time.
- If *n* is 2, DFSMSdss reads two tracks at a time.
- If *n* is 3, DFSMSdss reads five tracks at a time.
- If *n* is 4, DFSMSdss reads one cylinder at a time.

If *n* is not specified, the default is 1 for SMSCIC and 4 for SMSNOCIC. Note that the higher values use more real and virtual storage and keeps the channel busier for longer blocks of time. You can abbreviate this option when specifying it. When a DBDS group (explicit or implicit) is specified, this option applies to all members of the group.

DBREL(L | P)

Indicates when the database is to be made available for update processing. DBREL is applicable only when SMSNOCIC is also specified. IMS ignores DBREL when it is specified with SMSCIC.

L L indicates that updates are to be allowed after the image copy is logically complete (after DFSMS has initialized a concurrent copy session). Update processing can occur (or be resumed) before the image copy is physically complete.

P P indicates that updates are not to be allowed until the image copy is physically complete.

SETPATCH((*xx1*,*y1*),(*xx2*,*y2*) ... (*xxn*,*yn*))
A keyword that specifies DFSMSdss processing options by defining the SET PATCH offsets that the Image Copy 2 utility passes to DFSMSdss. SETPATCH is applicable only when SMSCIC, SMSNOCIC, SMSONLC or SMSOFFLC is specified.

Offsets *xx1* through *xxn* are patch bytes, and *y1* through *yn* are the values to which the corresponding patch byte is to be set. Up to 32 pairs of patch bytes and values can be supplied.

SMSOFFLC
Indicates that the Database Image Copy utility is to be used to take an image copy while the database is unavailable for update processing. This option directs Database Image Copy 2 to invoke DFSMS Fast Replication.

HLQ(*high_level_qualifier***)**
Indicates the data set high-level qualifier to be used to create the output data set name, when the output data sets are created dynamically, by appending the dbdname and the DD name to the data set high-level qualifier (HLQ). HLQ is applicable only for SMSONLC and SMSOFFLC type copies. The value specified for HLQ must be surrounded by single quotation marks (').

VOLLIST must be specified if HLQ is specified and the target data set is non-SMS managed. If the VOLLIST keyword is specified in addition to the HLQ keyword, a DD statement is generated for the image copy data set and the specified volumes are included, but not a data set name for the image copy data set. The ddname for this DD statement is included in the OUTPUT keyword in the DBDS Select control statement. If no VOLLIST keyword is specified, neither a DD statement for the image copy data set nor a value for the OUTPUT keyword is included in the generated JCL.

If the DSNSUF keyword is also specified, a time stamp is appended as the low-level qualifier to the output data set name. The combination of HLQ value, dbdname, dsname, time stamp and the delimiters between qualifiers cannot exceed 44 characters in length.

If the HLQ keyword is omitted, the output data set name is generated as indicated by the skeletal JCL execution member used by the GENJCL.IC command.

Restriction: HLQ cannot be specified unless SMSOFFLC or SMSONLC are specified. If HLQ is specified without either SMSOFFLC or SMSONLC also specified, the copy fails, and message DSP0191 with reason code 04 is generated.

NODSNSUF | DSNSUF
Mutually exclusive, optional keywords that you use to specify whether a date and time stamp is used as the low-level qualifier in the generated output data set name.

NODSNSUF
Indicates that a date and time stamp will not be incorporated as the low-level qualifier in the generated output data set name. This keyword is the default.

DSNSUF
Indicates that a date and time stamp will be incorporated as the

low-level qualifier in the generated output data set name. Its format is *Dyyddd.Thhmmss.* The date and time are always used together if DSNSUF is specified.

Restrictions:

- DSNSUF cannot be specified unless HLQ is specified. If DSNSUF is specified without HLQ also specified, the copy will fail, and message DSP0191 with reason code 03 will be generated.
- DSNSUF cannot be specified unless SMSOFFLC or SMSONLC are specified. If DSNSUF is specified without either SMSOFFLC or SMSONLC also specified, the copy will fail, and message DSP0191 with reason code 04 will be generated.

MGMTCLAS(*management_class***)**
Optional keyword that indicates the management class that you want to use for the allocation of the image copy data set. This specification is used as input to the Access Control System (ACS) routines. You must have proper RACF authorization for the specified management class. This keyword is applicable only when SMSONLC or SMSOFFLC and HLQ are specified.

SETPATCH((*xx1*,*y1*),(*xx2*,*y2*) ... (*xxn*,*yn*))
A keyword that specifies DFSMSdss processing options by defining the SET PATCH offsets that the Image Copy 2 utility passes to DFSMSdss. SETPATCH is applicable only when SMSCIC, SMSNOCIC, SMSONLC or SMSOFFLC is specified.

Offsets *xx1* through *xxn* are patch bytes, and *y1* through *yn* are the values to which the corresponding patch byte is to be set. Up to 32 pairs of patch bytes and values can be supplied.

STORCLAS(*storage_class***)**
Optional keyword that indicates the storage class that you want to use for the allocation of the image copy data set. This specification is used as input to the ACS routines. You must have proper RACF authorization for the specified storage class. This keyword is applicable only when the Database Image Copy utility is used to invoke DFSMS Fast Replication, and the target data set is SMS managed.

This keyword is applicable only when SMSONLC or SMSOFFLC and HLQ are specified.

SMSONLC
Indicates that the Database Image Copy utility is to be used to take an image copy concurrent with update processing. This option directs Database Image Copy 2 to invoke DFSMS Fast Replication.

HLQ(*high_level_qualifier***)**
Indicates the data set high-level qualifier to be used to create the output data set name, when the output data sets are created dynamically, by appending the dbdname and the DD name to the data set high-level qualifier (HLQ). HLQ is applicable only for SMSONLC and SMSOFFLC type copies. The value specified for HLQ must be surrounded by single quotation marks (').

VOLLIST must be specified if HLQ is specified and the target data set is non-SMS managed. If the VOLLIST keyword is specified in addition to the HLQ keyword, a DD statement is generated for the image copy

data set and the specified volumes are included, but not a data set name for the image copy data set. The ddname for this DD statement is included in the OUTPUT keyword in the DBDS Select control statement. If no VOLLIST keyword is specified, neither a DD statement for the image copy data set nor a value for the OUTPUT keyword is included in the generated JCL.

If the DSNSUF keyword is also specified, a time stamp is appended as the low-level qualifier to the output data set name. The combination of HLQ value, dbdname, dsname, time stamp and the delimiters between qualifiers cannot exceed 44 characters in length.

If the HLQ keyword is omitted, the output data set name is generated as indicated by the skeletal JCL execution member used by the GENJCL.IC command.

Restriction: HLQ cannot be specified unless SMSOFFLC or SMSONLC are specified. If HLQ is specified without either SMSOFFLC or SMSONLC also specified, the copy fails, and message DSP0191 with reason code 04 is generated.

NODSNSUF | DSNSUF

Mutually exclusive, optional keywords that you use to specify whether a date and time stamp is used as the low-level qualifier in the generated output data set name.

NODSNSUF

Indicates that a date and time stamp will not be incorporated as the low-level qualifier in the generated output data set name. This keyword is the default.

DSNSUF

Indicates that a date and time stamp will be incorporated as the low-level qualifier in the generated output data set name. Its format is *Dyyddd.Thhmmss.* The date and time are always used together if DSNSUF is specified.

Restrictions:

- DSNSUF cannot be specified unless HLQ is specified. If DSNSUF is specified without HLQ also specified, the copy will fail, and message DSP0191 with reason code 03 will be generated.
- DSNSUF cannot be specified unless SMSOFFLC or SMSONLC are specified. If DSNSUF is specified without either SMSOFFLC or SMSONLC also specified, the copy will fail, and message DSP0191 with reason code 04 will be generated.

MGMTCLAS(*management_class*)

Optional keyword that indicates the management class that you want to use for the allocation of the image copy data set. This specification is used as input to the Access Control System (ACS) routines. You must have proper RACF authorization for the specified management class. This keyword is applicable only when SMSONLC or SMSOFFLC and HLQ are specified.

SETPATCH((*xx1*,*y1*),(*xx2*,*y2*) ... (*xxn*,*yn*))

A keyword that specifies DFSMSdss processing options by defining the

SET PATCH offsets that the Image Copy 2 utility passes to DFSMSdss. SETPATCH is applicable only when SMSCIC, SMSNOCIC, SMSONLC or SMSOFFLC is specified.

Offsets *xx1* through *xxn* are patch bytes, and *y1* through *yn* are the values to which the corresponding patch byte is to be set. Up to 32 pairs of patch bytes and values can be supplied.

STORCLAS(*storage_class***)**
Optional keyword that indicates the storage class that you want to use for the allocation of the image copy data set. This specification is used as input to the ACS routines. You must have proper RACF authorization for the specified storage class. This keyword is applicable only when the Database Image Copy utility is used to invoke DFSMS Fast Replication, and the target data set is SMS managed.

This keyword is applicable only when SMSONLC or SMSOFFLC and HLQ are specified.

DEFAULTS(*member***)**
Optional keyword that you use to specify up to 10 names of skeletal JCL default members to be used when generating JCL. Default members are searched to resolve keywords in the order in which the members are specified on this keyword.

If a keyword is assigned a value in both the DEFAULTS and USERKEYS keyword, the value specified in USERKEYS is used.

JCLOUT(JCLOUT | *ddname*)
Optional keyword that you use to specify the output data set for the generated JCL. The data set is specified by *ddname*. A JCL DD statement with this ddname must be included in the job step containing the GENJCL command. The specified data set can be a member of a partitioned data set, but only if it is not the same data set that is used for the default, JCLOUT.

JCLPDS(JCLPDS | *ddname*)
Optional keyword that you use to specify the skeletal JCL data set that is to be used for input when generating JCL. The data set is specified by *ddname*. A JCL DD statement with this ddname must be included in the job step containing the GENJCL command.

JOB | JOB(*member*) | NOJOB
Mutually exclusive, optional keywords that you use to specify whether to produce the job statement in the generated JCL.

JOB
Specifies that the job statement is to be produced. When JOB is specified without a member name, the IBM-supplied execution member JOBJCL produces the job statement. When JOB(*member*) is specified, the specified execution member produces the job statement.

NOJOB
Specifies that the job statement is not to be produced in the generated JCL.

NOLIST | LIST
Mutually exclusive, optional keywords that you use to specify whether to write the generated JCL to the SYSPRINT data set.

NOLIST
Suppresses the printing of the generated JCL.

LIST
Prints the generated JCL.

MEMBER(*member*)
Optional keyword you use to specify the name of the skeletal JCL execution member to be used. If this keyword is not specified, the default specified on the INIT.DBDS command is used.

MULTIJOB | ONEJOB
Mutually exclusive, optional keywords that you use to control how many JOB and EXEC statements are to be generated when a DBDS group is specified either explicitly or implicitly.

MULTIJOB
Produces one job for each group member.

ONEJOB
Produces one job with:

- A separate step for each group member with the CIC or NOCIC option.
- A single step to copy all members of the group with the SMSCIC or SMSNOCIC option.

These keywords are invalid if the NOJOB subparameter is specified on the JOB keyword, or if a DBDS group is not specified.

NODEFLT
Optional keyword you use to specify that the implicit skeletal JCL default member, if any, for the DBDS is not to be used.

TIMEFMT(*sublist*)
Optional keyword you use to define the form in which time stamps appear in user-defined output. This specification overrides both the GENJCL default values and any values set on %SET statements in the input skeletal JCL.

The default for the GENJCL output time format is compressed, with a two-digit year, and the offset in numeric form: 960021315001 +0700. If you want the output time stamps to appear without offsets, for example, you can override the default with TIMEFMT(,N).

The override is good only for the duration of a single GENJCL command.

UNIT(3400 | *unittype*)
Optional keyword you use to specify the unit type of the first, second, third, or fourth output image copy data sets. These keywords are valid only if both of the following conditions are true:

- The corresponding VOLLIST keyword is specified
- The DBDS for which the JCL is being generated was defined with the NOREUSE option

If this keyword is specified, its value is associated with DD statement D1OU1 in the generated JCL and the OUTPUT keyword in the control statements for the Database Image Copy 2 utility. If the GROUP keyword is also specified, all of the DBDS Select control statements following the Group Name control statement will contain D1OU1 as the OUTPUT keyword.

UNIT2(3400 | *unittype*)
Optional keyword you use to specify the unit type of the first, second, third, or fourth output image copy data sets. These keywords are valid only if both of the following conditions are true:

- The corresponding VOLLIST keyword is specified

- The DBDS for which the JCL is being generated was defined with the NOREUSE option

This keyword cannot be specified if the user selects DFSMS fast replication by specifying either SMSOFFLC or SMSONLC.

UNIT3(3400 | *unittype*)

Optional keyword you use to specify the unit type of the first, second, third, or fourth output image copy data sets. These keywords are valid only if both of the following conditions are true:

- The corresponding VOLLIST keyword is specified
- The DBDS for which the JCL is being generated was defined with the NOREUSE option

This keyword cannot be specified if the user selects DFSMS fast replication by specifying either SMSOFFLC or SMSONLC.

UNIT4(3400 | *unittype*)

Optional keyword you use to specify the unit type of the first, second, third, or fourth output image copy data sets. These keywords are valid only if both of the following conditions are true:

- The corresponding VOLLIST keyword is specified
- The DBDS for which the JCL is being generated was defined with the NOREUSE option

This keyword cannot be specified if the user selects DFSMS fast replication by specifying either SMSOFFLC or SMSONLC

USERKEYS(*%key1*,*'value'* | *%key2*)

Optional keyword you use to set the value of keywords you have defined. Up to 64 keywords can be specified.

%key1
User-defined keyword that is being assigned a value. The maximum length of the keyword is 8 characters, including the percent sign. The first character after the percent sign must be alphabetic (A-Z). The remaining characters must be alphanumeric (A-Z, 0-9).

'value'
Value assigned to the user-defined keyword when it is encountered. *value* can be any character string enclosed in single quotation marks. The maximum length of *value* is 132 characters (excluding the quotation marks). If *value* contains a quotation mark, use two single quotation marks. *value* can be a null string (''). If *value* is a time stamp, it can be 0.

%key2
Any simple keyword that was previously assigned a value, including DBRC-defined and user-defined keywords.

Any keyword can be assigned a value with the USERKEYS keyword. However, if you assign a value to DBRC-defined keywords, the value is ignored.

If a keyword is assigned a value in both the USERKEYS and DEFAULTS keyword, the value specified in USERKEYS is used.

VOLLIST(*volser*)

Optional keywords that you use to specify the volumes on which the image copy data set copies are to reside. These keywords are valid only if:

- The DBDS for which the JCL is being generated is defined with the NOREUSE option.

- DBD is specified with DDN or AREA.

If you specify this keyword, its value is associated with DD statement D1OU1 in the generated JCL and the OUTPUT keyword in the control statements for the Database Image Copy 2 utility. If also specify the GROUP keyword, all of the DBDS Select control statements following the Group Name control statement will contain D1OU1 as the OUTPUT parameter.

VOLLIST2(*volser*)
Optional keywords that you use to specify the volumes on which the image copy data set copies are to reside. These keywords are valid only if:

- The DBDS for which the JCL is being generated is defined with the NOREUSE option.
- DBD is specified with DDN or AREA.

This keyword cannot be specified if the user selects DFSMS fast replication by specifying either SMSOFFLC or SMSONLC.

VOLLIST3(*volser*)
Optional keywords that you use to specify the volumes on which the image copy data set copies are to reside. These keywords are valid only if:

- The DBDS for which the JCL is being generated is defined with the NOREUSE option.
- DBD is specified with DDN or AREA.

This keyword cannot be specified if the user selects DFSMS fast replication by specifying either SMSOFFLC or SMSONLC.

VOLLIST4(*volser*)
Optional keywords that you use to specify the volumes on which the image copy data set copies are to reside. These keywords are valid only if:

- The DBDS for which the JCL is being generated is defined with the NOREUSE option.
- DBD is specified with DDN or AREA.

This keyword cannot be specified if the user selects DFSMS fast replication by specifying either SMSOFFLC or SMSONLC.

Examples

Here are some examples of using the GENJCL.IC command.

Example for DBDS defined by the DBD and DDN keywords

In the following example, a GENJCL.IC command generates the JCL and control statements required to run the Database Image Copy utility for the DBDS identified in the DBD and DDN keywords. The default Concurrent Image Copy (CIC) is used. The database is defined as REUSE. If the INIT.DBDS command for the DBDS for which JCL is being generated is specified with ICJCL(*member*), that skeletal member is used from the data set that is identified in the JCLPDS DD statement. If not, the default skeletal member ICJCL from the JCLPDS data set is used. Output from the generated JCL goes to the data set identified in the JCLOUT DD statement. Skeletal member JOBJCL produces a job statement.

```
//GENJIC1  JOB
//JCLOUT   DD   . . .
//JCLPDS   DD   . . .
```

```
.
.
.
//SYSIN    DD   *
     GENJCL.IC   DBD(DBDKSDS1) DDN(DDNKSDS1)
```

The EXEC and SYSIN statements for the generated JCL are shown below:

```
.
.
.
//IC1   EXEC  PGM=DFSUDMP0,REGION=nnnK,
//              PARM='CIC,GSGNAME='
//D1OU1  DD  DSN=...

.
.
.
//SYSIN   DD *
D1 DBDKSDS1 DDNKSDS1 D1OU1
/*
```

Example for all DBDSs in a group with NOCIC

In the following example, the GENJCL.IC command generates JCL and control statements to run the Image Copy utility for all DBDSs of GROUP1 and batch image copies (NOCIC) are also taken. The skeletal member used is ICJCL1 from the data set identified in PDS4. The keyword, %DEFIC, is a user-defined value in ICJCL1 that is resolved to '1ST USERKEYS PARM'. The default member for the database initialized in INIT.DBDS DEFLTJCL(MEMBER) is not used to resolve keywords. Skeletal member JOBJCL produces a job statement.

```
//GENJIC2 JOB
//JCLOUT  DD   . . .
//PDS4    DD   . . .
//SYSIN   DD   *
  GENJCL.IC GROUP(GROUP1) JOB MEMBER(ICJCL1) JCLPDS(PDS4) ONEJOB -
  NOCIC USERKEYS((%DEFIC,'1ST USERKEYS PARM')) NODEFLT
```

The following statements are examples of one of the EXEC statements and one of the SYSIN statements for the generated JCL:

```
.
.
.
//IC1   EXEC  PGM=DFSUDMP0,REGION=nnnK,
//              PARM=',GSGNAME='
//D1OU1  DD  DSN=...

.
.
.
//SYSIN   DD *
D1 DBD1GRP1 DDN1GRP1 D1OU1
/*
//IC2   EXEC  PGM=DFSUDMP0,REGION=nnnK,
//              PARM=',GSGNAME='

.
.
.
```

Example of running the Database Image Copy 2 utility with SMSCIC

In the following example, a GENJCL.IC command generates the JCL and control statements required to run the Database Image Copy 2 utility in shared database mode (SMSCIC) for the DBDS that is identified in the DBD and DDN keywords. The database is defined as NOREUSE and four copies are requested. The default skeletal member ICJCL from the JCLPDS data set is used. Output from the generated JCL goes to the data set identified in the JCLOUT DD statement. Skeletal member JOBJCL produces a job statement.

```
//GENJIC3  JOB
//JCLOUT   DD   . . .
//JCLPDS   DD   . . .

⋮
//SYSIN    DD   *
  GENJCL.IC   DBD(DBDVSAM1) DDN(DDNVSAM1) COPIES(4) SMSCIC(1)      -
              VOLLIST(IC2001) VOLLIST2(IC2002) VOLLIST3(IC2003) -
              VOLLIST4(IC2004)
/*
```

The EXEC and SYSIN statements for the generated JCL are shown below:

```
⋮
//IC1   EXEC  PGM=DFSRRC00,REGION=nnnK,
//              PARM='ULU,DFSUDMT0,,,,,,,,,,,Y,,,,,,,'

⋮
//SYSIN   DD *
 4 DBDVSAM1 DDNVSAM1 D1OU1    D2OU1    D3OU1    D4OU1   S  1
/*
```

Example of running the Database Image Copy 2 utility with SMSNOCIC

In the following example, the GENJCL.IC command generates JCL and control statements to run the Database Image Copy 2 utility with exclusive database usage (SMSNOCIC) for the DBDS that is identified in the DBD and DDN keywords. The database is defined as REUSE and the default, one copy, is requested. The default skeletal member ICJCL from the JCLPDS data set is used. Output from the generated JCL goes to the data set identified in the JCLOUT DD statement. Skeletal member JOBJCL produces a job statement. The database is to be unlocked after the physical copy completes.

```
//GENJIC4  JOB
//JCLOUT   DD   . . .
//JCLPDS   DD   . . .

⋮
//SYSIN    DD   *
  GENJCL.IC   DBD(DBDVSAM2) DDN(DDNVSAM2) DBREL(P) SMSNOCIC
/*
```

The EXEC and SYSIN statements for the generated JCL are shown below:

```
⋮
//IC1   EXEC  PGM=DFSRRC00,REGION=nnnK,
//              PARM='ULU,DFSUDMT0,,,,,,,,,,,Y,,,,,,,'

⋮
//SYSIN   DD *
 1 DBDVSAM2 DDNVSAM2 DATAOUT1                            XP
/*
```

Example of running the Database Image Copy 2 utility with an implied database group

All data sets for the database will be copied in a single step to a single output data set, with the Compress option.

```
//GENJIC5    JOB
//JCLOUT     DD   . . .
//JCLPDS     DD   . . .
```

```
 .
 .
 .
//SYSIN      DD   *
  GENJCL.IC   DBD(DBDVSAM) SMSCIC(SAMEDS,COMPRESS) ONEJOB
/*
```

The following are the EXEC and SYSIN cards in the generated JCL:

```
:
//IC1  EXEC PGM=DFSRRC00,REGION=800K,
//            PARM='ULU,DFSUDMT0,,,,,,,,,,,Y,,,,,,,,'
//D1OU0  DD  DSN=...
//SYSIN DD *
G  DHVNTZ02                                          S C1
 1 DHVNTZ02 HIDAM    D1OU0
S  DHVNTZ02 HIDAM2                                   /*
```

Example 1 of running the Database Image Copy utility with a fast replication (non-SMS-managed)

```
GENJCL.IC DBD(DB1) DDN(DBDS1) SMSONLC VOLLIST(VOLXXX)
//ICJCL    EXEC PGM=DFSRRC00,PARM='ULU,DFSUDMT0,,,,,,,,,,,Y,N'
//SYSPRINT DD   SYSOUT=A
//SYSUDUMP DD   SYSOUT=A
//RECON1   DD DSN=IMS.RECON1,DISP=SHR
//RECON2   DD DSN=IMS.RECON2,DISP=SHR
//RECON3   DD DSN=IMS.RECON3,DISP=SHR
//IMS      DD DSN=IMS.DBDLIB,DISP=SHR
//DBDS1    DD DSN=IMS.USER.DBDS1,DISP=SHR
//*
//D1OU1    DD DISP=(NEW,CATLG),
//   UNIT=SYSDA,VOL=SER=VOLXXX
//   SPACE=(TRK,(5,5),RLSE),
//   DSN=HLQ.DBDS1.COPY,
//DFSVSAMP DD   *
//SYSIN    DD   *
1 DB1      DBDS1    D1OU1                             S   F
```

This example generates comparable JCL and a control statement. There is a DD statement for the image copy data set (D1OU1). Because the target image copy data set is not SMS-managed, the target volume must be specified on the DD statement for the image copy data set.

Example 2 of running the Database Image Copy utility with a fast replication (SMS-managed)

```
GENJCL.IC DBD(DB1) DDN(DBDS1) SMSONLC HLQ(HLQ1)
//ICJCL    EXEC PGM=DFSRRC00,PARM='ULU,DFSUDMT0,,,,,,,,,,,Y,N'
//SYSPRINT DD   SYSOUT=A
//SYSUDUMP DD   SYSOUT=A
//RECON1   DD DSN=IMS.RECON1,DISP=SHR
//RECON2   DD DSN=IMS.RECON2,DISP=SHR
//RECON3   DD DSN=IMS.RECON3,DISP=SHR
//IMS      DD DSN=IMS.DBDLIB,DISP=SHR
//DBDS1    DD DSN=IMS.USER.DBDS1,DISP=SHR
//*
//DFSVSAMP DD   *
//SYSIN    DD   *
 1 DB1      DBDS1    D1OU1                             S   FH
H1 HLQ1
```

This example generates comparable JCL and a control statement where the name of the image copy data set is generated as HLQ1.DB1.DBDS1. There is no DD statement for the image copy data set on which a volume is specified; neither is there a storage class specification on the HLQ control statement. In this case, the source database data set must have been SMS-managed, and the storage class specification for the target image copy data set will be inherited from the attributes of the source database.

Example 3 of running the Database Image Copy utility with a fast replication (Image Copy is non-SMS-managed)

```
GENJCL.IC DBD(DB1) DDN(DBDS1) SMSOFFLC VOLLIST(VOLSER) HLQ('HLQ1') NODSNSUF
//ICJCL    EXEC PGM=DFSRRC00,PARM='ULU,DFSUDMT0,,,,,,,,,,,Y,N'
//SYSPRINT DD   SYSOUT=A
//SYSUDUMP DD   SYSOUT=A
//RECON1   DD DSN=IMS.RECON1,DISP=SHR
//RECON2   DD DSN=IMS.RECON2,DISP=SHR
//RECON3   DD DSN=IMS.RECON3,DISP=SHR
//IMS      DD DSN=IMS.DBDLIB,DISP=SHR
//DBDS1    DD DSN=IMS.USER.DBDS1,DISP=SHR
//*
//D1OU1    DD DISP=SHR,
//   UNIT=SYSDA,VOL=SER=VOLSER
//DFSVSAMP DD   *
//SYSIN    DD   *
 1 DB1      DBDS1    D1OU1                              X   FH
H1 HLQ1
```

This example generates comparable JCL and a control statement where the name of the image copy data set is generated as HLQ1.DB1.DBDS1. There is a DD statement for the image copy data set, but it does not contain the data set name. Because the target image copy data set is not SMS-managed, the target volume must be specified on the DD statement for the image copy data set.

Example 4 of running the Database Image Copy utility with a fast replication (Image Copy is SMS-managed)

```
GENJCL.IC DBD(DB1) DDN(DBDS1) SMSONLC STORCLAS(STORCLS1) HLQ('HLQ1')
NODSNSUF
//ICJCL    EXEC PGM=DFSRRC00,PARM='ULU,DFSUDMT0,,,,,,,,,,,Y,N'
//SYSPRINT DD   SYSOUT=A
//SYSUDUMP DD   SYSOUT=A
//RECON1   DD DSN=IMS.RECON1,DISP=SHR
//RECON2   DD DSN=IMS.RECON2,DISP=SHR
//RECON3   DD DSN=IMS.RECON3,DISP=SHR
//IMS      DD DSN=IMS.DBDLIB,DISP=SHR
//DBDS1    DD DSN=IMS.USER.DBDS1,DISP=SHR
//*
//DFSVSAMP DD   *
//SYSIN    DD   *
 1 DB1      DBDS1    D1OU1                              S   FH
H1 HLQ1                 STORCLS1
```

This example generates comparable JCL and a control statement where the name of the image copy data set is generated as HLQ1.DB1.DBDS1. There is no DD statement for the image copy data set. When the target image copy is SMS-managed, indicate the SMS storage class on the HLQ control statement.

Example 5 of running the Database Image Copy utility with a fast replication (SMS-managed)

```
GENJCL.IC DBD(DB1) DDN(DBDS1) SMSONLC HLQ(HLQ1) DSNSUF
//ICJCL    EXEC PGM=DFSRRC00,PARM='ULU,DFSUDMT0,,,,,,,,,,,Y,N'
//SYSPRINT DD   SYSOUT=A
//SYSUDUMP DD   SYSOUT=A
//RECON1   DD DSN=IMS.RECON1,DISP=SHR
//RECON2   DD DSN=IMS.RECON2,DISP=SHR
//RECON3   DD DSN=IMS.RECON3,DISP=SHR
//IMS      DD DSN=IMS.DBDLIB,DISP=SHR
//DBDS1    DD DSN=IMS.USER.DBDS1,DISP=SHR
//*
//DFSVSAMP DD   *
//SYSIN    DD   *
 1 DB1      DBDS1    D1OU1                              S   FH
H1YHLQ1
```

This example generates comparable JCL and a control statement where the name of the image copy data set is generated as HLQ1.DB1.DBDS1.Dyyddd.Thhmmss. There is no DD statement for the image copy data set on which a volume is specified; neither is there a storage class specification on the HLQ control statement. In this case, the source database data set must have been SMS-managed, and the storage class specification for the target image copy data set will be inherited from the attributes of the source database.

Related concepts:

z/OS: Tuning DFSMShsm with supported tuning patches

Related reference:

"Generating JCL and user-defined output" on page 223

"DBRC time stamps" on page 52

"Skeletal JCL syntax" on page 225

GENJCL.OIC command

Use the GENJCL.OIC command to generate the JCL and utility control statements needed to run the Online Database Image Copy utility.

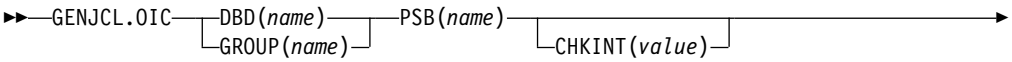
Important:

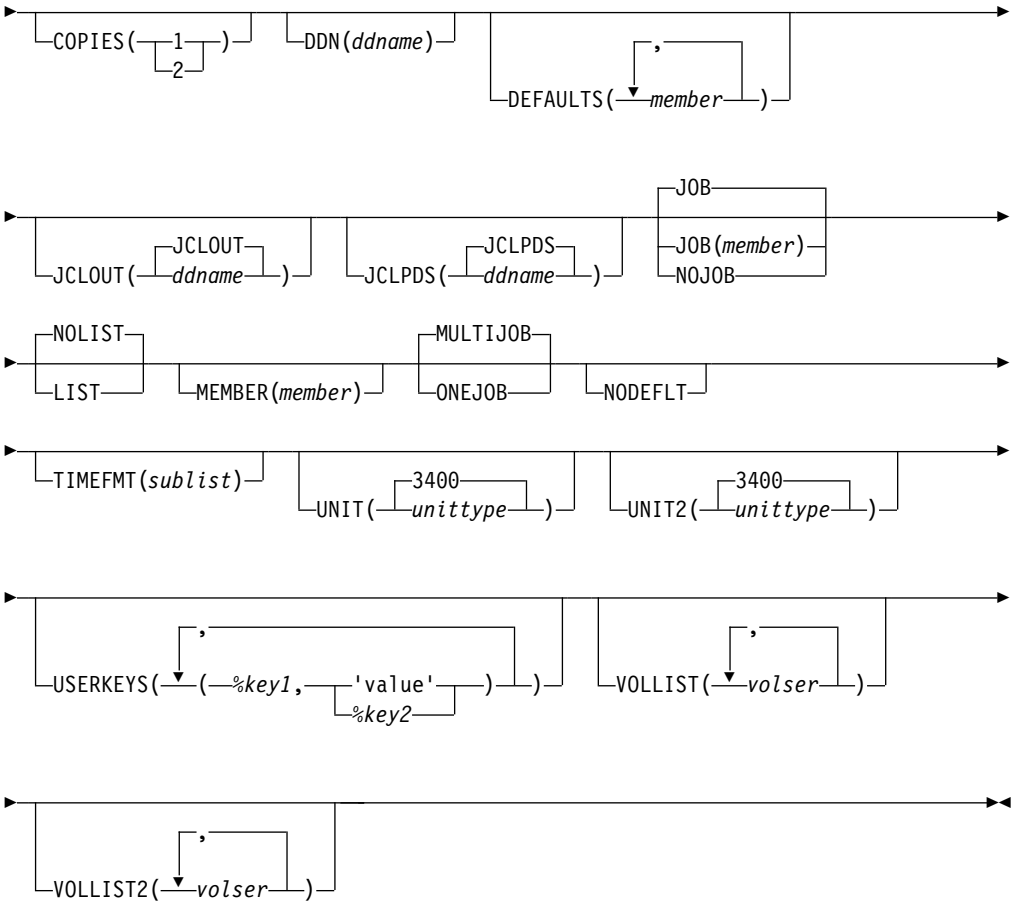
- For HALDB partitions, the GENJCL.OIC command treats ILDS and index data sets differently than data DBDSs. The GENJCL.OIC command skips these data sets in groups, regardless of whether the groups are explicit or implicit. If you explicitly specify one of these data sets, the GENJCL.OIC command fails.
- For HALDB DBDSs, the GENJCL.OIC command does not generate DD statements for the DBDSs to be copied. The SYSIN control statement identifies which active DBDSs to copy. These can be either the A-through-J data set or the M-through-V data sets.

Subsections:

- "Syntax"
- "Keywords" on page 207
- "Examples" on page 210

Syntax





Keywords

DBD(*name***) | GROUP(***name***)**
Mutually exclusive, required keywords that you use to specify the database that contains the DBDS to be copied.

DBD
Specifies the name of the database that contains the DBDS to be copied.

Restriction for HALDB databases: For the GENJCL.OIC command, *name* specifies the name of either a HALDB master database name or a HALDB partition name.

GROUP
Specifies that all DBDSs of a DBDS group are to be copied. If GROUP is specified, the GENJCL.OIC command executes repeatedly, once for each DBDS of the DBDS group.

PSB(*name***)**
Required keyword you use to specify the name of the PSB that is required for a run of the Online Database Image Copy utility.

If you specify GROUP, the same PSB name is used for all members of the group.

CHKINT(*value***)**
Optional keyword you use to specify the checkpoint interval for the Online Database Image Copy utility. *value* must be a decimal number from 1 to 9999.

If this keyword is omitted, the Online Database Image Copy utility uses its own default value for the checkpoint interval.

COPIES(1 | 2)
Optional keyword you use to request that the Online Database Image Copy utility in order to produce one or two image copy data sets for the specified DBDS.

If the specified DBDS is identified in the RECON data set with the NOREUSE attribute, you must specify the COPIES keyword in order to produce two image copy data sets; otherwise, one image copy data set is produced.

If the specified DBDS is identified in the RECON data set with the REUSE attribute, you cannot specify a COPIES keyword; the number of image copy data sets that are produced for this DBDS is determined by keywords in the INIT.IC command.

DDN(*ddname*)
Optional keyword you use to identify the DBDS that is to be copied.

The DDN keyword can be specified only if the DBD keyword is specified. If DDN is not specified, the GENJCL.OIC command executes repeatedly, once for each DBDS of the specified database. If you specify a HALDB master name, the GENJCL.OIC command is performed for all data DBDSs for each partition in the HALDB master. If you specify a HALDB partition name, the GENJCL.OIC command is performed for all data DBDSs of the identified HALDB partition.

For HALDB databases, you must specify a partition database name with the DBD keyword in order to use the DDN keyword. The DDN keyword value is the partition DDN. The GENJCL.OIC command is performed for the identified DBDS of the partition. The GENJCL.OIC command fails if DDN does not identify a data DBDS in the partition.

DEFAULTS(*member*)
Optional keyword you use to specify up to 10 names of skeletal JCL default members to be used when generating JCL. Default members are searched in order to resolve keywords in the order in which the members are specified on this keyword.

If a keyword is assigned a value in both the DEFAULTS and USERKEYS keywords, the value specified in USERKEYS is used.

JCLOUT(JCLOUT | *ddname*)
Optional keyword you use to specify the output data set for the generated JCL. The data set is specified by *ddname*. A JCL DD statement with this ddname must be included in the job step containing the GENJCL command. The specified data set can be a member of a partitioned data set, but only if it is not the same data set used for the default, JCLOUT.

JCLPDS(JCLPDS | *ddname*)
Optional keyword you use to specify the skeletal JCL data set that is to be used for input when generating JCL. The data set is specified by *ddname*. A JCL DD statement with this ddname must be included in the job step that contains the GENJCL command.

JOB | JOB(*member*) | NOJOB
Mutually exclusive, optional keywords that you use to specify whether to produce the job statement in the generated JCL.

JOB
Specifies that the job statement is to be produced. When JOB is specified

without a member name, the IBM-supplied execution member JOBJCL produces the job statement. When JOB(*member*) is specified, the specified execution member produces the job statement.

NOJOB
Specifies that the job statement is not to be produced in the generated JCL.

NOLIST | LIST
Mutually exclusive, optional keywords that you use to specify whether to write the generated JCL to the SYSPRINT data set.

NOLIST
Suppresses the printing of the generated JCL.

LIST
Prints the generated JCL.

MEMBER(*member*)
Optional keyword you use to specify the name of the skeletal JCL execution member that is to be used. If this keyword is not specified, the default specified on the INIT.DBDS command is used.

MULTIJOB | ONEJOB
Mutually exclusive, optional keywords that you use to control how many JOB statements are to be generated when a DBDS group is specified either explicitly or implicitly.

MULTIJOB
Processes the skeletal JCL JOB member for each group member (multiple JOB statements are to be produced).

ONEJOB
Processes the skeletal JCL JOB member only for the first group member.

These keywords are invalid if NOJOB is specified or if a DBDS group is not specified.

NODEFLT
Optional keyword you use to specify that the implicit skeletal JCL default member, if any, for the DBDS is not to be used.

TIMEFMT(*sublist*)
Optional keyword you use to define the form in which time stamps appear in user-defined output. This specification overrides both the GENJCL default values and any values set on %SET statements in the input skeletal JCL.

The default for the GENJCL output time format is compressed, with a two-digit year, and the offset in numeric form: 960021315001 +0700. If you want the output time stamps to appear without offsets, for example, you can override the default with TIMEFMT(,N).

The override is good only for the duration of a single GENJCL command.

UNIT(3400 | *unittype*)
Optional keyword you use to specify the unit type of the primary output data set. This keyword is valid only if:

- The VOLLIST keyword was specified.
- The DBDS for which the JCL is being generated was defined with the NOREUSE option.

UNIT2(3400 | *unittype*)
Optional keyword you use to specify the unit type of the secondary output data set. This keyword is valid only if:

- The VOLLIST2 keyword was specified.
- The DBDS for which the JCL is being generated was defined with the NOREUSE option.

USERKEYS(*%key1*,*'value'* | *%key2*)

Optional keyword you use to set the value of keywords you have defined. Up to 64 keywords can be specified.

%key1
User-defined keyword that is being assigned a value. The maximum length of the keyword is 8 characters, including the percent sign. The first character after the percent sign must be alphabetic (A-Z). The remaining characters must be alphanumeric (A-Z, 0-9).

'value'
Value assigned to the user-defined keyword when it is encountered. *value* can be any character string enclosed in single quotation marks. The maximum length of *value* is 132 characters (excluding the quotation marks). If *value* contains a quotation mark, use two single quotation marks. *value* can be a null string (''). If *value* is a time stamp, it can be 0.

%key2
Any simple keyword that was previously assigned a value, including DBRC-defined and user-defined keywords.

Any keyword can be assigned a value with the USERKEYS keyword. However, if you assign a value to DBRC-defined keywords, the value is ignored.

If a keyword is assigned a value in both the USERKEYS and DEFAULTS keyword, the value specified in USERKEYS is used.

VOLLIST(*volser*)
Optional keyword you use to specify the volumes on which the image copy data set is to reside. This keyword is valid only if the DBDS for which the JCL is being generated is defined with the NOREUSE option, and if the DBDS is being used with DBDs, not with groups.

VOLLIST2(*volser*)
Optional keyword you use to specify the volumes on which the duplicate image copy data set is to reside. This keyword is valid only if the DBDS for which the JCL is being generated is defined with the NOREUSE option, and if the DBDS is being used with DBDs, not with groups.

Examples

Here are some examples of using the GENJCL.OIC command.

Example using JCLPDS member OICJCL

In this example, a GENJCL.OIC command generates the JCL and control statements required to run the Online Database Image Copy utility for the DBDS identified in the DBD and DDN keywords. The database is defined as REUSE. If the INIT.DBDS command for the DBDS for which the JCL is being generated is specified with OICJCL(*member*), that member is used and is found in the data set identified in the JCLPDS DD statement. If not, default skeletal member OICJCL

from the JCLPDS data set is used. Output from the generated JCL goes to the data set defined in the JCLOUT DD statement. Skeletal member JOBJCL produces a job statement.

```
//GENJOIC  JOB
//JCLPDS   DD   . . .
//JCLOUT   DD   . . .

⋮
//SYSIN    DD   *
     GENJCL.OIC     DBD(DBDKSDS1) DDN(DDNKSDS1)  -
                    PSB(MYJOB)
/*
```

Example using JCLPDS member OICJCL2

In this example, the GENJCL.OIC command generates JCL and control statements to run the Online Image Copy utility for all DBDSs of GROUP1. The skeletal member used is OICJCL2 from the data set identified in the OICPDS DD statement. One job statement for each group member is generated from the JOBCARD member found in the data set identified in the OICPDS DD statement. %DEFDBDS is a user-defined value in OICJCL2 and is resolved with 'DATABASE DEFINED HERE'. Members DEF1, DEF2, and DEF3 are used to resolve user-defined keywords in OICJCL2. The default member for the database initialized in INIT.DBDS DEFLTJCL(MEMBER) is not used to resolve keywords. The values specified in the USERKEYS keyword for a keyword override the values found in the DEFAULTS member.

```
//GENJOIC1 JOB
//OICOUT   DD  . . .
//OICPDS   DD  . . .
//SYSIN    DD  *
  GENJCL.OIC GROUP(GROUP1) JOB(JOBCARD) MEMBER(OICJCL2) -
         NODEFLT  JCLPDS(OICPDS) JCLOUT(OICOUT) PSB(PCBOIC6)      -
         USERKEYS((%DEFDBDS,'DATABASE DEFINED HERE'))     -
    DEFAULTS(DEF1,DEF2,DEF3)
```

Related reference:

"Generating JCL and user-defined output" on page 223

"DBRC time stamps" on page 52

"Skeletal JCL syntax" on page 225

GENJCL.RECOV command

Use the GENJCL.RECOV command to generate the JCL and utility control statements that are required to run the Database Recovery utility (DFSURDB0).

You can generate the JCL and utility control statements for a full recovery or a time-stamp recovery of a specified DBDS or area. All log data must be archived; otherwise the GENJCL.RECOV command fails.

Restrictions:

- A GENJCL.RECOV command cannot be issued for a user-recoverable database.
- The GENJCL.RECOV command does not support ILDS and index data sets. To generate JCL for the HALDB Index/ILDS Rebuild utility (DFSPREC0), use the GENJCL.USER command.

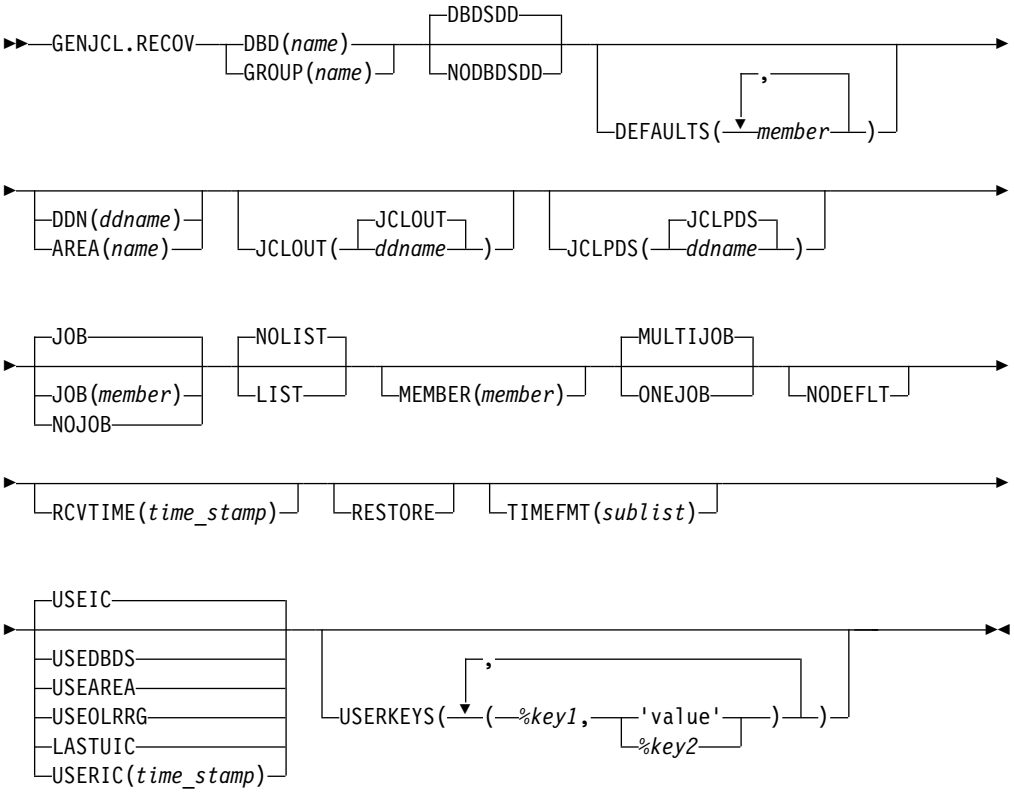
Important:

- For HALDB partitions, the GENJCL.RECOV command processes ILDS and index data sets differently than data DBDSs. The GENJCL.RECOV command skips these data sets in groups, regardless of whether the groups are explicit or implicit. If you specify one of these data sets explicitly, the GENJCL.RECOV command fails.

Subsections:

- "Syntax"
- "Keywords"
- "Examples" on page 217

Syntax



Keywords

DBD(*name***) | GROUP(***name***)**
Mutually exclusive, required keywords that you use to specify the DBDSs that are to be recovered.

DBD(*name***)**
Specifies the database name of the DBDSs to be recovered.

Restriction for HALDB databases: For the GENJCL.RECOV command, *name* specifies the name of either a HALDB master database or a HALDB partition.

GROUP(*name***)**
Specifies that all DBDSs of a DBDS or CA group are to be recovered. If GROUP is specified, the GENJCL.RECOV command executes repeatedly for each DBDS of the DBDS or CA group. If you attempt an implicit or

explicit group execution with recoverable and nonrecoverable DBDSs (and RESTORE is not specified), JCL is not generated for the nonrecoverable DBDSs.

DBDSDD|NODBDSDD
Mutually exclusive, optional keywords that specify whether a DD statement for the DBDS is included in the generated JCL.

NODBDSDD
Specifies that a DD statement for the DBDS is not included in the generated JCL. The Database Recovery utility receives the data set name for the DBDS from DBRC.

The NODBDSDD keyword is applicable only to recoveries from Fast Replication image copies.

For example, the following command:

```
GENJCL.RECOV DBD(DB1) DDN(DBDS1) USEIC NODBDSDD
```

generates the following control statement.

```
S  DB1      DBDS1                                                D
```

In this case, the DD statement DBDS1, which identifies the database data set, is not required in the job stream, because the DBDS name is extracted from DBRC.

DBDSDD
Specifies that a DD statement for the DBDS is included in the generated JCL.

For example, the following command:

```
GENJCL.RECOV DBD(DB1) DDN(DBDS1) USEIC
```

generates the following control statement.

```
S DB1 DBDS1 D1OU1
```

In this case, you must include the DD statement DBDS1 in the job stream and the DD statement must reference the data set name of the target DBDS.

DEFAULTS(*member*)
Optional keyword you use to specify up to 10 names of skeletal JCL default members to be used when generating JCL. Default members are searched to resolve keywords in the order in which the members are specified on this keyword.

If a keyword is assigned a value in both the DEFAULTS and the USERKEYS keywords, the value specified in USERKEYS is used.

DDN(*ddname*) | AREA(*name*)
Mutually exclusive, optional keywords that you use to identify the DBDS ddname or DEDB area to be recovered.

The DDN or AREA keyword is specified only if the DBD keyword is specified.

For HALDB databases, you must specify a partition database name with the DBD keyword in order to use the DDN keyword. The DDN keyword value is the partition DDN. The GENJCL.RECOV command is performed for the identified DBDS of the partition. The GENJCL.RECOV command fails if DDN does not identify a data DBDS in the partition.

If DDN or AREA is not specified, the GENJCL.RECOV command executes repeatedly for each DBDS or area of the specified database. If you specify a HALDB master name, the GENJCL.RECOV command is performed for all data DBDSs for each HALDB partition in the HALDB master. If you specify a HALDB partition name, the GENJCL.RECOV command is performed for all data DBDSs of the identified partition.

JCLOUT (JCLOUT | *ddname*)
Optional keyword you use to specify the output data set for the generated JCL. The data set is specified by *ddname*. A JCL DD statement with this ddname must be included in the job step containing the GENJCL command. The specified data set can be a member of a partitioned data set, but only if it is not the same data set used for the default (JCLOUT).

JCLPDS (JCLPDS | *ddname*)
Optional keyword you use to specify the skeletal JCL data set that is to be used for input when generating JCL. The data set is specified by *ddname*. A JCL DD statement with this ddname must be included in the job step containing the GENJCL command.

JOB | JOB(*member*) | NOJOB
Mutually exclusive, optional keywords that you use to specify whether to produce the job statement in the generated JCL.

JOB
Specifies that the job statement is to be produced. When JOB is specified without a member name, the IBM-supplied execution member JOBJCL produces the job statement. When JOB(*member*) is specified, the specified execution member produces the job statement.

NOJOB
Specifies that the job statement is not to be produced in the generated JCL.

NOLIST | LIST
Mutually exclusive, optional keywords that you use to specify whether to write the generated JCL to the SYSPRINT data set.

NOLIST
Suppresses printing of the generated JCL.

LIST
Prints the generated JCL.

MEMBER(*member*)
Optional keyword you use to specify the name of the skeletal JCL execution member to be used. If this keyword is not specified, the default specified on the INIT.DBDS command is used.

MULTIJOB | ONEJOB
Mutually exclusive, optional keywords that you use to control how many JOB statements are generated when a DBDS group is specified either explicitly or implicitly.

MULTIJOB
Processes the skeletal JCL JOB member for each group member (multiple JOB statements are produced).

ONEJOB
Processes the skeletal JCL JOB member only for the first group member.

These keywords are invalid if NOJOB is specified or a DBDS group is not specified.

NODEFLT
Optional keyword you use to specify that the implicit skeletal JCL default member, if any, for the DBDS is not to be used.

RCVTIME(*time_stamp***)**
Optional keyword you use to specify a time-stamp recovery, which is a partial recovery of a DBDS or area to a point in time earlier than its most recent state. If you omit this keyword, you are requesting a full recovery to the most recent state.

A valid time stamp for a partial recovery is any point at which there are no allocations of the DBDS or area and there is not a merge of logs needed that cannot be resolved by running the Change Accumulation utility.

Attention: An allocation that has no de-allocation time recorded persists until the stop time of the current log.

RESTORE
Optional keyword that generates JCL for a DBDS that is designated as nonrecoverable. If the last image copy was taken before the DBDS was designated as nonrecoverable, normal recovery JCL is generated to recover the DBDS up to the point of the recovery-status change. If the last image copy was taken after the DBDS was designated as nonrecoverable, the generated JCL uses only the image copy for recovery.

If you attempt an implicit or explicit group execution with recoverable and nonrecoverable DBDSs (and RESTORE is specified), JCL is generated only for the nonrecoverable DBDSs.

Do not specify RESTORE for a recoverable DBDS.

TIMEFMT(*sublist***)**
Optional keyword that defines the form in which time stamps appear in user-defined output. This specification overrides both the GENJCL default values and any values set on %SET statements in the input skeletal JCL.

The default for the GENJCL output time format is compressed, with a two-digit year, and the offset in numeric form: 960021315001 +0700. If you want the output time stamps to appear without offsets, for example, you can override the default with TIMEFMT(,N).

The override is in effect only for the duration of a single GENJCL command.

USEIC | USEDBDS | USEAREA | USEOLRRG | LASTUIC | USERIC
Mutually exclusive, optional keywords that you use to specify the starting point of the requested recovery action.

USEIC
Starts the recovery with an image copy data set. You can then apply subsequent changes that occurred in the DBDS.

USEIC is the default.

USEDBDS
Recovery is performed using only the changes that have occurred to the DBDS in its current state. An image copy data set is not used as input to this recovery. You can specify the USEDBDS keyword only if you also specify the DBDS keyword, and only after performing a time-stamp recovery in which an image copy data set is used as input.

USEAREA
Recovery is performed using only the changes that have occurred to the

DEDB area in its current state. An image copy data set is not used as input to this recovery. You can specify USEAREA only if you also specify the AREA keyword, and only after performing a time-stamp recovery in which an image copy data set is used as input.

USEOLRRG
Starts the recovery with a HALDB Online Reorganization updates.

LASTUIC
Specifies that the recovery of a restored database data set that uses the latest concurrent user image copy created from user-generated JCL.

When you specify the LASTUIC keyword, the concurrent image copy data set is not included in the recovery JCL, but all log data sets that are required to perform the time stamp recovery are included. To recover from a concurrent user image copy, you must specify either the USERIC or LASTUIC keyword. All logs that were used between the time indicated by the USERIC keyword and the time indicated by the RCVTIME keyword are obtained by the GENJCL.RECOV command, and these logs are used in the recovery.

Restriction: You must restore the database data sets from concurrent user image copies before you recover database data sets by using the GENJCL.RECOV LASTUIC command.

USERIC(*time_stamp*)

Specifies the start time of a specific concurrent user image copy created from user-generated JCL. You must specify this keyword to indicate the run time of a concurrent user image copy generated from user-generated JCL that is used to restore a database data set for recovery. The time stamp must be in standard form.

The Database Recovery utility uses the time stamp only to determine the time from which to start applying the log data sets. The Database Recovery utility does not use the time stamp to determine which image copies to use. The Database Recovery utility will look for the most recent image copy record related to a user image copy. You must then include the image copies yourself. You must specify either the USERIC or LASTUIC keyword to recover from a concurrent user image copy.

Restriction: You must restore the database data sets from concurrent user image copies before you recover database data sets by using the GENJCL.RECOV USERIC command.

You can use these keywords to recover a DBDS or area to a specified time stamp using an image copy data set and then apply the changes that have occurred since the image copy by specifying an additional recovery using the USEDBDS or USEAREA keyword.

Restriction: If this required time-stamp recovery restored the DBDS or DEDB area to a time that falls within an existing time-stamp recovery's range (the time between the RECOV TO and RUN times), the USEDBDS or USEAREA keyword is invalid.

USERKEYS(*%key1*,*'value'* | *%key2*)

Optional keyword you use to set the value of keywords you have defined. Up to 64 keywords can be specified.

%key1
User-defined keyword that is being assigned a value. The maximum length of the keyword is 8 characters, including the percent sign. The first character after the percent sign must be alphabetic (A-Z). The remaining characters must be alphanumeric (A-Z, 0-9).

'value'
Value assigned to the user-defined keyword when it is encountered. *value* can be any character string enclosed in single quotation marks. The maximum length of *value* is 132 characters (excluding the quotation marks). If *value* contains a quotation mark, use two single quotation marks. *value* can be a null string (''). If *value* is a time stamp, it can be 0.

%key2
Any simple keyword that was previously assigned a value, including DBRC-defined and user-defined keywords.

Any keyword can be assigned a value with the USERKEYS keyword. However, if you assign a value to DBRC-defined keywords, the value is ignored.

If a keyword is assigned a value in both the USERKEYS and DEFAULTS keyword, the value specified in USERKEYS is used.

Examples

The following examples show how to use the GENJCL.RECOV command to generate the JCL that is needed to run the Database Recovery utility.

Example for the DBDS identified in the DBD and DDN keywords

In this example, a GENJCL.RECOV command generates the JCL and control statements required to run the Database Recovery utility for the DBDS identified in the DBD and DDN keywords. The USEIC keyword indicates that the time-stamp recovery starts with an image copy data set and ends with the log data set that has the stop time stamp specified in the RCVTIME keyword.

If the INIT.DBDS command for the DBDS for which the JCL is being generated is specified with RECOVJCL(*member*), that member is used and is found in the data set identified in the JCLPDS DD statement. If not, default skeletal member RECOVJCL from the JCLPDS data set is used. Output from the generated JCL goes to the data set identified in the JCLOUT DD statement. Skeletal member JOBJCL produces a job statement.

```
//GENJRCOV JOB
//JCLPDS   DD   . . .
//JCLOUT   DD   . . .
⋮
//SYSIN    DD   *
     GENJCL.RECOV  DBD(DBESDSA) DDN(DDESDSA) USEIC   -
      RCVTIME(071001212130)
/*
```

Example for all DBDSs in a group

In this example, the GENJCL.RECOV command generates JCL and control statements to run the Database Recovery utility for all DBDSs of GROUP1. The skeletal member that is used is RECJCL2 from the data set identified in the PDS DD statement.

Skeletal member JOBJCL produces a job statement for each member of the group. %DEFDBD1 and %DEFDBD2 are user-defined values in member RECJCL2 which resolve to 'DEFINE DB1' and 'DEFINE DB2'. Default members DEF1, DEF2, and DEF3 are used to resolve user-defined keywords in RECJCL2. The default member for the DBDS, if initialized in the INIT.DBDS DEFLTJCL(MEMBER) command, is also used to resolve keywords. The values in the explicitly defined DEFAULTS members override values in the predefined DEFLTJCL member. The values specified in the USERKEYS keyword for a keyword override the values found in the DEFAULTS members.

```
//GENJRCV1 JOB
//OUT      DD  . . .
//PDS      DD  . . .
//SYSIN    DD  *
  GENJCL.RECOV GROUP(GROUP1) MEMBER(RECJCL2) -
        JCLPDS(PDS) JCLOUT(OUT)       -
        USERKEYS((%DEFDBD1,'DEFINE DB1'),(%DEFDBD2,'DEFINE DB2'))   -
    DEFAULTS(DEF1,DEF2,DEF3)
```

Related reference:

"Generating JCL and user-defined output" on page 223

"DBRC time stamps" on page 52

"Skeletal JCL syntax" on page 225

"IBM-supplied skeletal JCL execution members" on page 258

GENJCL.USER command

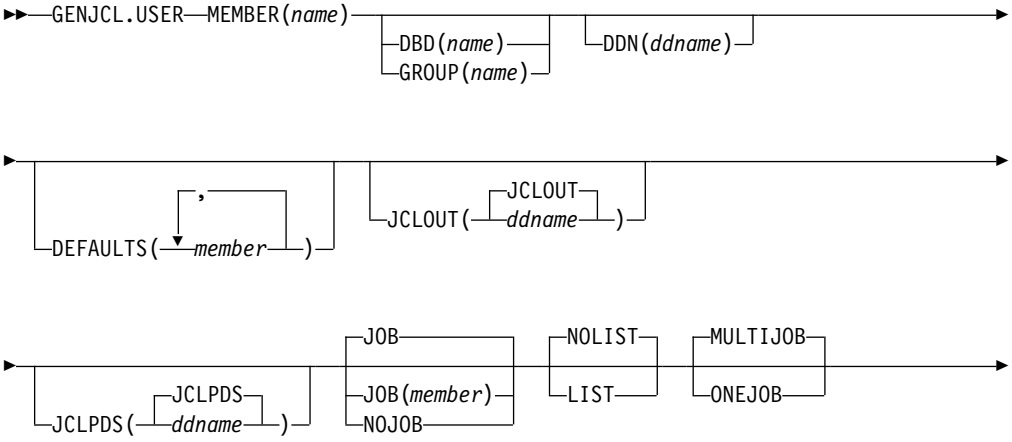
Use the GENJCL.USER command to generate JCL or any kind of user output.

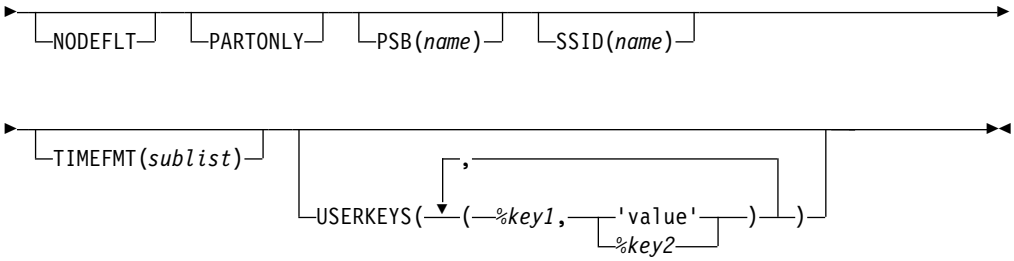
You must provide the skeletal JCL execution member that is needed for the GENJCL.USER command.

Subsections:

- "Syntax"
- "Keywords" on page 219
- "Example of generating JCL" on page 222

Syntax





Keywords

MEMBER(*name***)**
Required keyword you use to specify the name of the skeletal JCL execution member that is used to generate output. You must have already supplied the execution member.

The name can be any valid member name for a partitioned data set. If the specified member does not exist in the skeletal JCL data set, the command fails.

DBD(*name***) | GROUP(***name***)**
Mutually exclusive, optional keywords that you use to set the value of the `%dbname` keyword.

DBD(*name***)**

If you specify DBD without the DDN keyword, the GENJCL.USER command executes repeatedly for each DBDS or the specified database.

Restriction for HALDB databases: For the GENJCL.USER command, *name* specifies the name of either a HALDB master database name or a HALDB partition name. You can use this keyword to set the value of the `%dbname` keyword to be either a HALDB master name or a HALDB partition name. If you use a HALDB master name, the GENJCL.USER command is performed for all data DBDSs for each HALDB partition in the HALDB master. If you use a HALDB partition name, the GENJCL.USER command is performed for all DBDSs of the identified partition.

GROUP(*name***)**
If you specify GROUP, the GENJCL.USER command executes repeatedly, once for each DBDS of the specified DBDS group. For each repeated execution, the DBD and DDN keywords are set to the corresponding group member.

If you specify neither DBD nor GROUP, the value of the `%dbname` keyword is null unless a value is assigned in the USERKEYS keyword or a skeletal JCL default member.

DDN(*ddname***)**
Optional keyword you use to set the value of the `%ddname` keyword. If you do not specify DDN, the value of the `%ddname` keyword is null unless a value is assigned in the USERKEYS keyword or a skeletal JCL default member.

For HALDB databases, you must specify a partition database name with the DBD keyword in order to use the DDN keyword. In this case, the DDN is the partition DDN. The GENJCL.USER command is performed for the identified DBDS of the partition. The GENJCL.USER command fails if DDN does not identify a DBDS in the partition.

You cannot specify DDN if you also specify GROUP.

DEFAULTS(*member***)**
Optional keyword you use to specify up to 10 names of skeletal JCL default members to be used when generating JCL or other user-defined output. Default members are searched to resolve keywords in the order in which the members are specified on this keyword.

If a keyword is assigned a value in both the DEFAULTS and USERKEYS keywords, the value specified in USERKEYS is used.

JCLOUT(JCLOUT | *ddname*)
Optional keyword you use to specify the output data set for the generated JCL or other user-defined output. The data set is specified by *ddname*. A JCL DD statement with this ddname must be included in the job step containing the GENJCL.USER command. The specified data set can be a member of a partitioned data set, but only if it is not the same data set that is used for the default (JCLOUT).

JCLPDS(JCLPDS | *ddname*)
Optional keyword you use to specify the skeletal JCL data set that is to be used for input when generating the JCL or other user-defined output. The data set is specified by *ddname*. A JCL DD statement with this ddname must be included in the job step containing the GENJCL.USER command.

JOB | JOB(*member*) | NOJOB
Mutually exclusive, optional keywords that you use to specify whether to produce the job statement in the generated JCL.

JOB
Specifies that the job statement is to be produced. When JOB is specified without a member name, the IBM-supplied execution member JOBJCL produces the job statement. When JOB(*member*) is specified, the specified execution member produces the job statement.

NOJOB
Specifies that the job statement is not to be produced in the generated JCL.

NOLIST | LIST
Mutually exclusive, optional keywords that you use to specify whether to write the generated JCL to the SYSPRINT data set.

NOLIST
Suppresses printing of the generated JCL.

LIST
Prints the generated JCL.

MULTIJOB | ONEJOB
Mutually exclusive, optional keywords that you use to control how many JOB statements are generated when a DBDS group is specified either explicitly or implicitly.

MULTIJOB
Processes the skeletal JCL JOB member for each group member (multiple JOB statements are produced).

ONEJOB
Only processes the skeletal JCL JOB member for the first group member.

These keywords are invalid if NOJOB is specified or if a DBDS group is not specified.

NODEFLT
Optional keyword you use to specify that the implicit skeletal JCL default member, if any, for the DBDS is not to be used. If you do not specify GROUP or DBD, this keyword is ignored.

PARTONLY
Optional keyword you use to specify that only the HALDB partitions are to be processed instead of all data DBDSs for each HALDB partition. PARTONLY is only meaningful for HALDB. If the specified DBD is not a HALDB master name or a HALDB partition name then this keyword is ignored. This keyword sets the value of %mdbname to the HALDB master name and %dbname to the HALDB partition name.

PSB(*name*)
Optional keyword you use to set the value of the %PSB keyword.

name can be any character string. It does not need to be an actual PSB ddname. The maximum length of the name is 8 characters.

If you do not specify PSB, the value of the %PSB keyword is null unless a value is assigned in the USERKEYS keyword or a skeletal JCL default member.

SSID(*name*)
Optional keyword used to set the value of the %SSID keyword.

name can be any character string. It does not need to be an actual IMS subsystem ID. The maximum length of the name is 8 characters.

If the SSID keyword is not specified, the value of the %SSID keyword is null, unless a value is assigned in the USERKEYS keyword or a skeletal JCL default member.

TIMEFMT(*sublist*)
Optional keyword you use to define the form in which time stamps appear in user-defined output. This specification overrides both the GENJCL default values and any values set on %SET statements in the input skeletal JCL.

The default for the GENJCL output time format is compressed, with a two-digit year, and the offset in numeric form: 960021315001 +0700. If you want the output time stamps to appear without offsets, for example, you can override the default with TIMEFMT(,N).

The override is good only for the duration of a single GENJCL command.

USERKEYS(*%key1*,*'value'* | *%key2*)

Optional keyword you use to set the value of keywords you have defined. Up to 64 keywords can be specified.

%key1
User-defined keyword being assigned a value. The maximum length of the keyword is 8 characters, including the percent sign. The first character after the percent sign must be alphabetic (A-Z). The remaining characters must be alphanumeric (A-Z, 0-9).

'value'
Value assigned to the user-defined keyword when it is encountered. *value* can be any character string enclosed in single quotation marks. The maximum length of *value* is 132 characters (excluding the quotation marks). If *value* contains a quotation mark, use two single quotation marks. *value* can be a null string (''). If *value* is a time stamp, it can be 0.

%key2
Any simple keyword that was previously assigned a value, including DBRC-defined and user-defined keywords.

Any keyword can be assigned a value with the USERKEYS keyword. However, if you assign a value to DBRC-defined keywords, the value is ignored.

If a keyword is assigned a value in both the USERKEYS and DEFAULTS keyword, the value specified in USERKEYS is used.

Example of generating JCL

In this example, the GENJCL.USER command generates JCL from member USER4 in the data set that is identified in DD statement MYJCLPDS. Output from the generated JCL goes to the data set identified in DD statement JCLOUT. Substitutions for %SSID, %DBNAME and %DDNAME should be made. Skeletal member JOBJCL produces a job statement.

```
//GENUSER  JOB
//JCLOUT   DD
//MYJCLPDS DD
//SYSIN    DD *
  GENJCL.USER MEMBER(USER4) JCLPDS(MYJCLPDS)  DBD(DHONTZ04) -
        SSID(IMSA) DDN(HIDAM)
```

The following example shows the member USER4 that is to be executed:

```
/        ADD LIST=ALL,NAME=USER4,LEVEL=01,SOURCE=0
/        NUMBER  NEW1=00000100,INCR=100
//*******************************************************
//*                MEMBER NAME = USER4                  *
//* (SSID) SHOULD BE SUBSTITUTED IN LIST.SUBSYS COMMAND *
//* (DBNAME) SHOULD BE SUBSTITUTED IN LIST.DBDS COMMAND *
//* (DDNAME) SHOULD BE SUBSTITUTED IN LIST.DBDS COMMAND *
//*******************************************************
//USER4    EXEC PGM=DSPURX00
//SYSPRINT DD SYSOUT=A
//SYSIN    DD *
  LIST.SUBSYS SSID(%SSID)    /* (SSID) SHOULD BE SUBSTITUTED */
  LIST.DBDS DBD(%DBNAME) DDN(%DDNAME)      -
       /* (DBNAME) and (DDNAME) SHOULD BE SUBSTITUTED */
/*
```

Related reference:

"Generating JCL and user-defined output" on page 223

"DBRC time stamps" on page 52

"Skeletal JCL syntax" on page 225

Skeletal JCL for DBRC

DBRC uses partitioned data set (PDS) members as input models (or templates) to generate input for some of the recovery utilities. These PDS members are distributed with IMS and are called *skeletal JCL*.

DBRC uses the skeletal JCL, information from the RECON data set, and instructions from a GENJCL command to generate the JCL and control statements that are needed to correctly run some of the recovery utilities. Modify the skeletal JCL to reflect your installation's system configuration.

Generating JCL and user-defined output

Eight GENJCL commands are supported by the Database Recovery Control utility. Seven of these commands generate the JCL and control statements necessary to run various IMS recovery-related utilities. You can use the remaining command, GENJCL.USER, to generate user-defined output, including JCL.

The following table shows the GENJCL commands and their output.

Table 5. GENJCL commands and output

Command (PDS member)	Output generated
`GENJCL.ARCHIVE` (ARCHJCL member)	Log Archive utility JCL and control statements
`GENJCL.CA` (CAJCL member)	Database Change Accumulation utility JCL and control statements
`GENJCL.CLOSE` (LOGCLJCL member)	Log Recovery utility JCL and control statements
`GENJCL.IC` (ICJCL member)	Database Image Copy or Database Image Copy 2 utility JCL and control statements
`GENJCL.OIC` (OICJCL member)	Online Database Image Copy utility JCL and control statements
`GENJCL.RECEIVE` (ICRCVJCL member)	Database Recovery utility JCL and control statements
`GENJCL.RECOV` (RECOVJCL member)	Database Recovery utility JCL and control statements
`GENJCL.USER` (DSPUPJCL member)	User-defined output, including JCL and control statements. Sample JCL for the HALDB INDEX/ILDS Rebuild utility (DSPUPJCL) is shipped with IMS and can be used with the GENJCL.USER command, but other types of JCL can be used as well. No default is defined.

When you issue a GENJCL command, it uses a skeletal JCL execution member. The execution member is a model of the JCL or user output that you are producing. The execution member contains symbolic keywords. DBRC substitutes current information for the symbolic keywords. The substituted information comes from the RECON data set and from skeletal JCL default members, and from your USERKEY values. Typical of the information DBRC substitutes for symbolic keywords are data set names and volume information. DBRC performs the keyword substitution and then generates the JCL or user output you requested by issuing the GENJCL command.

IBM supplies a JOB statement execution member that is used by all GENJCL commands. If the IBM-supplied skeletal JCL execution members meet your general requirements, you can modify them slightly to provide installation-specific information.

If the IBM-supplied skeletal JCL does not meet your general requirements or if you plan to use the GENJCL.USER command, you must write your own skeletal JCL members or define new symbolic keywords to include in the IBM-supplied skeletal JCL. IBM provides no skeletal JCL execution member for the GENJCL.USER command.

Related reference:

"Symbolic keywords recognized by DBRC" on page 249

"Using IBM-supplied skeletal JCL" on page 224

Using IBM-supplied skeletal JCL

If you are generating JCL for the IMS recovery-related utilities using the IBM-supplied skeletal JCL execution members, the process is simple. It involves modifying the IBM-supplied skeletal JCL execution members.

Before you use them:

- Add two DD statements (JCLPDS and JCLOUT) to the DBRC dependent address space procedure used for online IMS. JCLPDS identifies the partitioned data set containing the skeletal JCL execution members. JCLOUT identifies the data set to which the generated job is to be written. Output is in card image format. The output data set can be a punch file, a DASD data set that you plan to examine before submitting the job for execution, or the z/OS internal reader.

  These two DD names can be specified on the GENJCL command. When GENJCL is used, the two specified data sets are in effect for the GENJCL command only, and not for the life of the job. The JCLOUT data set is opened at the start of the command execution and closed at the end of the command execution. Consequently, if multiple GENJCL commands are concatenated in the job stream, the JCLOUT data set (if other than the z/OS internal reader) only contains the results from the last command that was processed.
- Add any STEPLIB DD names, and job accounting information that your installation requires to the skeletal JCL execution member. Except for the skeletal JCL member for the JOB statement, do not add any JOBLIB, and JES control statements to your skeletal JCL; doing so causes errors if multiple steps are generated.
- Change the default value for the REGION parameter on the skeletal JCL EXEC statement if the existing one is not correct for your installation.
- If you plan to generate JCL to run the Log Recovery utility (member `LOGCLJCL`), replace the DFSWADS0 DD statement.

Recommendation: Exercise care when modifying the skeletal JCL, because DBRC does not verify any of the JCL that is generated.

Skeletal JCL data set members

The skeletal JCL data set contains the skeletal JCL members used by the GENJCL command processor to generate output.

As shown in the following figure, the two types of skeletal JCL members are execution members and default members.

Execution members are models of the output you are generating. Execution members can be IBM supplied or supplied by you. Execution members contain symbolic keywords, which represent information DBRC provides.

Default members specify default values for symbolic keywords in the execution members. The use of default members is optional. You provide the default members. To use a default member, you specify the member on the GENJCL command. Or, in the case of DBDS and CA groups, you can implicitly specify the default member.

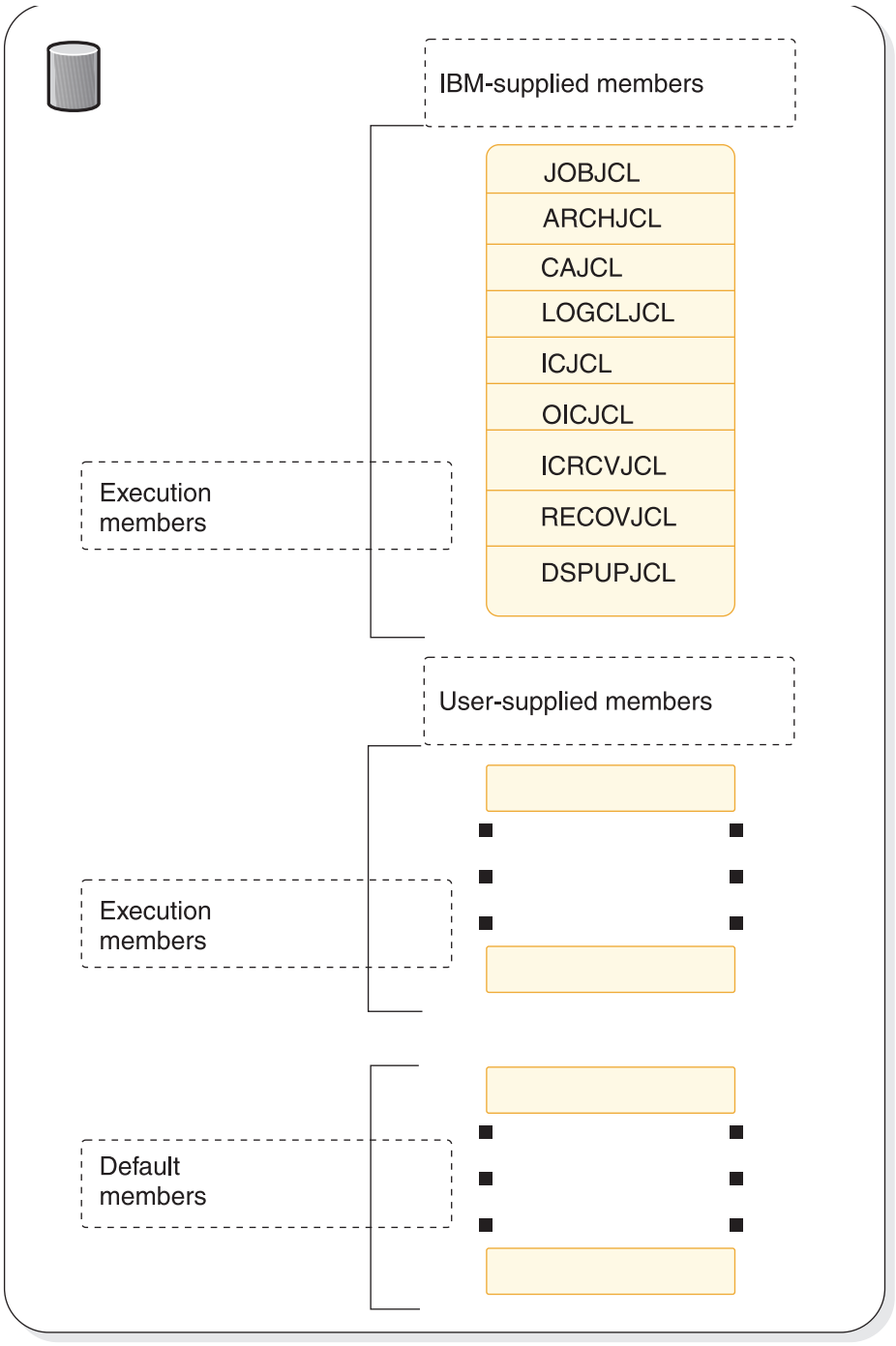


Figure 1. Skeletal JCL data set contents

Skeletal JCL syntax

Skeletal JCL execution members are models of the output to be produced. The execution members contain symbolic keywords and control keywords. Symbolic keywords are symbolic parameters that DBRC substitutes with a value in the output stream (for example, data set names), based upon what is specified on a `GENJCL` command and information obtained from the RECON data set.

Control keywords regulate what output is generated by DBRC, based upon what is specified on a GENJCL command (for example, what RECON records are to be used for keyword substitution).

The keywords in skeletal JCL members must be in uppercase.

Subsections:

- "Symbolic keywords"
- "%SELECT and %ENDSEL control keywords" on page 227
- "%DELETE and %ENDDEL control keywords" on page 230
- "%SET MEMBER control keyword" on page 231
- "%SET TIMEFMT control keyword" on page 232
- "Conventions for control keywords" on page 233
- "Writing skeletal JCL default members" on page 248
- "Specifying default members" on page 248
- "Using user-supplied or modified skeletal JCL" on page 249

Symbolic keywords

When JCL is generated, symbolic keywords in the skeletal JCL execution members are replaced with the current keyword value. For example, the IBM-supplied skeletal JCL execution members use %TIME as a symbolic keyword. When DBRC encounters %TIME, it replaces it with the time of day. Keyword substitution occurs each time DBRC encounters a symbolic keyword. Multiple symbolic keywords can exist in a skeletal JCL execution member.

Symbolic keywords must be assigned a value before you use them. Keyword values are assigned (or set) in several different ways as specified below:

- The GENJCL command specifies values for some of the symbolic keywords in skeletal JCL execution or default members. User-defined keywords are assigned a value in the USERKEYS parameter in the command. Other keyword values are set by various parameters on the command. For example, the SSID parameter sets the value for the %SSID keyword (the subsystem ID).
- Skeletal JCL default members set default values for keywords in skeletal JCL execution members.
- The RECON data set also provides keyword values. For example, when the GENJCL.ARCHIVE command is issued, the ddnames and data set names for the OLDS are obtained from the PRIOLDS and SECOLDS records.
- Some keyword values are implicitly known, for example the time of day.

If during the JCL generation process, a keyword is encountered that has not been assigned a value, no substitution takes place. Instead, DBRC issues a warning message.

When writing your own skeletal JCL execution members, you can define your own symbolic keywords as well as use the symbolic keywords already recognized by DBRC. You can also define your own symbolic keywords and add them to the IBM-supplied skeletal JCL execution members.

Here are some conventions, restrictions, and other detail you should know when writing symbolic keywords:

- Keywords must begin with a percent (%) sign.

- The minimum keyword length is two characters, including the percent sign. The maximum length is eight characters, including the percent sign.
- Keywords must be written using uppercase letters only (`A` rather than `a`).
- The first character after the percent sign must be alphabetic (A-Z); the remaining characters must be alphanumeric (A-Z, 0-9). Keywords are delimited by a non-alphanumeric character or when the maximum length is reached.
- DBRC does not use any keywords beginning with `%W`, `%X`, `%Y`, or `%Z`. You can, therefore, use these characters for your own keywords without conflicting with predefined keywords.
- User-defined symbolic keywords must be assigned a value with the USERKEYS parameter on the GENJCL command or with a skeletal JCL default member.
- Keyword substitution is performed on columns 1-71 of the skeletal JCL records. Columns 72-80 are not modified. If the keyword value is shorter than the keyword, the remaining data on the record is shifted to the left and filled with blanks. If the keyword value is longer than the keyword, the remaining data is shifted to the right. If any non-blank characters are shifted beyond column 71, a JCL continuation statement is generated. In some cases (for example, when the output is not a JCL statement), it might not be possible to generate a JCL continuation statement, because a comma or blank must exist in the output record for DBRC to split it. When DBRC cannot find a break in the statement, it splits the statement at column 71.

%SELECT and %ENDSEL control keywords

The `%SELECT` control keyword selects the RECON records that are needed in order to resolve symbolic keywords. The `%ENDSEL` control keyword indicates the end of the records selected by the `%SELECT` keyword. These control keywords always occur in pairs. A `%SELECT` keyword is followed by one or more execution member records, which is followed by the `%ENDSEL` keyword. This sequence of records is called a control group or, more specifically, a select group.

Use the `%SELECT` keyword to select one or more records from the RECON data set. The selected records identify IMS data sets or events tracked by DBRC. Information from the selected records is used to resolve symbolic keywords in the select group. Symbolic keywords can occur in the execution member records or in the `%SELECT` keyword statement.

The format for a select group follows:

```
%SELECT record_type(selection_criteria)
 execution_member_record(s)
%ENDSEL
```

The `record_type` is the type of RECON record to be selected. You can select any of the following *record_type*s:

- OLDS (PRIOLD)
- SLDS (PRISLD)
- RLDS (PRILOG)
- IC (IMAGE)
- CA (CA)
- ALLOC (ALLOC)
- DBDS (DBDS)

The selection_criteria depend on the type of record you select, and can be time ranges and ddnames.

Both the record_type and selection_criteria can be symbolic keywords.

As RECON records are selected, information from them is used to set the values of symbolic keywords. Depending on the type of record being selected, the values of a particular group of symbolic keywords associated with the selected record are set.

Any values assigned to a keyword before the select group is processed are overridden when the select group is processed. The keyword values in effect after the select group is processed are the values set from the last selected record. Keyword values remain unchanged if no records are selected. In this case, the records in the select group are not processed. The next records to be processed are those that appear just after the %ENDSEL statement. A select group can occur within a delete group. When this occurs and the delete group is deleted, the select group is not processed, and no keyword values are set (or changed).

The selection_criteria for a select group can cause one or more RECON records to be selected. One execution member can be output more than once depending upon the type of records that have been selected.

When the output stream is JCL, a select group can generate either concatenated or repeated DD statements. The first execution member record of the select group determines which is to be generated. Repeated DD statements are generated if this record is a JCL DD statement and the ddname is a symbolic keyword. Otherwise, a concatenated DD statement is generated.

Example:

Assume that the first record is:

```
//DDNAME DD DSN= . . .
```

In this case, concatenated DD statements are generated. Alternatively, the first record might be:

```
//%DDNAME DD DSN= . . .
```

In this case, repeated DD statements are generated. When repeated DD statements are generated, you must provide some mechanism to ensure that the repeated ddnames are unique. When selecting OLDSs, DBRC uses the OLDS ddname, which is in the OLDS RECON record. DBRC does not track ddnames for any other type of data set. Therefore, DBRC might not be able to generate unique ddnames for data sets that are not OLDSs.

The two sections that follow explain the record_type and selection_criteria parameters in more detail.

Specifying the record type parameter

The types of records that can be specified on the %SELECT keyword are shown in the following table.

Table 6. Records that can be selected using the %SELECT keyword

record_type	What is selected
OLDS	Specifies that OLDSs are to be selected. If dual logging is in effect, both PRIOLDS and SECOLDS can be selected.
SLDS	Specifies that PRISLDs are to be selected. The PRISLD is selected unless the SLDS record in the RECON data set shows the SLDS has an error. In this case, the SECSLD is selected. The SLDS is the one created by the Log Archive utility when archiving OLDSs, not the one created by an IMS batch region. To select an IMS batch SLDS, specify RLDS.
SSLDS	Specifies that SECSLDs are to be selected.
RLDS	Specifies that RLDSs are to be selected. The PRIRLDS is selected unless the PRILOG record in the RECON data set indicates the RLDS has an error. In this case, the SECRLDS is selected. RLDS refers to both the RLDS created by the Log Archive utility and the SLDS created by an IMS batch region.
SRLDS	Specifies that SECRLDSs are to be selected.
IC	Specifies that image copy data sets are to be selected.
CA	Specifies that change accumulation data sets are to be selected.
ALLOC	Specifies that DBDS allocation records are to be selected.
DBDS	Specifies that DBDS records are to be selected.

Understanding the selection criteria parameter

The selection criteria depend on the record type you select. Selection criteria are described under the sections on selecting individual record types.

The following common terms, used for selection criteria, are used in the remainder of this chapter.

dbds_qualifier
Specifies the DBDS with which the selected records are to be associated. The DBDS can be specified as dbname, ddname, or CA group name. When a CA group name is specified, all DBDSs in the CA group are used for selection. The DBDS qualifier is used when selecting:

- RLDSs
- Change accumulation data sets
- Image copy data sets
- ALLOC records
- DBDSs

time_qualifier
Specifies a time stamp or a range of time stamps.

DBRC selects RECON records by their record key. Many records contain a time stamp and the time that is contained in the record key is signified by an adjacent asterisk (*) in a listing. The time qualifier that is specified in a FROMTIME or TOTIME parameter determines what records DBRC selects.

Some records such as PRILOG or PRISLD records consist of multiple DSN entries, each of which has a start time and stop time. DBRC cannot select specific DSN entries without first selecting the entire log record. The

FROMTIME and TOTIME values must be specified such that the entire log record that contains the required DSN entries is selected based on the time stamp that is in the record key.

For example, if you specify a FROMTIME of 12:00, DSN entries with time stamps later than 12:00 (but that are included in a PRISLDS record with a start time of 11:00) would not be selected and displayed by DBRC, because the PRISLDS record itself has a time stamp earlier than the specified FROMTIME. You can specify a zero time value.

FIRST
Specifies that the oldest record is to be selected.

LAST
Specifies that the most recent record is to be selected.

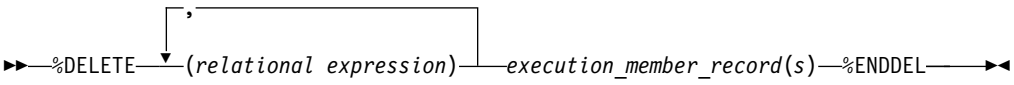
(FROM(*time*),TO(*time*)) or (FROM(*time*)) or FROM(*time*) or (TO(*time*)) or TO(*time*)
Specifies that all records with time greater than or equal to the FROM time and less than or equal to the TO time are to be selected.

ALL
Specifies that all records are to be selected.

%DELETE and %ENDDEL control keywords

The %DELETE keyword deletes records from the generated output stream. Deletion occurs based on a specific condition. The %ENDDEL keyword delimits the scope of the %DELETE keyword. These control keywords always occur in pairs. A %DELETE keyword is followed by one or more execution member records, which is followed by the %ENDDEL keyword. This sequence of records is called a control group or, more specifically, a delete group.

Use the %DELETE and %ENDDEL keywords to delete records from the output stream based on a specific condition. The syntax of a delete group follows:



%DELETE statements cannot be nested. Each %DELETE keyword must be followed by a corresponding %ENDDEL before another %DELETE keyword is encountered.

The relational expression must be of the form %keyword op 'value' or %keyword op '%userkey' where:

- %keyword is any symbolic keyword.
- 'value' is any character string enclosed in single quotation marks. A null string ('') can be specified for the value. You can specify a zero time value.
- %userkey is any keyword defined through the USERKEYS parameter in the GENJCL command. The %userkey must be enclosed in quotation marks and the %userkey value must exclude leading zeros.
- op is one of the following operators:

  EQ Equal

  NE Not equal

  LT Less than

LE Less than or equal

GT Greater than

GE Greater than or equal

When a %DELETE keyword is encountered in a skeletal JCL execution member, the relational expression is evaluated. If the expression is true, the delete group is deleted from the output stream. If the expression is false, the applicable records are copied to the output stream after keywords are resolved. If a value has not been assigned to a keyword, the value is the null string (''). If an undefined keyword is encountered in the skeletal JCL, an error message is received and no substitution takes place.

Specifying complex expressions

You can specify complex expressions consisting of multiple relational expressions joined by connectives

Definitions: A connective is one of the following logical functions:

& AND function

| OR function

The following is an example of a complex expression:

```
%DELETE (relexp1 | relexp2 & relexp3)
```

The DELETE group is deleted when the entire complex expression is logically true. Complex expressions should have the following characteristics:

- The entire DELETE statement (including the %DELETE) is limited to 80 characters, within which up to five expressions are allowed.
- A connective must be the first character following a relational expression (blanks are optional).
- The statement is processed from left to right with no connective priority and no bracketing.

where:

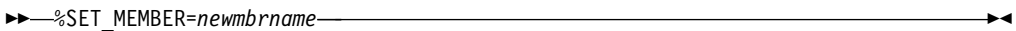
```
relexpx = relational expression
```

This complex expression takes the results of the OR operation between relexp1 and relexp2 and performs the AND operation with relexp3.

%SET MEMBER control keyword

The %SET MEMBER keyword specifies a different skeletal JCL execution member that is to be used in the next step of a multistep job. The %SET MEMBER keyword can be used when you are generating multistep jobs (such as GENJCL.CA with the VOLNUM parameter specified). You use %SET MEMBER to specify a different skeletal JCL execution member than the one that is executed for the first step of the job. The execution member you specify is used in all job steps after the first. You can explicitly code various %keywords in the execution member that is used in job steps after the first one. For example, you can explicitly code the %CAODSN keyword, which is the name of the input change accumulation data set.

The syntax of the %SET MEMBER keyword is:



The %SET MEMBER keyword can be placed anywhere in the current skeletal JCL execution member. However, it takes effect only after processing of the current execution member is complete. If you specify more than one %SET MEMBER keyword, the last one specified is the one that is used. In the new member, you can place a %SET statement that specifies any member name.

newmbrname is the name of the skeletal JCL execution member that is to be used for all job steps after the first job step. newmbrname must reside in the library named in the JCLPDS DD statement. newmbrname is not used until it is necessary to begin processing of the new member. It is possible to specify an incorrect member name and not have an error condition occur until a GENJCL command is issued that causes enough steps to be generated to cause the member to be read.

%SET TIMEFMT control keyword

The %SET TIMEFMT keyword is used to specify a format for time stamps that appear in GENJCL output. JCL from previous versions with time stamps that were not set to include microseconds cannot be reused.

For GENJCL.USER, the default is TIMEFMT(O,O,C,2,6).

Note that the GENJCL TIMEFMT default values have been chosen to produce correct output with IBM-supplied skeletal JCL. If you use the %SET statement to change the TIMEFMT values in a way that affects the values substituted into the IBM-supplied JCL statements, the results might be invalid.

Example:

Here is an example of the %SET TIMEFMT keyword in skeletal JCL.

```
%SET TIMEFMT(,N)
%SELECT RLDS(%SSID,LAST)
LOGEND =%LOGETIM
%ENDSEL
```

And here is what the output from the preceding example of %SET would render:

```
LOGEND =96011131500023456
```

The next four examples are based on the following skeletal JCL member (called USER01) that is used with GENJCL.USER.

```
%SELECT RLDS(%SSID,LAST)
 LOGETIM=%LOGETIM
%ENDSEL
```

- This sample output format was obtained by using the USER01 JCL, specifying SSID(XXXX), and using the default for TIMEFMT, which is:

  ```
  TIMEFMT(O,O,C,2,6)
  LOGETIM=96002131400123456-0700
  ```

- This sample output format was obtained by using the USER01 JCL, specifying SSID(XXXX), and using the default for TIMEFMT on an open log.

  ```
  LOGETIM=00000000000000000+0000
  ```

- This sample output format was obtained by using the USER01 JCL, specifying SSID(XXXX), and using the specification, TIMEFMT(,N).

  ```
  LOGETIM=96011131454423456
  ```

- This sample output format was obtained by using the USER01 JCL, specifying SSID(XXXX), and using the specification, TIMEFMT(,,P,4).

  ```
  LOGETIM=1996.011 13:15:00.023456 -07:00
  ```

- This sample output format was obtained by using the USER01 JCL, specifying SSID(XXXXX), and using the specification, TIMEFMT(,,P,4,1).

  ```
  LOGETIM=1996.011 13:15:00.0 -07:00
  ```

Restriction: The %SET TIMEFMT keyword affects GENJCL output only if it is issued through the GENJCL command or from a %SET statement in the skeletal JCL.

The syntax of the %SET TIMEFMT keyword is:

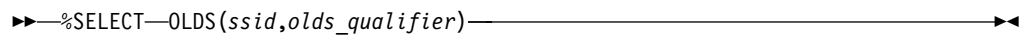
Conventions for control keywords

When writing control keywords, observe the following conventions and restrictions:

- Control keywords must begin in column 1 of a skeletal JCL execution member record.
- Everything specified for the keyword must be contained on one record. Any data following the control statement is ignored.
- Any number of skeletal JCL execution member records can be contained in a control group.
- Delete groups and select groups cannot be nested. However, a select group can be contained within a delete group, or a delete group can be contained within a select group.
- Execution member records containing control keywords are not copied to the output stream.

Selecting OLDSs

The syntax of the %SELECT keyword to select OLDSs is as follows:



ssid
Subsystem ID of the IMS online control region that created the OLDS.

olds_qualifier
Specifies the OLDSs that are to be selected as follows:

INUSE
Specifies that the OLDS that is currently in use by the specified subsystem is to be selected. If dual logging is in effect, both the primary and secondary OLDSs are selected.

LATEST
Specifies that the OLDS that was most recently opened by the specified subsystem is to be selected. If dual logging is in effect, both the primary and secondary OLDSs are selected.

UNARCH
Specifies that all unarchived OLDSs for the specified subsystem are to be selected. If dual logging is in effect, both the primary and secondary OLDSs are selected.

(*DDNAME***)**
Specifies one or more OLDSs by ddname. If dual logging is in effect and both the primary and secondary OLDS are to be selected, both ddnames should be specified.

ALL
Specifies that all OLDSs for the specified subsystem are to be selected.

In the execution member records following the %SELECT keyword, you use symbolic keywords to specify the type of information to be gathered for each OLDS record that is selected. The types of information you can gather are:

%OLDSDDN
The ddname of the OLDS.

%OLDSDSN
The data set name of the OLDS.

%OLDSTYP
The OLDS type. DBRC sets the %OLDSTYP to P for primary OLDS or S for secondary OLDS.

%OLDOTIM
The time the OLDS was opened. DBRC sets %OLDOTIM in the form yydddhhmmssthmiju{offset}.

%OLDCTIM
The time the OLDS was closed. DBRC sets %OLDCTIM in the form yydddhhmmssthmiju{offset}. If the OLDS has not been closed, DBRC sets the time to 00000000000000000+0000.

%OLDSSEL
Set to YES if any OLDS was selected. Otherwise, set to NO.

%OLDFRID
The log record sequence number of the first log record of the OLDS.

%OLDLRID
The log record sequence number of the last log record of the OLDS.

Example 1: The following select group generates repeated DD statements for all unarchived OLDSs belonging to subsystem IMSA.

```
%SELECT     OLDS(IMSA,UNARCH)
//%OLDSDDN  DD DSN=%OLDSDSN,DISP=SHR
%ENDSEL
```

The JCL generated by this select group might be:

```
//DFSOLP00  DD  DSN=IMS.OLDSP00,DISP=SHR
//DFSOLS00  DD  DSN=IMS.OLDSS00,DISP=SHR
//DFSOLP01  DD  DSN=IMS.OLDSP01,DISP=SHR
//DFSOLS01  DD  DSN=IMS.OLDSS01,DISP=SHR
```

Example 2: The following select group generates a list of all OLDSs belonging to subsystem IMSA:

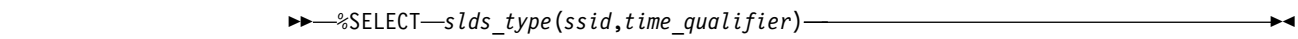
```
%SELECT       OLDS(IMSA,ALL)
%OLDSTYPOLDS  DD NAME=%OLDSDDN
              DSN=%OLDSDSN
              CLOSE TIME=%OLDSCTIM
%ENDSEL
```

The output generated by this select group might be:

```
POLDS  DD NAME=DFSOLP00
       DSN=IMS.POLDS00
       CLOSE TIME=84235163819300000+0012
POLDS  DD NAME=DFSOLP01
       DSN=IMS.POLDS01
       CLOSE TIME=84235171224600000+0055
POLDS  DD NAME=DFSOLP02
       DSN=IMS.POLDS02
       CLOSE TIME=00000000000000000+0000
```

Selecting SLDSs

The syntax of the %SELECT keyword to select SLDS is:



slds_type
Can be specified as SLDS (for the PRISLD) or SSLDS (for the SECSLD). This keyword selects the entire RECON record, not individual data sets. Therefore, all data sets identified in the SLDS record are selected.

If the selected PRILOG data set is marked in error, DBRC selects the associated secondary data set if one that is not also in error exists. If the associated data set exists but is also in error, DBRC selects the original record. If SSLDS is specified, the SECLOG data set is selected regardless of whether it is marked in error. An SLDS record might not contain a DSN entry. In this case, the values for %SLDSDSN, %SLDUNIT, and %SLDVOLS are null. By using the LASTDSET time qualifier you can have the last PRISLDS data set that contains data set entries returned. You must use the %DELETE statement in order to ensure that DBRC substitutes valid data in the generated JCL.

ssid
The subsystem ID (of the IMS online control region) that created the OLDSs that were archived to become SLDSs.

time_qualifier
Specifies a time stamp or a range of time stamps.

DBRC selects RECON records by their record key. Many records contain a time stamp and the time that is contained in the record key is signified by an adjacent asterisk (*) in a listing. The time qualifier that is specified in a FROMTIME or TOTIME parameter determines what records DBRC selects.

Some records such as PRILOG or PRISLD records consist of multiple DSN entries, each of which has a start time and stop time. DBRC cannot select specific DSN entries without first selecting the entire log record. The FROMTIME and TOTIME values must be specified such that the entire log record that contains desired DSN entries is selected based on the time stamp that is in the record key.

For example, if you specify a FROMTIME of 12:00, DSN entries with time stamps later than 12:00 (but that are included in a PRISLDS record with a start

time of 11:00) would not be selected and displayed by DBRC, because the PRISLDS record itself has a time stamp earlier than the specified FROMTIME. You can specify a zero time value.

FIRST
Specifies that the oldest record is to be selected.

LAST
Specifies that the most recent record is to be selected.

LASTDSET
LASTDSET is similar to LAST, but this keyword ignores empty log records and returns the last SLDS that contains DSN entries. This keyword is valid only for RLDS or SLDS (including secondary). If LASTDSET is specified for any other record type, it will default to LAST.

(FROM(*time***),TO(***time***)) or (FROM(***time***)) or FROM(***time***) or (TO(***time***)) or TO(***time***)**
Specifies that all records with time greater than or equal to the FROM time and less than or equal to the TO time are to be selected.

ALL
Specifies that all records are to be selected.

In the execution member records following the `%SELECT` keyword, you specify (using symbolic keywords) the type of information to be gathered for each SLDS record that is selected. The types of information you can gather are:

%SLDSDSN
The data set name of the SLDS.

%SLDUNIT
The unit type of the SLDS.

%SLDVOLS
The volume serial number of the SLDS.

%SLDFSEQ
The file sequence number of the SLDS.

%SLDSTIM
The start time of the SLDS. DBRC sets the `%SLDSTIM` in the form `yydddhhmmssthmiju{offset}`.

%SLDETIM
The stop time of the SLDS. DBRC sets the `%SLDETIM` in the form `yydddhhmmssthmiju{offset}`.

%SLDOTIM
The start time of the PRISLDS/SECSLDS. DBRC sets `%SLDOTIM` in the form *yydddhhmmssthmiju*{*offset*}}.

%SLDSSEL
Set to `YES` if any SLDS was selected. Otherwise, set to `NO`.

%SLDRMT
Set to `YES` if the SLDS was created at the tracking site. Otherwise, set to `NULL`.

%SLDFRID
The log record sequence number of the first log record of the SLDS.

%SLDLRID
The log record sequence number of the last log record of the SLDS.

%SLDCHKT

The Checkpoint Types value of the SLDS data set entry. This is a hexadecimal representation of the bit settings of the value where:

- Bit 0 - simple checkpoint
- Bit 1 - snapq checkpoint
- Bit 2 - dumpq checkpoint
- Bit 3 - purge checkpoint
- Bit 4 - freeze checkpoint
- Bit 5-7 - not used

Example 1: The following select group generates the most recent SLDS for subsystem IMSA.

```
%SELECT       SLDS(IMSA,LAST)
LATEST SLDS:  DSN=%SLDSDSN
               STOP TIME=%SLDETIM
%ENDSEL
```

The output generated by this select group might be:

```
LATEST SLDS:  DSN=IMS.SLDS
               STOP TIME=84123081233900000
```

If the SLDS record has more than one data set, then all the data sets to be selected and your output may look like this:

```
LATEST SLDS:  DSN=IMS.IMSA.SLDSP.D97107.T1405235.V06
               STOP TIME=97107142046900000+0100
LATEST SLDS:  DSN=IMS.IMSA.SLDSP.D97107.T1420469.V03
               STOP TIME=97107142057900000+0100
LATEST SLDS:  DSN=IMS.IMSA.SLDSP.D97107.T1420579.V00
               STOP TIME=97107143008700000+0100
```

Example 2: The following select group generates a concatenated DD statement for all SLDSs for subsystem IMSA that have an open time greater than or equal to 840031903298.

```
%SELECT   SLDS(IMSA,FROM(840031903298))
%DELETE   (%SLDSDSN EQ '')
//SLDS    DD  DSN=%SLDSDSN,DISP=OLD,
//            UNIT=%SLDUNIT,
//            VOL=SER=(%SLDVOLS),
//            LABEL=(1,SL)
%ENDDEL
%ENDSEL
```

The generated DD statements might be:

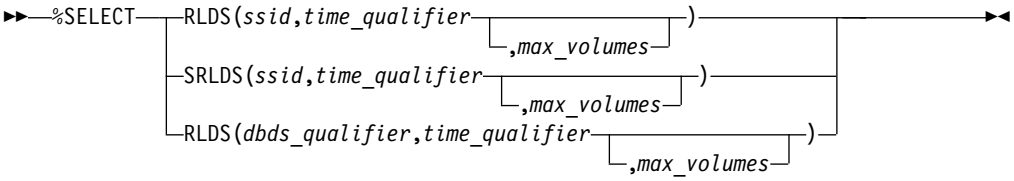
```
//SLDS    DD  DSN=IMS.SLDS1,DISP=OLD,
//            UNIT=3400,
//            VOL=SER=(VOLUM1,VOLUM2,VOLUM3),
//            LABEL=(1,SL)
//        DD  DSN=IMS.SLDS2,DISP=OLD,
//            UNIT=3400,
//            VOL=SER=(VOLUM4,VOLUM5,VOLUM6,
//            VOLUM7,                                C
// VOLUM8,VOLUM9),
//            LABEL=(1,SL)
```

In this example, a JCL continuation card was generated. This is because the volume serial number list was longer than the output record.

The %DELETE statement prevents the JCL statement from being generated for an SLDS record that does not contain a DSN entry.

Selecting RLDSs

The syntax of the %SELECT keyword to select RLDSs can be specified as follows:



Use SRLDS to request that secondary RLDS records be selected. Secondary RLDS records can be specifically requested only when you specify SSID. When you specify *dbds_qualifier*, you are specifically requesting primary RLDS records. If the primary RLDS is marked in error, DBRC selects the associated secondary data set if one that is not also in error exists. If no associated data set exists or if it is also in error, DBRC selects the original record.

An RLDS record might not contain a DSN entry. In this case, the values for %LOGDSN, %LOGUNIT, and %LOGVOLS are null. By using the LASTDSET time qualifier, you can have the last PRILOG data set that contains data set entries returned.

ssid
The subsystem ID of the IMS online control region or an IMS batch region. PRILOG (or SECLOG) records corresponding to the specified SSID are selected to satisfy the specified search criteria. Because RECON records are selected, all data sets identified by the record are selected.

time_qualifier
Specifies a time stamp or a range of time stamps.

DBRC selects RECON records by their record key. Many records contain a time stamp and the time that is contained in the record key is signified by an adjacent asterisk (*) in a listing. The time qualifier that is specified in a FROMTIME or TOTIME parameter determines what records DBRC selects.

Some records such as PRILOG or PRISLD records consist of multiple DSN entries, each of which has a start time and stop time. DBRC cannot select specific DSN entries without first selecting the entire log record. The FROMTIME and TOTIME values must be specified such that the entire log record that contains desired DSN entries is selected based on the time stamp that is in the record key.

For example, if you specify a FROMTIME of 12:00, DSN entries with time stamps later than 12:00 (but that are included in a PRISLDS record with a start time of 11:00) would not be selected and displayed by DBRC, because the PRISLDS record itself has a time stamp earlier than the specified FROMTIME. You can specify a zero time value.

FIRST
Specifies that the oldest record is to be selected.

LAST
Specifies that the most recent record is to be selected.

LASTDSET
LASTDSET is similar to LAST, but this keyword ignores empty log records and returns the last SLDS that contains DSN entries. This keyword is valid only for RLDS or SLDS (including secondary). If LASTDSET is specified for any other record type, it will default to LAST.

(FROM(*time*),TO(*time*)) or (FROM(*time*)) or FROM(*time*) or (TO(*time*)) or TO(*time*)
Specifies that all records with time greater than or equal to the FROM time and less than or equal to the TO time are to be selected.

ALL
Specifies that all records are to be selected.

dbds_qualifier
Specifies the DBDS with which the selected records are to be associated. The DBDS can be specified as dbname, ddname, or CA group name. When a CA group name is specified, all DBDSs in the CA group are used for selection. The DBDS qualifier is used when selecting:

- RLDSs
- Change accumulation data sets
- Image copy data sets
- ALLOC records
- DBDSs

When a *dbds_qualifier* is specified, only RLDSs that contain log records corresponding to the specified DBDS are selected. (In other words, those RLDSs for which an ALLOC record exists in the RECON data set.) Only primary RLDSs can be selected when the *dbds_qualifier* is specified.

max_volumes
The maximum number of log volumes to be selected. If `max_volumes` is specified, processing of the select group terminates when the specified number of log volumes is reached. If `max_volumes` is specified and a log merge situation exists, more than the specified number of volumes can be selected. This is to ensure that a valid subset of logs is selected.

In the execution member records following the `%SELECT` keyword, you use symbolic keywords to specify the type of information to be gathered for each RLDS record that is selected. The types of information you can gather are:

%LOGDSN
The data set name of the RLDS.

%LOGFSEQ
The file sequence number of the RLDS.

%LOGUNIT
The unit type of the RLDS.

%LOGVOLS
The volume serial number of the RLDS.

%LOGSTIM
The start time of the RLDS. DBRC sets `%LOGSTIM` in the form `yydddhhmmssthmiju{offset}`.

%LOGETIM
The stop time of the RLDS. DBRC sets %LOGETIM in the form yydddhhmmssthmiju{offset}. If the data set is still open, the time is set to 000000000000+0000.

%LOGSEL
Set to YES if any log data sets were selected. Otherwise, set to NO.

%LOGMERG
Set to YES if a log merge is required. Otherwise, set to NO. %LOGMERG is always set to NO if SSID is specified.

%LOGONL
Set to YES if the RLDS is associated with an online region. Set to NO for batch logs.

%LOGRMT
Set to YES if the RLDS was created at the tracking site. Otherwise, set to NULL.

%LOGFRID
The log record sequence number of the first log record of the RLDS.

%LOGLRID
The log record sequence number of the last log record of the RLDS.

%LOGOTIM
The start time of the PRILOG/SECLOG. DBRC sets %LOGOTIM in the form yydddhhmmssthmiju{offset}.

Example: The following select group generates a DD statement for the most-recent RLDS for subsystem BATCHJOB. This example assumes the RLDS is still open.

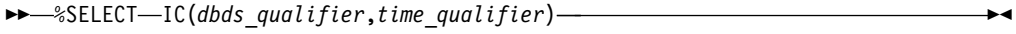
```
%SELECT   RLDS(BATCHJOB,LAST)
%DELETE   (%LOGETIM NE '00000000000000000+0000')
//LOGDD   DD  DSN=%LOGDSN,DISP=OLD,
//            UNIT=%LOGUNIT,
//            VOL=SER=(%LOGVOLS),
//            LABEL=(%LOGFSEQ,SL)
%ENDDEL
%ENDSEL
```

If no RLDS is recorded in the RECON data set for the subsystem or if the most-recent RLDS has been closed, no DD statement is generated. Otherwise, the generated DD statement might be:

```
//LOGDD   DD  DSN=IMS.RLDS,DISP=OLD,
//            UNIT=3400,
//            VOL=SER=(VOLUM1,VOLUM2)
//            LABEL=(1,SL)
```

Selecting image copy data sets

The syntax of the %SELECT keyword to select image copy data sets is:



dbds_qualifier
Specifies the DBDS with which the selected records are to be associated. The DBDS can be specified as dbname, ddname, or CA group name. When a CA group name is specified, all DBDSs in the CA group are used for selection. The DBDS qualifier is used when selecting:

- RLDSs
- Change accumulation data sets
- Image copy data sets
- ALLOC records
- DBDSs

time_qualifier
Specifies a time stamp or a range of time stamps.

DBRC selects RECON records by their record key. Many records contain a time stamp and the time that is contained in the record key is signified by an adjacent asterisk (*) in a listing. The time qualifier that is specified in a FROMTIME or TOTIME parameter determines what records DBRC selects.

Some records such as PRILOG or PRISLD records consist of multiple DSN entries, each of which has a start time and stop time. DBRC cannot select specific DSN entries without first selecting the entire log record. The FROMTIME and TOTIME values must be specified such that the entire log record that contains the required DSN entries is selected based on the time stamp that is in the record key.

For example, if you specify a FROMTIME of 12:00, DSN entries with time stamps later than 12:00 (but that are included in a PRISLDS record with a start time of 11:00) would not be selected and displayed by DBRC, because the PRISLDS record itself has a time stamp earlier than the specified FROMTIME. You can specify a zero time value.

FIRST
Specifies that the oldest record is to be selected.

LAST
Specifies that the most recent record is to be selected.

(FROM(*time***),TO(***time***)) or (FROM(***time***)) or FROM(***time***) or (TO(***time***)) or TO(***time***)**
Specifies that all records with time greater than or equal to the FROM time and less than or equal to the TO time are to be selected.

ALL
Specifies that all records are to be selected.

In the execution member records following the `%SELECT` keyword, you specify (using symbolic keywords) the type of information to be gathered for each image copy record that is selected. If the duplicate image copy is marked in error, the DBRC selects the primary image copy. The types of information you can gather are:

%ICDSN The data set name of the image copy data set.

%ICTYPE
The image copy's type: AVAIL, BATCH, ONLINE, CIC, SMSCIC, SMSNOCIC, SMSOFFLC, and SMSONLC.

%ICFSEQ
The file sequence number of the image copy data set if it is a NONHSSP type; otherwise, `ICFSEQ` is null.

%ICSEL Set to `YES` if any image copy data set was selected. Otherwise, `ICSEL` is set to `NO`.

%ICSTOP
The stop time of the image copy data set ID that is present; otherwise ICSTOP is null.

%ICTIME
The run time of the image copy. DBRC sets %ICTIME in the form yydddhhmmssthmiju{offset}.

%ICUNIT
The unit type of the image copy data set if it is a NONHSSP type; otherwise, ICUNIT is null.

%ICVCNT
The number of volumes of the image copy data set if it is a NONHSSP type; otherwise, ICVCNT is null.

%ICVOLS
The volume serial number list of the image copy data set if it is a NONHSSP type; otherwise, ICVOLS is null.

%ICUSID
The update set identifier (USID).

%ICCAT Set to YES if the image copy is cataloged (HSSP CICs only). Otherwise, ICCAT is set to NO.

%IC2SEL
Set to YES if a duplicate image copy data set is associated with the selected image copy data set. Otherwise, IC2SEL is set to NO.

The following keywords are set only when a duplicate image copy data set exists; otherwise, they are null:

%IC2DSN
The data set name of the duplicate image copy data set.

%IC2FSEQ
The file sequence number of the duplicate image copy data set. If the IC was created by HSSP, IC2FSEQ is set to null.

%IC2UNIT
The unit type of the duplicate image copy data set. If the IC was created by HSSP, IC2UNIT is set to null.

%IC2VCNT
The number of volumes of the duplicate image copy data set. If the IC was created by HSSP, IC2VCNT is set to null.

%IC2VOLS
The volume serial number list of the duplicate image copy data set. If the IC was created by HSSP, IC2VOLS is set to null.

Example: The following select group generates a DD statement for the oldest image copy data set for the DBDS with a database name of SHISAMDB and a ddname of SHISAMDD.

```
%SELECT   IC((SHISAMDB,SHISAMDD),FIRST)
//ICDD    DD  DSN=%ICDSN,DISP=OLD,
//            VOL=SER=(%ICVOLS),
//            UNIT=%ICUNIT,
//            LABEL=(%ICFSEQ,SL)
%ENDSEL
```

The generated DD statement might be:

```
//ICDD    DD  DSN=SHISAMDB.SHISAMDD.IC,DISP=OLD,
//            VOL=SER=(VOLUM1),
//            UNIT=3400,
//            LABEL=(1,SL)
```

Selecting change accumulation data sets

The syntax of the %SELECT keyword to select change accumulation data sets is:

dbds_qualifier
Specifies the DBDS with which the selected records are to be associated. The DBDS can be specified as dbname, ddname, or CA group name. When a CA group name is specified, all DBDSs in the CA group are used for selection. The DBDS qualifier is used when selecting:

- RLDSs
- Change accumulation data sets
- Image copy data sets
- ALLOC records
- DBDSs

time_qualifier
Specifies a time stamp or a range of time stamps.

DBRC selects RECON records by their record key. Many records contain a time stamp and the time that is contained in the record key is signified by an adjacent asterisk (*) in a listing. The time qualifier that is specified in a FROMTIME or TOTIME parameter determines what records DBRC selects.

Some records such as PRILOG or PRISLD records consist of multiple DSN entries, each of which has a start time and stop time. DBRC cannot select specific DSN entries without first selecting the entire log record. The FROMTIME and TOTIME values must be specified such that the entire log record that contains the required DSN entries is selected based on the time stamp that is in the record key.

For example, if you specify a FROMTIME of 12:00, DSN entries with time stamps later than 12:00 (but that are included in a PRISLDS record with a start time of 11:00) would not be selected and displayed by DBRC, because the PRISLDS record itself has a time stamp earlier than the specified FROMTIME. You can specify a zero time value.

FIRST
Specifies that the oldest record is to be selected.

LAST
Specifies that the most recent record is to be selected.

(FROM(*time*),TO(*time*)) or (FROM(*time*)) or FROM(*time*) or (TO(*time*)) or TO(*time*)
Specifies that all records with time greater than or equal to the FROM time and less than or equal to the TO time are to be selected.

ALL
Specifies that all records are to be selected.

In the execution member records following the %SELECT keyword, you use symbolic keywords to specify the type of information to be gathered for each change accumulation record that is selected. The types of information you can gather are:

%CADSN The change accumulation data set name.

%CAFSEQ
The file sequence number of the change accumulation data set.

%CAUNIT
The unit type of the change accumulation data set.

%CAVCNT
The number of volumes of the change accumulation data set.

%CAVOLS
The volume serial number list of the change accumulation data set.

%CALGTM
The volume stop time of the last log volume that was used as input to the change accumulation data set. DBRC sets %CALGTM in the form yydddhhmmssthmiju{offset}.

%CATIME
The change accumulation data set time in the form yydddhhmmssthmiju{offset}.

%CASEL Set to YES if any change accumulation data sets are selected. Otherwise, set to NO.

Example: The following select group lists all change accumulation data sets created since time 07231000000000000+0000 for CA group CAGRP1.

```
%SELECT   CA((CAGRP1),FROM(07231000000000000+0000))
          DSNAME=%CADSN
               VOLUMES=%CAVOLS
               RUNTIME=%CATIME
               LOGTIME=%CALGTM
%ENDSEL
```

The generated output might be:

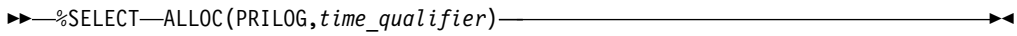
```
    DSNAME=CAGRP1.DSN1
         VOLUMES=VOLUM1,VOLUM2,VOLUM3,
          VOLUM4,                              C
//  VOLUM5,VOLUM6
         RUNTIME=07231061823000000
         LOGTIME=07230231555700000
    DSNAME=CAGRP1.DSN2
         VOLUMES=VOLUM1,VOLUM2
         RUNTIME=07236182444300000
         LOGTIME=07236093451900000
```

In this example, the volume serial number list for the first data set does not fit on the output record. Therefore, a JCL continuation statement is generated (even though JCL is not being generated).

Selecting DBDS allocation records

The syntax of the %SELECT keyword to select ALLOC records can be one of the following:

```
►►—%SELECT—ALLOC(dbds_qualifier,time_qualifier)———————————————————►◄
```



dbds_qualifier
Specifies the DBDS with which the selected records are to be associated. The DBDS can be specified as dbname, ddname, or CA group name. When a CA group name is specified, all DBDSs in the CA group are used for selection. The DBDS qualifier is used when selecting:

- RLDSs
- Change accumulation data sets
- Image copy data sets
- ALLOC records
- DBDSs

When a *dbds_qualifier* is specified, ALLOC records corresponding to the specified DBDSs are selected.

time_qualifier
Specifies a time stamp or a range of time stamps.

DBRC selects RECON records by their record key. Many records contain a time stamp and the time that is contained in the record key is signified by an adjacent asterisk (*) in a listing. The time qualifier that is specified in a FROMTIME or TOTIME parameter determines what records DBRC selects.

Some records such as PRILOG or PRISLD records consist of multiple DSN entries, each of which has a start time and stop time. DBRC cannot select specific DSN entries without first selecting the entire log record. The FROMTIME and TOTIME values must be specified such that the entire log record that contains the required DSN entries is selected based on the time stamp that is in the record key.

For example, if you specify a FROMTIME of 12:00, DSN entries with time stamps later than 12:00 (but that are included in a PRISLDS record with a start time of 11:00) would not be selected and displayed by DBRC, because the PRISLDS record itself has a time stamp earlier than the specified FROMTIME. You can specify a zero time value.

FIRST
Specifies that the oldest record is to be selected.

LAST
Specifies that the most recent record is to be selected.

(FROM(*time***),TO(***time***)) or (FROM(***time***)) or FROM(***time***) or (TO(***time***)) or TO(***time***)**
Specifies that all records with time greater than or equal to the FROM time and less than or equal to the TO time are to be selected.

ALL
Specifies that all records are to be selected.

All ALLOC records for the specified DBDS with allocation times within the bounds of the specified *time_qualifier* are selected. When PRILOG is specified, all ALLOC records corresponding to PRILOG records within the specified time bounds are selected.

In the execution member records following the %SELECT keyword, you use symbolic keywords to specify the type of information to be gathered for each ALLOC record that is selected. The types of information you can gather are:

%DBNAME
The database name.

%DBDDN The database ddname or area name.

%ALLTIME
The allocation time stamp in the form `yydddhhmmssthmiju{offset}`.

%DALTIME
The deallocation time stamp in the form `yydddhhmmssthmiju{offset}`. Set to `000000000000+0000` if there is no deallocation time stamp.

%ALLDSSN
The data set sequence number.

%PLGTIME
The start time of the corresponding PRILOG record.

%ALLSEL
Set to `YES` if any ALLOC records are selected. Otherwise, `ALLSEL` is set to `NO`.

%ALLUSID
The update set identifier (USID)

%DBTYPE
The type of database that the selected ALLOC record is associated with. The %DBTYPE value can be one of the following:
FP Area of a Fast Path database
DLI Non-HALDB
PDATA
Data DBDS of a HALDB

Example: The following select group generates a list of information about all ALLOC records for the DBDS with a database name of SHISAMDB and ddname of SHISAMDD:

```
%SELECT ALLOC((SHISAMDB,SHISAMDD),ALL)
        DBNAME        %DBNAME
        DDNAME        %DBDDN
        ALLOC time    %ALLTIME
        DEALL time    %DALTIME
        PRILOG time   %PLGTIME
```

The generated output might be:

```
        DBNAME        SHISAM
        DDNAME        SHISAM
        ALLOC TIME    83256080000000000+0000
        DEALL TIME    00000000000000000+0000
        PRILOG TIME   83256063000000000+0000
```

Selecting DBDS records

The syntax of the `%SELECT` keyword to select DBDS records is:

dbds_qualifier
Specifies the DBDS with which the selected records are to be associated. The DBDS can be specified as dbname, ddname, or CA group name. When a CA group name is specified, all DBDSs in the CA group are used for selection. The DBDS qualifier is used when selecting:

- RLDSs
- Change accumulation data sets
- Image copy data sets
- ALLOC records
- DBDSs

For DEDBs, the select group is processed once for each defined area data set (ADS) for each specified area. For other types of databases, the select group is processed once for each specified DBDS.

In the execution member records following the %SELECT keyword, you use symbolic keywords to specify the type of information to be gathered for each DBDS record that is selected. The types of information you can gather are:

%DBNAME
The database name.

%DBDDN The DBDS ddname or DEDB area name.

%DBTYPE
Set to FP when the selected DBDS is an area of a Fast Path database. Set to DLI for DBDSs of non-HALDB databases. Set to PDATA for data DBDSs of HALDB databases. Set to PINDEX for primary index DBDSs of HALDB databases. Set to PILDS for ILDS DBDSs of HALDB databases.

%DBDSN The data set name of the DBDS or ADS.

%DBADDN
For DEDBs, the ddname of the ADS. For other types of databases, DBADDN is set to null.

%DBADSAV
For DEDBs, set to AVAIL if the ADS is indicated as available in the RECON data set. Set to UNAVAIL if the ADS is unavailable. For other types of databases, DBADSAV is set to null.

%DBDSSEL
Set to YES if any DBDS records are selected. Otherwise, DBDSDEL is set to NO.

%DBUSID
For DEDBs, the update set identifier (USID) of the area. For other types of databases, DBUSID is set to NULL.

%DBDSNRV
Set to YES if the DBDS is nonrecoverable. Otherwise, DBDSNRV is set to NO.

Example: The following select group generates a series of DD statements for available area data sets for the area named DBHVSAM1. This area is in the DEDB named DIVNTZ04.

```
%SELECT     DBDS((DIVNTZ04,DBHVSAM1))
%DELETE     (%DBADSAV ne 'AVAIL')
//%DBADDN   DD  DSN=%DBDSN,DISP=OLD
%ENDDEL
%ENDSEL
```

The generated output might be:

```
//FP1ADD1  DD  DSN=IMS.FP1ADD1,DISP=OLD
//FP1ADD2  DD  DSN=IMS.FP1ADD2,DISP=OLD
```

Writing skeletal JCL default members

Skeletal JCL default members are used to set default values for keywords you have defined in the skeletal JCL execution members. The use of default members is optional. You must supply any default members to be used.

Default members can have two types of records: assignment records or comment records. Assignment records assign default values to user-defined keywords. Assignment records must contain a percent sign (%) in column 1. If a record does not contain a percent sign in column 1, it is a comment record, which DBRC ignores.

The value assigned to the keyword can be any text string, including the null string (''). If the value contains a single quotation mark, you must use two single quotation marks. The entire value must be contained on one record. Any data following the closing single quotation mark is ignored. A closing single quotation mark is required. If a closing single quotation mark is missing, an error message is generated and the GENJCL command fails.

Example: A default member contains these records:

```
%DEDBNAM = 'DIVNTZ04'
%AREANAM = 'DBHVSAM1'
```

A skeletal JCL member contains:

```
DATABASE NAME = %DEDBNAM
AREA NAME = %AREANAM
```

Using the DEFAULTS parameter and assuming the values are not overridden, the GENJCL command generates the following:

```
DATABASE NAME = DIVNTZ04
AREA NAME = DBHVSAM1
```

Specifying default members

You can specify default members either explicitly or implicitly.

Members are explicitly specified using the DEFAULTS parameter on the GENJCL command. Up to 10 default members can be specified.

Implicit specification can be used for the GENJCL commands that apply to a DBDS (GENJCL.IC, GENJCL.OIC, and GENJCL.RECOV) or CA group (GENJCL.CA). In addition, implicit specification can be used on the GENJCL.USER command. The default members to be implicitly used are specified using the DEFLTJCL parameter on the INIT.DBDS, CHANGE.DBDS, INIT.CAGRP, and CHANGE.CAGRP commands. Only one default member is allowed per DBDS or CA group.

The use of an implicit default member can be overridden with the NODEFLT parameter on the GENJCL command. When both explicitly and implicitly specified default members are used, explicitly specified members have precedence. That is, if a keyword is assigned a value in both members, the value assigned by the explicitly specified member is used.

If a keyword is assigned a value both in a default member and in the USERKEYS parameter of the GENJCL command, the latter value is used. USERKEYS parameter values override default member values.

Using user-supplied or modified skeletal JCL

Before using your skeletal JCL execution or default members, you must do the following:

- Add the JCLPDS and JCLOUT ddnames to the JCL required to run DBRC. JCLPDS identifies the partitioned data set containing the skeletal JCL execution members. JCLOUT identifies the data set to which the generated job is to be written. Output is in card image format. The output data set can be a punch file, a DASD data set that you plan to examine before submitting the job for execution, or directly to the z/OS internal reader.
- Add to the skeletal JCL execution member any STEPLIB ddnames and job accounting information that your installation requires. If the DD statements contain comments or continuation characters, they are ignored when the JCL is generated. Except for the skeletal JCL member for the JOB statement, do not add to your skeletal JCL any JOBLIB and JES control statements; doing so causes errors if multiple steps are generated.

Related reference:

"DBRC time stamps" on page 52

Symbolic keywords recognized by DBRC

DBRC recognizes symbolic keywords in the IBM-supplied skeletal JCL execution members.

These topics describe the symbolic keywords that are recognized by the recovery-related utilities.

- "Symbolic keywords recognized by all utilities"
- "Symbolic keywords recognized by the Log Archive utility (ARCHJCL)" on page 250
- "Symbolic keywords recognized by the Database Change Accumulation utility (CAJCL)" on page 251
- "Symbolic keywords recognized by the Log Recovery utility (LOGCLJCL)" on page 252
- "Symbolic keywords recognized by the Database Image Copy utilities" on page 253
- "Symbolic keywords recognized by the Database Recovery utility - receive (ICRCVJCL)" on page 255
- "Symbolic keywords recognized by the Database Recovery utility - recovery (RECOVJCL)" on page 256

Symbolic keywords recognized by all utilities

The following table explains the symbolic keywords recognized by all the supported utilities.

Table 7. Symbolic keywords for all supported utilities

Keyword	Description
%RCNDSN1	Name of the RECON1 data set if RECON data sets are allocated by JCL. Set to null if RECON data sets are dynamically allocated.

Table 7. Symbolic keywords for all supported utilities (continued)

Keyword	Description
%RCNDSN2	Name of the RECON2 data set if RECON data sets are allocated by JCL. Set to null if RECON data sets are dynamically allocated.
%RCNDSN3	Name of the RECON3 data set if RECON data sets are allocated by JCL. Set to null if RECON data sets are dynamically allocated.
%STPNO	The current step number. The step number is set to 0 if the JOB parameter was specified on the GENJCL command. The step number is increased by 1 when DBRC first encounters it in a skeletal JCL execution member. The step number remains at that value while the execution member is processed. As the keyword is encountered in the remaining skeletal JCL, the current value is substituted. The JCL execution member can be processed again because of a multi-step generation, or because the subsequent GENJCL command specifies NOJOB. When it is processed again the step number is increased by 1 from its current value when the keyword is first encountered in the next skeletal JCL execution member. This increase takes place before the keyword value is substituted.
%TIME	The time of day, in the form hhmmss.
%GRPINDX	The DBDS group member index. This keyword is set only when a DBDS group is specified, implicitly or explicitly, on the GENJCL command. (A DBDS group can be specified on the GENJCL.IC, GENJCL.OIC, GENJCL.RECOV, and GENJCL.USER commands.) When you specify a DBDS group, the keyword is initialized to 1. It is then increased by 1 as each successive group member is processed.
%CNTR	A counter controlled by DBRC. The counter is set to 0 whenever the first GENJCL command is issued or a JOB statement is reproduced from the skeletal JCL execution member JOBJCL. DBRC increases the counter by 1 each time the keyword is encountered in a skeletal JCL execution member. The JCL execution member can be processed again because of a multi-step generation, or because the subsequent GENJCL command specifies NOJOB. If so, the counter continues to increase from its current value when the keyword is encountered in next skeletal JCL execution member. This increase takes place before the keyword value is substituted.
%DATE	The day of the year, in the form yyddd.
%DATE7	The day of the year, in the form yyyyddd.
%IMSPLEX	The IMSplex name set in the RECON data sets. Set to NULL if DBRC does not register with SCI.
%DBRCGRP	The DBRC group set in the RECON data sets. Set to NULL if DBRC does not register with SCI.

Symbolic keywords recognized by the Log Archive utility (ARCHJCL)

The Log Archive utility recognizes the following symbolic keywords.

Table 8. Symbolic keywords for Log Archive utility

Keyword	Description
%SSID	The subsystem ID, which is set from the SSID parameter on the GENJCL.ARCHIVE command. If the SSID parameter is not specified, the default subsystem ID is used. The default subsystem ID is set by you in the INIT.RECON or CHANGE.RECON command. If no default subsystem ID was specified, the command fails.
%DDNAMES	The ddnames of the OLDSs that are to be archived. If ALL is specified or used as the default on the GENJCL.ARCHIVE command, the ddnames of all unarchived OLDSs are determined from the RECON data set. Otherwise, the ddnames specified on the command are used.

Table 8. Symbolic keywords for Log Archive utility (continued)

Keyword	Description
%OLDSDDN	The ddname of one or more specific OLDS.
%OLDSDSN	The data set name of the one or more OLDS.
%ARDATE	The date (from the open time stamp) of the first OLDS that is to be archived. The date is in the form yyddd where: yy is the year ddd is the day
%ARDATE7	The date (from the open time stamp) of the first OLDS that is to be archived. The date is in the form yyyyddd where: yyyy is the 4-digit year ddd is the Julian day
%ARTIME	The time (from the open time stamp) of the first OLDS that is to be archived. The time is in the form hhmmsst where: hh is the hour mm is the minute ss is the second t is the tenth of a second
%ARVERS	The archive version number of the first OLDS to be archived.
%ARCSLDS	Set to YES when the SLDS parameter is specified.

Symbolic keywords recognized by the Database Change Accumulation utility (CAJCL)

The Database Change Accumulation utility recognizes the following symbolic keywords.

Table 9. Symbolic keywords for Database Change Accumulation utility

Keyword	Description
%CAGRP	The CA group name.
%DSLLGTM	The start time for selecting input log data. If an input change accumulation data set is used, %DSLLGTM is set to the volume stop time of the last-accumulated log volume.
%CAODSN	The data set name of the input change accumulation data set. This keyword is set to null if no existing change accumulation data set is defined in the RECON data set for the CA group.
%CAOUNIT	The unit type of the input change accumulation data set. This keyword is set to null if no existing change accumulation data set is defined in the RECON data set for the CA group.
%CAOVOLS	The volume serial number list of the input change accumulation data set. This keyword is set to null if there is no existing change accumulation data set is defined in the RECON data set for the CA group.
%CAOFSEQ	The file sequence number of the input change accumulation data set. This keyword is set to null if no existing change accumulation data set is defined in the RECON data set for the CA group.
%CANDSN	The data set name of the output change accumulation data set. If REUSE is specified for the CA group, the keyword is set from information in the RECON data set. If NOREUSE is specified, DBRC generates a data set name. The generated name is: IMSVS.cagrpname.CA.CAhhmmss where *cagrpname* is the CA group name, and *hhmmss* is the current time of day.

Table 9. Symbolic keywords for Database Change Accumulation utility (continued)

Keyword	Description
%CANUNIT	The unit type of the output change accumulation data set. If REUSE is specified for the CA group, the keyword is set from information in the RECON data set. If NOREUSE is specified, this keyword is set from the UNIT parameter on the GENJCL.CA command. If UNIT is not specified, the keyword is set to 3400.
%CANVCNT	The number of volumes in the output change accumulation data set. If REUSE is specified for the CA group, the keyword is set from information in the RECON data set. If NOREUSE is specified, this keyword is set from the VOLLIST parameter on the GENJCL.CA command.
%CANVOLS	The volume serial number list of the output change accumulation data set. If REUSE is specified for the CA group, the keyword is set from information in the RECON data set. If NOREUSE is specified, this keyword is set from the VOLLIST parameter on the GENJCL.CA command.
%CABFSEQ	The file sequence number of the output change accumulation data set. If REUSE is specified for the CA group, the keyword is set from information in the RECON data set. If NOREUSE is specified, this keyword is set to 1.
%LOGDSN	The data set name of the log data set.
%LOGUNIT	The unit type of the log data set.
%LOGVSEQ	The volume sequence number of the log data set.
%LOGVOLS	The volume serial numbers of the log data set.
%LOGFSEQ	The file sequence number of the log data set.
%LOGSEL	Set to YES if any log data sets were selected. Otherwise, set to NO.
%CADB0	This keyword generates the DB0 control statements for the Database Change Accumulation utility. **Note:** %CADB0 always generates DB0 control statements with the time stamps in UTC format (indicated by an offset of zeroes).

Symbolic keywords recognized by the Log Recovery utility (LOGCLJCL)

The Log Recovery utility recognizes the following symbolic keywords.

Table 10. Symbolic keywords for Log Recovery utility

Keyword	Description
%SSID	The subsystem ID, which is set from the SSID parameter on the GENJCL.CLOSE command. If the SSID parameter is not specified, the default subsystem ID is used. The default subsystem ID is set by you in the INIT.RECON or CHANGE.RECON command. If no default subsystem ID is specified, the command fails.
%CDDNAME	The ddname of the OLDS to be closed. This keyword is set from the OLDS parameter on the GENJCL.CLOSE command. If GENJCL.CLOSE did not specify an OLDS, the most recent open OLDS for the specified subsystem is used.
%OLDSTYP	The type of OLDS, primary or secondary (set to P or S, respectively).
%OLDSDSN	The data set name of the OLDS.
%WADS	If the OLDS to be closed is currently open, this keyword is set to YES. Otherwise, this keyword is set to NO.
%NDDNAME	The ddname of the 'next OLDS' to be used to close the OLDS. If %WADS is set to NO, this keyword is set to the ddname of the OLDS used immediately after the OLDS being closed. If %WADS is set to YES, this keyword is set to null.
%PDDNAME	The ddname of the immediately prior OLDS to be used to close the OLDS by providing a last block sequence number for base point information.

Symbolic keywords recognized by the Database Image Copy utilities

The Database Image Copy utilities recognize the following symbolic keywords.

Table 11. Symbolic keywords for Database Image Copy utilities

Keyword	Description
%PSB	The PSB name, which is set from the PSB parameter on the GENJCL command. This keyword is applicable only for the Database Online Image Copy utility.
%DBNAME	The database name, which is set from the DBD parameter on the GENJCL command.
%DBDDN	The DBDS ddname, which is set from the DDN parameter on the GENJCL command.
%DBDSN	The DBDS data set name, which is set from the DBDS record in the RECON data set.
%DBDSAM	This keyword is set to VSAM for VSAM DBDS. Otherwise, it is set to null.
%DBADDN	For DEDBs, the ddname of the ADS. Otherwise, set to null. This keyword is applicable only for the Database Image Copy utility.
%DBADSAV	For DEDBs, set to AVAIL if RECON indicates that the ADS is available, or UNAVAIL if the ADS is unavailable. For other types of databases, this keyword is set to null. This keyword is applicable only for the Database Image Copy utility.
%COPIES	The number of image copy data sets to be produced. This keyword is set to 1 or 2 from the COPIES parameter on the GENJCL command.
%MDBNAME	The HALDB master name, if this is a DBDS of a HALDB partition. This keyword is set to NULL for non-HALDB databases.
%SMS	Indicates whether a Database Image Copy 2 (DFSUMDT0) image copy data set is being used for the requested utility execution. If used, the keyword is set to 1; otherwise, the keyword is set to 0.
%ICDDN1, %ICDDN2, %ICDDN3, %ICDDN4	The DD name of the first image copy data set. If multiple image copy data sets are produced, %ICDDN2, %ICDDN3, and %ICDDN4 are similarly set.
%ICSYSIN	The Database Image Copy utility control statement. Columns in the statement are set as follows: **Column** **Setting** **1** D **2** Number of image copy data sets to be produced (either 1 or 2) **4-11** Database name **13-20** ddname of the DBDS **22-30** ddname of the primary image copy data set **31-38** ddname of the duplicate image copy data set, if one is produced. **40-43** Checkpoint interval (applicable only for Online Database Image Copy utility). All other columns are set to blanks.

Table 11. Symbolic keywords for Database Image Copy utilities (continued)

Keyword	Description
%ICDSN1, %ICDSN2, %ICDSN3, %ICDSN4	The data set name of the image copy data set is %ICDSN1. If NOREUSE is specified for the DBDS, DBRC generates the following data set name: IMSVS.dbname.ddname.IC.IChhmmss where: • *dbname* is the database name of the DBDS • *ddname* is the ddname of the DBDs • *hhmmss* is the current time of day If Sameds is specified for a group of DBDSs, DBRC generates the following data set name: IMSVS.GROUP.grpname.IC.IChhmmss where : • *grpname* is the name specified in the GROUP or DBD parameter • *hhmmss* is the current time of day If multiple image copy data sets are to be produced, %ICDSN3, or %ICDSN4 are set similarly.
%ICUNIT1, %ICUNIT2, %ICUNIT3, %ICUNIT4	The unit type of the image copy data set. If NOREUSE is specified for the DBDS, %ICUNIT1 is set from the UNIT parameter on the command. If multiple image copy data sets are to be produced, %ICUNIT2, %ICUNIT3, or %ICUNIT4 are set similarly.
%ICFSEQ1, %ICFSEQ2, %ICFSEQ3, %ICFSEQ4	The file sequence number of the image copy data set. If NOREUSE is specified for the DBDS, %ICFSEQ1 is set to 1. If multiple image copy data sets are to be produced, %ICFSEQ2, %ICFSEQ3, or %ICFSEQ4 are set similarly.
%ICVOLS1, %ICVOLS2, %ICVOLS3, %ICVOLS4	The volume serial number of the image copy data set. If NOREUSE is specified for the DBDS, %ICVOLS1 is set from the VOLLIST parameter on the command. If multiple image copy data sets are to be produced, %ICVOLS2, %ICVOLS3, or %ICVOLS4 are set similarly.
%ICVCNT1, %ICVCNT2, %ICVCNT3, %ICVCNT4	The number of volumes of the image copy data set. If NOREUSE is specified for the DBDS, %ICVCNT1 is set to the number of volumes specified on the VOLLIST parameter on the command. If multiple image copy data sets are to be produced, %ICVCNT2, %ICVCNT3, or %ICVCNT4 is set similarly.
%SMSGRP	A numeric value indicating the status of group processing: **Column Setting** **0** One job step is being generated for each DBDS one or the other or both area. Otherwise a single job is being generated to copy multiple DBDSs one or the other or both areas (Image Copy 2, GROUP or DB without DDN, and ONEJOB were specified). This means that the skeletal JCL member will be processed more than once per job step generated. **1** Processing the only member of the group **2** Processing the first of multiple members **3** Processing an intermediate member **4** Processing the last member
%SMS1DS	A numeric value indicating whether Sameds was specified: **Column Setting** **0** Sameds was not specified **1** Sameds was specified

Table 11. Symbolic keywords for Database Image Copy utilities (continued)

Keyword	Description
%GROUP	A character value containing the value of the GROUP parameter, or the value of the DBD parameter if DDn was omitted (an *implied group*)--otherwise null.
%ICRTYPE	The image copy type being requested by GENJCL.IC or GENJCL.OIC. Valid values are BATCH, ONLINE, CIC, SMSCIC, SMSNOCIC, SMSOFF, SMSOFFLC, and SMSONLC.

Symbolic keywords recognized by the Database Recovery utility - receive (ICRCVJCL)

The Database Recovery utility - receive recognizes the following symbolic keywords.

Table 12. Symbolic keywords for Database Recovery utility - receive (ICRCVJCL)

Keyword	Description
%DBNAME	The database name of the DBDS to be covered. %DBNAME is set from the DBD parameter on the GENJCL.RECEIVE command.
%DBDDN	The ddname of the DBDS, %DBDDN is set from the DDN parameter on the GENJCL.RECEIVE command.
%DBDSN	The data set name of the DBDS, %DBDSN is set from the DBDS record in the RECON data set.
%DBDSAM	Set to VSAM for a VSAM DBDS. Otherwise, set to null.
%DBUSID	The update set identifier for the DBDS.
%ALLUSID	The update set identifier of the most-recent ALLOC record for the DBDS.
%MDBNAME	The HALDB master name, if this is a DBDS of a HALDB partition. This keyword is set to NULL for non-HALDB databases.
%DSLLGTM	The start time for selecting input log data. If an input change accumulation data set is used, %DSLLGTM is set to the volume stop time of the last-accumulated log volume. Otherwise, the keyword value is set to image copy time.
%ICDSN	The data set name of the image copy data set. Set to null if the USEDBDS parameter is specified on the GENJCL.RECEIVE command. Otherwise, set from the image copy record for the DBDS.
%ICUNIT	The unit type of the image copy data set. Set to null if the USEDBDS parameter is specified on the GENJCL.RECEIVE command. Otherwise, set from the image copy record for the DBDS.
%ICVOLS	The volume serial number list of the image copy data set. Set to null if the USEDBDS parameter is specified on the GENJCL.RECEIVE command. Otherwise set from the image copy record for the DBDS.
%ICFSEQ	The file sequence number of the image copy data set. Set to null if the USEDBDS parameter is specified on the GENJCL.RECEIVE command. Otherwise, set from the image copy record for the DBDS.
%ICUSID	The update set identifier for the image copy.
%CADSN	The data set name of the change accumulation data set. Set to null if no change accumulation is available for the DBDS. Otherwise, set from the change accumulation record.
%CAUNIT	The unit type of the change accumulation data set. Set to null if no change accumulation is available for the DBDS. Otherwise, set from the change accumulation record.
%CAVOLS	The volume serial number list of the change accumulation data set. Set to null if no change accumulation is available for the DBDS. Otherwise, set from the change accumulation record.
%CAFSEQ	The file sequence number of the change accumulation data set. Set to null if no change accumulation is available for the DBDS. Otherwise, set from the change accumulation record.

Table 12. Symbolic keywords for Database Recovery utility - receive (ICRCVJCL) (continued)

Keyword	Description
%OLDFLRID	The log record sequence number (log record ID) of the first log record in the OLDS.
%OLDLLRID	The log record sequence number (log record ID) of the last log record in the OLDS. If the OLDS has not been closed, %OLDLLRID is set to null.
%SLDFLRID	The log record sequence number (log record ID) of the first log record in the SLDS.
%SLDFSEQ	The file sequence number of the SLDS.
%SLDLLRID	The log record sequence number (log record ID) of the last log record in the SLDS. If the SLDS has not been closed, %SLDLLRID is set to null.
%SLDUNIT	The unit type of the log data set.
%SLDVOLS	The volume serial numbers of the log data set.
%LOGDSN	The data set name of the log data set.
%LOGUNIT	The unit type of the log data set.
%LOGVSEQ	The volume sequence number of the log data set.
%LOGVOLS	The volume serial numbers of the log data set.
%LOGFSEQ	The file sequence number of the log data set.
%LOGSEL	Set to YES if any log data sets are selected by the select group; in this case, the delete group following the select group is deleted. Otherwise, the %LOGSEL keyword is set to NO, and a DD DUMMY statement is generated.
%LOGFLRID	The log record sequence number (log record ID) of the first log record in the RLDS.
%LOGLLRID	The log record sequence number (log record ID) of the last log record in the RLDS. If the RLDS has not been closed, %LOGLLRID is set to null.
%RVSYSIN	The Database Recovery utility control statement. Columns in the statement are set as follows: **Column** **Setting** **1** S **4-11** Database name **13-20** Data set or area ddname **22-29** DFSUDUMP **63** V, to indicate this is a RECEIVE All other columns are set to blanks.

Symbolic keywords recognized by the Database Recovery utility - recovery (RECOVJCL)

The Database Recovery utility - recovery recognizes the following symbolic keywords.

Table 13. Symbolic keywords for Database Recovery utility - recovery (RECOVJCL)

Keyword	Description
%DBNAME	The database name of the DBDS to be recovered. %DBNAME is set from the DBD parameter on the GENJCL.RECOV command.
%DBDDN	The ddname of the DBDS; %DBDDN is set from the DDN parameter on the GENJCL.RECOV command.
%DBDSN	The data set name of the DBDS; %DBDSN is set from the DBDS record of the DBDSs.

Table 13. Symbolic keywords for Database Recovery utility - recovery (RECOVJCL) (continued)

Keyword	Description
%DBDSAM	Set to VSAM for a VSAM DBDS. Otherwise, set to null.
%MDBNAME	The HALDB master name, if this is a DBDS of a HALDB partition. This keyword is set to NULL for non-HALDB databases.
%DSLLGTM	The start time for selecting input log data. If an input change accumulation data set is used, %DSLLGTM is set to the volume stop time of the last-accumulated log volume. Otherwise, the keyword value is set to image-copy time.
%SMS	Indicates whether or not an Image Copy 2 image copy data set is being used for the requested utility execution. Set to 1 if yes; otherwise, set to 0.
%ICDSN	The data set name of the image copy data set. Set to null if the USEDBDS parameter is specified on the GENJCL.RECOV command. Otherwise, set from the image copy record for the DBDS.
%ICUNIT	The unit type of the image copy data set. Set to null if the USEDBDS parameter is specified on the GENJCL.RECOV command. Otherwise, set from the image copy record for the DBDS.
%ICVOLS	The volume serial number list of the image copy data set. Set to null if the USEDBDS parameter is specified on the GENJCL.RECOV command. Otherwise, set from the image copy record for the DBDS.
%ICFSEQ	The file sequence number of the image copy data set. Set to null if the USEDBDS parameter is specified on the GENJCL.RECOV command. Otherwise, set from the image copy record for the DBDS.
%CADSN	The data set name of the change accumulation data set. Set to null if no change accumulation is available for the DBDS. Otherwise, set from the change accumulation record.
%CAUNIT	The unit type of the change accumulation data set. Set to null if no change accumulation is available for the DBDS. Otherwise, set from the change accumulation record.
%CAVOLS	The volume serial number list of the change accumulation data set. Set to null if no change accumulation is available for the DBDS. Otherwise, set from the change accumulation record.
%CAFSEQ	The file sequence number of the change accumulation data set. Set to null if no change accumulation is available for the DBDS. Otherwise, set from the change accumulation record.
%LOGDSN	The data set name of the log data set.
%LOGUNIT	The unit type of the log data set.
%LOGVSEQ	The volume sequence number of the log data set.
%LOGVOLS	The volume serial numbers of the log data set.
%LOGFSEQ	The file sequence number of the log data set.
%LOGSEL	Set to YES if any log data sets are selected by the select group; in this case, the delete group following the select group is deleted. Otherwise, the %LOGSEL keyword is set to NO, and a DD DUMMY statement is generated.

Table 13. Symbolic keywords for Database Recovery utility - recovery (RECOVJCL) (continued)

Keyword	Description
%RCSYSIN	The Database Recovery utility control statement. Columns in the statement are set as follows: **Column** **Setting** **1** S **4-11** Database name **13-20** Data set ddname **31-61** The specified time stamp if the RCVTIME parameter was specified on the GENJCL.RECOV command. Otherwise, blank. **63** C, if USEDBDS was specified on the GENJCL.RECOV command. M, if USERIC or LASTUIC were specified on the GENJCL.RECOV command. Otherwise, blank. **64** D, if NODBDSDD was specified on the GENJCL.RECOV command. Otherwise, blank. All other columns are set to blanks.
%RCVFULL	Indicates whether full recoveries are to be generated. When set to YES, full recoveries are generated. If the RCVTIME parameter was specified on the GENJCL.RECOV command, %RCVFULL is set to NO.

IBM-supplied skeletal JCL execution members

This section lists and describes each of the skeletal JCL execution members that are provided by IBM. This skeletal JCL generates executable JCL for running the applicable utilities.

- "The JOB statement"
- "Log Archive utility JCL (ARCHJCL)" on page 259
- "Database Change Accumulation utility JCL (CAJCL)" on page 263
- "Log Recovery utility JCL (LOGCLJCL)" on page 266
- "Database Image Copy utility JCL (ICJCL)" on page 268
- "Online Database Image Copy utility JCL (OICJCL)" on page 273
- "Database Recovery utility JCL-image copy receive-tracking site (ICRCVJCL)" on page 275
- "Database Recovery utility JCL (RECOVJCL)" on page 277
- "HALDB Index/ILDS Rebuild utility JCL (DSPUPJCL)" on page 281

The JOB statement

The IBM-supplied skeletal JCL execution member for the JOB statement is named JOBJCL. JOBJCL is invoked when any GENJCL command is issued.

JOBJCL consists of a single statement, as follows:

```
//JT%TIME JOB
```

You need to modify JOBJCL to add job accounting information that is required by your installation. In addition, you can add JOBLIB, STEPLIB, and JES control statements to JOBJCL. The default job name can be modified. If you use this supplied JOB statement, the job name is generated as JThhmmss, where hhmmss is the time (hour, minute, second) that the JCL was generated.

Log Archive utility JCL (ARCHJCL)

The IBM-supplied skeletal JCL execution member for the Log Archive utility is named ARCHJCL. ARCHJCL is used when the GENJCL.ARCHIVE command is issued.

The following is a listing of ARCHJCL. A description of the statements in ARCHJCL follows it.

```
Note: The following is the OLDS archive EXEC statement.
 %DELETE (%ARCSLDS EQ 'YES')
 //AR%STPNO EXEC PGM=DFSUARC0,PARM='%SSID'
 %ENDDEL
Note: The following is the SLDS archive EXEC statement.
 %DELETE (%ARCSLDS EQ 'NO')
 //AR%STPNO EXEC PGM=DFSUARC0,PARM='DBRC=Y'
 %ENDDEL
 //*
 //*    THIS JCL ORIGINATES FROM THE USER'S 'JCLPDS' LIBRARY.
 //*    KEYWORDS ARE REPLACED BY THE GENJCL FUNCTION OF
 //*    THE IMS/ESA DATABASE RECOVERY CONTROL FEATURE.
 //*
 //*                 JCL FOR ARCHIVE UTILITY
 //*
 //STEPLIB  DD  DSN=IMSVS.RESLIB,DISP=SHR
 //SYSPRINT DD  SYSOUT=A
 %DELETE    (%RCNDSN1 EQ '')
 //RECON1   DD  DSN=%RCNDSN1,DISP=SHR
 %ENDDEL
 %DELETE    (%RCNDSN2 EQ '')
 //RECON2   DD  DSN=%RCNDSN2,DISP=SHR
 %ENDDEL
 %DELETE    (%RCNDSN3 EQ '')
 //RECON3   DD  DSN=%RCNDSN3,DISP=SHR
 %ENDDEL
Note: The following lines are used to archive OLDS.
 %DELETE (%ARCSLDS EQ 'YES')
 %SELECT    OLDS(%SSID,(%ddnames))
 //%OLDSDDN DD  DSN=%OLDSDSN,DISP=SHR
 %ENDSEL
 //DFSSLOGP DD  DSN=IMS.SLDSP.%SSID.D%ARDATE.T%ARTIME.V%ARVERS,
 //             UNIT=3400,VOL=(,,,99),
 //             DISP=(NEW,KEEP),LABEL=(1,SL)
 //DFSSLOGS DD  DSN=IMS.SLDSS.%SSID.D%ARDATE.T%ARTIME.V%ARVERS,
 //             UNIT=3400,VOL=(,,,99),
 //             DISP=(NEW,KEEP),LABEL=(1,SL)
 //RLDSDD1  DD  DSN=IMS.RLDSP.%SSID.D%ARDATE.T%ARTIME.V%ARVERS,
 //             UNIT=3400,VOL=(,,,99),
 //             DISP=(NEW,KEEP),LABEL=(1,SL)
 //RLDSDD2  DD  DSN=IMS.RLDSS.%SSID.D%ARDATE.T%ARTIME.V%ARVERS,
 //             UNIT=3400,VOL=(,,,99),
 //             DISP=(NEW,KEEP),LABEL=(1,SL)
 %ENDDEL
Note: The following lines are used to archive primary SLDSs.
 %DELETE  (%ARCSLDS EQ 'NO')
 %SELECT  SLDS(%SSID,ALL)
 //DFSSLDSP  DD DSN=%SLDSDSN,DISP=(OLD,PASS)
 %ENDSEL
 %ENDDEL
 %DELETE  (%ARCSLDS EQ 'NO' | %SLDSSEL EQ 'NO')
 //DFSSLOGP  DD DSN=IMSVS.ARCH1.%SSID.D%ARDATE.T%ARTIME,
 //             UNIT=3400,VOL=(,,,99),
 //             DISP=(NEW,PASS),LABEL=(1,SL)
 //RLDSDD1   DD DSN=IMSVS.RLDS1.%SSID.D%ARDATE.T%ARTIME,
 //             UNIT=3400,VOL=(,,,99),
 //             DISP=(NEW,PASS),LABEL=(1,SL)
```

```
 %ENDDEL
Note: The following lines are used to archive secondary SLDSs.
 %DELETE  (%ARCSLDS EQ 'NO')
 %SELECT  SSLDS(%SSID,ALL)
 //DFSSLDSS  DD DSN=%SLDSDSN,DISP=(OLD,PASS)
 %ENDSEL
 %ENDDEL
 %DELETE  (%ARCSLDS EQ 'NO' | %SLDSSEL EQ 'NO')
 //DFSSLOGS  DD DSN=IMSVS.ARCH2.%SSID.D%ARDATE.T%ARTIME,
 //             UNIT=3400,VOL=(,,,99),
 //             DISP=(NEW,PASS),LABEL=(1,SL)
 //RLDSDD2   DD DSN=IMSVS.RLDS2.%SSID.D%ARDATE.T%ARTIME,
 //             UNIT=3400,VOL=(,,,99),
 //             DISP=(NEW,PASS),LABEL=(1,SL)
 %ENDDEL
Note: The following lines are common to both processes.
 //SYSIN    DD  *
 SLDS FEOV(08000)
 COPY DDNOUT1(RLDSDD1) DDNOUT2(RLDSDD2) DBRECOV
 /*
Note: The following lines are used for the SLDSs process
 %DELETE  (%ARCSLDS EQ 'NO')
 //*************************************************************
 //*                                                           *
 //* The following optional steps are used to manage the data  *
 //* sets used in the previous SLDS archive step. If the previous *
 //* step completed successfully, the input data sets will be   *
 //* deleted and the output data sets will be cataloged. The   *
 //* output data sets will be deleted if the previous step     *
 //* failed.                                                   *
 //*                                                           *
 //*************************************************************
 //GOODRC%STPNO   EXEC PGM=IEFBR14,COND=(0,NE,AR%STPNO)
 %SELECT  SLDS(%SSID,ALL)
 //PSLDS1   DD DSN=%SLDSDSN,DISP=(OLD,DELETE)
 %ENDSEL
 %SELECT  SSLDS(%SSID,ALL)
 //SSLDS1   DD DSN=%SLDSDSN,DISP=(OLD,DELETE)
 %ENDSEL
 //DD1      DD DSN=*.AR%STPNO.DFSSLOGP,DISP=(OLD,CATLG)
 //DD2      DD DSN=*.AR%STPNO.DFSSLOGS,DISP=(OLD,CATLG)
 //DD3      DD DSN=*.AR%STPNO.RLDSDD1,DISP=(OLD,CATLG)
 //DD4      DD DSN=*.AR%STPNO.RLDSDD2,DISP=(OLD,CATLG)
 /*
 //BADRC%STPNO   EXEC PGM=IEFBR14,COND=(0,EQ,AR%STPNO)
 //DD1      DD DSN=*.AR%STPNO.DFSSLOGP,DISP=(OLD,DELETE)
 //DD2      DD DSN=*.AR%STPNO.DFSSLOGS,DISP=(OLD,DELETE)
 //DD3      DD DSN=*.AR%STPNO.RLDSDD1,DISP=(OLD,DELETE)
 //DD4      DD DSN=*.AR%STPNO.RLDSDD2,DISP=(OLD,DELETE)
 /*
 %ENDDEL
```

You can modify this JCL to suit your needs. It is important to maintain the position of the output DD statements (DFSSLOGP and RLDSDD1) or (DFSSLOGS and RLDSDD2) with respect to the correct %SELECT group. So, the DD statements for the primary output data sets (DFSSLOGP and RLDSDD1) must follow the %SELECT SLDS(%SSID,ALL) select group and precede the %SELECT SSLDS(%SSID,ALL) select group.

Restrictions:

- The %ARVERS keyword is not supported for the SLDS archive process and must not be used.

- Do not use the FREE=CLOSE JCL parameter on the DFSSLOGP DD statement. The data set is dynamically deallocated and using FREE=CLOSE will produce unpredictable results.

EXEC statement
The %STPNO keyword is replaced with the current step number; then the current step number is increased by 1. The %SSID keyword is replaced with the ID of the IMS subsystem that created the OLDSs.

STEPLIB DD statement
DBRC makes no changes to this statement.

SYSPRINT DD statement
DBRC makes no changes to this statement.

RECONn DD statements
The RECON DD statements identify the RECON data sets.

Each of these statements is within a delete group that is controlled by a %RCNDSN keyword. The %RCNDSN keyword values are set from the RECON names that are used when the GENJCL command is executed.

- If RECON data sets are allocated dynamically, the %RCNDSN keywords are set to null, and the RECONn DD statements are deleted.
- If RECON data sets are allocated with JCL, the %RCNDSN keywords are set to the name of the corresponding RECON in the GENJCL command.
- If a RECON data set is not used when the GENJCL command is executed (for example, no spare RECON exists), the keyword is set to null, and the DD statement is deleted.

OLDS DD statements
The DD statements for the OLDSs that are to be archived are generated with a select group. The %SSID keyword identifies the subsystem ID. The %DDNAMES keyword identifies the OLDSs. A DD statement is generated for each specified OLDS. The OLDS ddname replaces the %OLDSDDN keyword. The data set name replaces the %OLDSDSN keyword.

DFSSLOGP DD statement
This DD statement defines the primary SLDS to be created. The subsystem ID replaces the %SSID keyword. The %ARDATE and %ARTIME keywords are replaced with the date (yyddd) and time (hhmmsst) from the open time stamp of the oldest OLDS being archived. The %ARVERS keyword is replaced with the archive version number (nn) of the oldest OLDS being archived.

DFSSLOGS DD statement
This DD statement defines the secondary SLDS that is to be created. The subsystem ID replaces the %SSID keyword. The %ARDATE and %ARTIME keywords are replaced with the date (yyddd) and time (hhmmsst) from the open time stamp of the oldest OLDS that is being archived. The %ARVERS keyword is replaced with the archive version number (nn) of the oldest OLDS that is being archived.

If you are not using dual SLDS logging, delete this DD statement from the skeletal JCL execution member.

RLDSDD1 DD statement
This DD statement defines the primary RLDS that is to be created. The subsystem ID replaces the %SSID keyword. The %ARDATE and %ARTIME keywords are replaced with the date (yyddd) and time (hhmmsst) from the open time

stamp of the oldest OLDS that is being archived. The %ARVERS keyword is replaced with the archive version number (nn) of the oldest OLDS that is being archived.

If you are not using an RLDS, delete this statement and the RLDSDD2 DD statement from the execution member. If this statement is deleted, the utility control COPY statement must be deleted from the SYSIN data. DBRC does not verify that the SYSIN data matches the DD statements.

RLDSDD2 DD statement
This DD statement defines the secondary RLDS that is to be created. The subsystem ID replaces the %SSID keyword. The %ARDATE and %ARTIME keywords are replaced with the date (yyddd) and time (hhmmsst) from the open time stamp of the oldest OLDS that is being archived. The %ARVERS keyword is replaced with the archive version number (nn) of the oldest OLDS being archived.

If you are not using RLDS logging, delete this statement from the execution member. If this statement is deleted, the DDNOUT2(RLDSDD2) parameter must be deleted from the utility control COPY statement in the SYSIN data. DBRC does not verify that the SYSIN data matches the DD statements.

SYSIN DD statement
DBRC makes no changes to the SYSIN DD statement or to the utility control statements in the SYSIN data.

DFSSLDSP DD statements
The DD statements for the primary SLDSs that are to be archived are generated with a select group. The %SSID keyword identifies the subsystem ID. A DD statement is generated for each unarchived SLDS. The SLDS data set name replaces the %SLDSDSN keyword.

DFSSLDSS DD statements
The DD statements for the secondary SLDSs that are to be archived are generated with a select group. The %SSID keyword identifies the subsystem ID. A DD statement is generated for each unarchived SLDS. The SLDS name replaces the %SLDSDSN keyword.

DFSSLOGP DD statement
This DD statement defines the primary SLDS that is to be created. The subsystem ID replaces the %SSID keyword. The %ARDATE and %ARTIME keywords are replaced with the date (yyddd) and time (hhmmsst) from the open time stamp of the oldest OLDS or SLDS that is being archived. The %ARVERS keyword is replaced with the archive version number (nn) of the oldest OLDS being archived.

DFSSLOGS DD statement
This DD statement defines the secondary SLDS that is to be created The subsystem ID replaces the %SSID keyword. The %ARDATE and %ARTIME keywords are replaced with the date (yyddd) and time (hhmmsst) from the open time stamp of the oldest OLDS or SLDS that is being archived. The %ARVERS keyword is replaced with the archive version number (nn) of the oldest OLDS that is being archived.

If you are not using dual SLDS logging, delete these DD statements and the DD2 DD statements from the skeletal JCL execution member.

RLDSDD1 DD statement
This DD statement defines the primary RLDS that is to be created. The subsystem ID replaces the %SSID keyword. The %ARDATE and %ARTIME keywords are replaced with the date (yyddd) and time (hhmmsst) from the open time

stamp of the oldest OLDS or SLDS being archived. The %ARVERS keyword is replaced with the archive version number (nn) of the oldest OLDS being archived.

If you are not using an RLDS, delete these statements, the RLDSDD2 DD statements, and the DD3 and DD4 DD statements from the execution member. If these statements are deleted, the utility control COPY statement must be deleted from the SYSIN data. DBRC does not verify that the SYSIN data matches the DD statements.

RLDSDD2 DD statement

This DD statement defines the secondary RLDS that is to be created. The subsystem ID replaces the %SSID keyword. The %ARDATE and %ARTIME keywords are replaced with the date (yyddd) and time (hhmmsst) from the open time stamp of the oldest OLDS or SLDS that is being archived. The %ARVERS keyword is replaced with the archive version number (nn) of the oldest OLDS that is being archived.

If you are not using dual logging, delete these statements and the DD4 DD statements from the execution member. If these statements are deleted, the DDNOUT2(RLDSDD2) parameter must be deleted from the utility control COPY statement in the SYSIN data. DBRC does not verify that the SYSIN data matches the DD statements.

Database Change Accumulation utility JCL (CAJCL)

The IBM-supplied skeletal JCL execution member for the Database Change Accumulation utility is named CAJCL. CAJCL is used when the GENJCL.CA command is issued. You can specify an execution member other than CAJCL by using the CAJCL parameter on the INIT.CAGRP or CHANGE.CAGRP commands.

The following is a listing of CAJCL. A description of the statements in CAJCL follows it.

```
//CA%STPNO EXEC PGM=DFSUCUM0,PARM='CORE=100000',REGION=800K
//*
//*  THIS JCL ORIGINATES FROM THE USER'S 'JCLPDS' LIBRARY.
//*  KEYWORDS ARE REPLACED BY THE GENJCL FUNCTION OF
//*  THE IMS/ESA DATABASE RECOVERY CONTROL FEATURE.
//*
//*      JCL FOR CHANGE ACCUMULATION
//*
//STEPLIB  DD  DSN=IMS.SDFSRESL,DISP=SHR
//SYSPRINT DD  SYSOUT=A
%DELETE    (%RCNDSN1 EQ '')
//RECON1   DD  DSN=%RCNDSN1,DISP=SHR
%ENDDEL
%DELETE    (%RCNDSN2 EQ '')
//RECON2   DD  DSN=%RCNDSN2,DISP=SHR
%ENDDEL
%DELETE    (%RCNDSN3 EQ '')
//RECON3   DD  DSN=%RCNDSN3,DISP=SHR
%ENDDEL
//IMS      DD  DSN=IMS.DBDLIB,DISP=SHR
//SYSOUT   DD  SYSOUT=A
//SORTLIB  DD  DSN=SYS1.SORTLIB,DISP=SHR
//SORTWK01 DD  UNIT=SYSDA,SPACE=(CYL,(2),,CONTIG)
//SORTWK02 DD  UNIT=SYSDA,SPACE=(CYL,(2),,CONTIG)
//SORTWK03 DD  UNIT=SYSDA,SPACE=(CYL,(2),,CONTIG)
//SORTWK04 DD  UNIT=SYSDA,SPACE=(CYL,(2),,CONTIG)
//SORTWK05 DD  UNIT=SYSDA,SPACE=(CYL,(2),,CONTIG)
//SORTWK06 DD  UNIT=SYSDA,SPACE=(CYL,(2),,CONTIG)
%DELETE    (%CAODSN EQ '')
```

```
//DFSUCUMO DD  DSN=%CAODSN,UNIT=%CAOUNIT,
//             VOL=(PRIVATE,,,,SER=(%CAOVOLS)),
//             LABEL=(%CAOFSEQ,SL),
//             DISP=OLD
%ENDDEL
%DELETE    (%CAODSN NE '')
//DFSUCUMO DD  DUMMY,DCB=BLKSIZE=100
%ENDDEL
//DFSUCUMN DD  DSN=%CANDSN,UNIT=%CANUNIT,
//             VOL=(PRIVATE,,,%CANVCNT,SER=(%CANVOLS)),
//             LABEL=(%CANFSEQ,SL),
//             DISP=(NEW,KEEP)
%SELECT    RLDS((%CAGRP),(FROM(%DSLLGTM)))
//DFSULOG  DD  DSN=%LOGDSN,UNIT=%LOGUNIT,
//             VOL=(PRIVATE,,%LOGVSEQ,,SER=(%LOGVOLS)),
//             LABEL=(%logfseq,SL),
//             DCB=RECFM=VB,
//             DISP=OLD
%ENDSEL
%DELETE    (%LOGSEL EQ 'YES')
//DFSULOG  DD  DUMMY,DCB=BLKSIZE=100
%ENDDEL
//DFSUDD1  DD  DUMMY
//SYSIN    DD  *
%CADB0
/*
```

EXEC statement
The %STPNO keyword is replaced with the current step number; then the current step number is increased by 1.

STEPLIB DD statement
DBRC makes no changes to this statement.

SYSPRINT DD statement
DBRC makes no changes to this statement.

RECONn DD statements
The RECON DD statements identify the RECON data sets.

Each of these statements is within a delete group that is controlled by a %RCNDSN keyword. The %RCNDSN keyword values are set from the RECON names that are used when the GENJCL command is executed.

- If RECON data sets are allocated dynamically, the %RCNDSN keywords are set to null, and the RECONn DD statements are deleted.
- If RECON data sets are allocated with JCL, the %RCNDSN keywords are set to the name of the corresponding RECON in the GENJCL command.
- If a RECON data set is not used when the GENJCL command is executed (for example, no spare RECON exists), the keyword is set to null, and the DD statement is deleted.

IMS DD statement
DBRC makes no changes to this statement.

SYSOUT DD statement
DBRC makes no changes to this statement.

SORTLIB DD statement
DBRC makes no changes to this statement.

SORTWKn DD statements
DBRC makes no changes to these statements.

DFSUCUMO DD statement

This statement identifies a previously created change accumulation data set that is used as input.

Two delete groups are used to generate this DD statement. If no existing change accumulation data set is defined in the RECON data set for the CA group, the value of the %CAODSN keyword is null. Thus, the first delete group is deleted, and the DFSUCUMO DD statement is generated as DUMMY,DCB=BLKSIZE=100.

If an input change accumulation data set is defined in the RECON data set, the %CAODSN keyword is set to the data set name. Thus, the second delete group is deleted, and the DFSUCUMO DD statement identifies the input data set. Other keywords relating to the output data set are replaced as follows:

%CAODSN
Data set name

%CAOUNIT
Unit type

%CAOVOLS
Volume serial number list

%CAOFSEQ
File sequence number

DFSUCUMN DD statement

This DD statement identifies the output change accumulation data set. Other keywords relating to the output data set are replaced as follows:

%CANDSN
Data set name

%CANUNIT
Unit type

%CANVCNT
Volume count

%CANVOLS
Volume serial numbers

%CANFSEQ
File sequence number

DFSULOG DD statement

This DD statement identifies the IMS log data sets that are to be used as input to the Database Change Accumulation utility. A select group selects the required log data sets. %CAGRP identifies the CA group for which log data sets are to be selected. All log volumes that are not previously processed for the CA group are selected. Other keywords for the selected data sets are replaced as follows:

%LOGDSN
Data set name

%LOGUNIT
Unit type

%LOGVSEQ
Volume sequence number

%LOGVOLS
Volume serial numbers

%LOGFSEQ
File sequence numbers

If any log data sets are selected by the select group, the value of the %LOGSEL keyword in the next delete group is YES; this causes the delete group to be deleted. Otherwise, the %LOGSEL keyword is set to NO, and a DD DUMMY statement is generated.

DFSUDD1 DD statement
DBRC makes no changes to this statement.

The DFSUDD1 DD statement identifies the optional output log data set that is produced by the Database Change Accumulation utility. DBRC does not record the optional output log data set; therefore, the skeletal JCL execution member specifies the DFSUDD1 DD statement as DUMMY.

SYSIN DD statement
DBRC makes no changes to this statement.

DB0 Control statements
A DB0 control statement is generated for each DBDS in the CA group.

Log Recovery utility JCL (LOGCLJCL)

The IBM-supplied skeletal JCL execution member for the Log Recovery utility is named LOGCLJCL. LOGCLJCL is used when the GENJCL.CLOSE command is issued.

The following is a listing of LOGCLJCL. A description of the statements in LOGCLJCL follows it.

```
//CL%STPNO        EXEC PGM=DFSULTR0,PARM='IMSID=%SSID'
//*
//*  THIS JCL ORIGINATES FROM THE USER'S 'JCLPDS' LIBRARY.
//*  KEYWORDS ARE REPLACED BY THE GENJCL FUNCTION OF
//*  THE IMS/ESA DATABASE RECOVERY CONTROL FEATURE.
//*
//*               JCL FOR LOG RECOVERY UTILITY
//*
//STEPLIB         DD DSN=IMS.SDFSRESL,DISP=SHR
//SYSPRINT        DD  SYSOUT=A
%DELETE           (%RCNDSN1 EQ '')
//RECON1          DD  DSN=%RCNDSN1,DISP=SHR
%ENDDEL
%DELETE           (%RCNDSN2 EQ '')
//RECON2          DD  DSN=%RCNDSN2,DISP=SHR
%ENDDEL
%DELETE           (%RCNDSN3 EQ '')
//RECON3          DD  DSN=%RCNDSN3,DISP=SHR
%ENDDEL
%SELECT           OLDS(%SSID,(%CDDNAME))
//DFSOL%OLDSTYP  DD  DSN=%OLDSDSN,DISP=SHR
%ENDSEL
%DELETE           (%WADS EQ 'NO')
//DFSWADS0        DD  DSN=IMS.WADS0,DISP=OLD
%ENDDEL
%DELETE           (%WADS EQ 'YES')
%SELECT           OLDS(%SSID,(%NDDNAME))
//DFSNOL%OLDSTYP DD  DSN=%OLDSDSN,DISP=SHR
%ENDSEL
%ENDDEL
%DELETE           (%PDDNAME EQ '')
```

```
%SELECT           OLDS(%SSID, (%PDDNAME))
//DFSPOL%OLDSTYP DD  DSN=%OLDSDSN,DISP=SHR
%ENDSEL
%ENDDEL
//SYSIN           DD  *
CLS
/*
```

EXEC statement
The %STPNO keyword is replaced with the current step number; then the current step number is increased by 1. The %SSID keyword is replaced with the ID of the IMS subsystem that created the OLDS that is to be closed.

STEPLIB DD statement
DBRC makes no changes to this statement.

SYSPRINT DD statement
DBRC makes no changes to this statement.

RECONn DD statements
The RECON DD statements identify the RECON data sets.

Each of these statements is within a delete group that is controlled by a %RCNDSN keyword. The %RCNDSN keyword values are set from the RECON names that are used when the GENJCL command is executed.

- If RECON data sets are allocated dynamically, the %RCNDSN keywords are set to null, and the RECONn DD statements are deleted.
- If RECON data sets are allocated with JCL, the %RCNDSN keywords are set to the name of the corresponding RECON in the GENJCL command.
- If a RECON data set is not used when the GENJCL command is executed (for example, no spare RECON exists), the keyword is set to null, and the DD statement is deleted.

OLDS DD statement
This DD statement identifies the OLDS that is to be closed. A select group is used to select the OLDS. The %SSID keyword identifies the subsystem ID, and the %CDDNAME identifies the OLDS by its DD name. The OLDS type, primary or secondary, replaces the %OLDSTYP keyword. The resulting ddname is DFSOLP or DFSOLS. The %OLDSDSN keyword is replaced with the data set name of the OLDS.

WADS DD statement
This statement is provided only as a model. You must change it before using the skeletal JCL execution member.

The supplied DFSWADS0 DD statement must be replaced with DD statements DFSWADS0 through DFSWADSn. n+1 is the number of WADSs that the online IMS control region uses. The WADS DD statements are contained in a select group that is controlled by the keyword %WADS. The GENJCL.CLOSE command processor sets the value of the %WADS keyword to YES if the OLDS is to be closed using the WADS. The command processor sets the value to NO if the OLDS is to be closed using the next OLDS. The WADS DD statements are, therefore, deleted if the OLDS is to be closed using the next OLDS.

Next OLDS DD statements
If the OLDS is to be closed using the next OLDS, these DD statements identify the next OLDSs. These statements are contained in a delete group that is controlled by the %WADS keyword. Thus, if the OLDS is to be closed using the WADS, these statements are deleted. A select group is used in order to select the next OLDSs. The %SSID keyword identifies the subsystem ID. The %NDDNAME keyword identifies the next OLDS by ddname. The OLDS type, primary or

secondary, replaces the %OLDSTYP keyword. The resulting ddname is DFSNOLP or DFSNOLS. The %OLDSDSN keyword is replaced with the data set name of the OLDS.

Prior OLDS DD statements
If an immediately prior OLDS exists, the corresponding DD statement identifies the immediately prior OLDS. These statements are contained in a delete group that is controlled by the %PDDNAME keyword. If its value is not null, a select group is used in order to select the immediately prior OLDS. The resulting ddname is DFSPOLP or DFSPOLS. Processing of the other keywords is as described under Next OLDS DD statements, above.

SYSIN DD statement
DBRC makes no changes to the SYSIN DD statement or to the utility control statements in the SYSIN data.

Database Image Copy utility JCL (ICJCL)

The IBM-supplied skeletal JCL execution member for the Database Image Copy and Database Image Copy 2 utility is named ICJCL.

ICJCL is used when the GENJCL.IC command is issued. The ICJCL may be processed more than once for each job step in the output JCL. You can specify an execution member other than ICJCL by using the ICJCL parameter on the INIT.DBDS or CHANGE.DBDS commands.

The JCL is divided into three parts:

First part (from the EXEC statement through the IMS DD statement)
Contained in delete groups controlled by keyword %SMSGRP so that they are processed only once per job.

Second part (from the %DBADDN DD statement through the %ICDDN4 DD statement)
Processed once for each DBDS or area to be copied in a job step. The second set of %ICDDNn DD statements is deleted if Sameds was specified on the GENJCL command.

Third part (SYSIN DD statement and the %ICSYSIN statement)
Contained in a delete group controlled by keyword %SMSGRP so that they are processed only for the last (or only) DBDS or area to be copied in a job step.

The following is a listing of ICJCL. A description of the statements in ICJCL follows it.

```
%DELETE  (%SMSGRP NE '
//*-----|  These comments are useful for modifying the skeletal
//*-----|    member, but are confusing when generated in the
//*-----|    output jcl. Force them to be deleted.
//*-----|
//*-----|  New variables for GENJCL.IC:
//*-----|    One DBDS per step:
//*-----|         %SMSGRP = 0 - Not SMS or not Group
//*-----|         %SMS1DS = 0 - Not SMS or not Group or not ONEDS
//*-----|
//*-----|    Multiple DBDSs per step:
//*-----|         %SMSGRP = 1 - SMS + Group + only member
//*-----|         %SMSGRP = 2 - SMS + Group + first member
//*-----|         %SMSGRP = 3 - SMS + Group + not first or last
//*-----|         %SMSGRP = 4 - SMS + Group + last  member
//*-----|         %SMS1DS = 1 - %SMSGRP > 0 + ONEDS
```

```
//*-------------------------------------------------------------------

//*-----|  Section 1 is used only once per step
//*-------------------------------------------------------------------

%ENDDEL
%DELETE  (%SMSGRP GT '2')
//IC%STPNO  EXEC PGM=%PGMIC,REGION=800K,
//             PARM='%PARMX'
//*
//*
//*     THIS JCL ORIGINATES FROM THE USER'S 'JCLPDS' LIBRARY.
//*     KEYWORDS ARE REPLACED BY THE GENJCL FUNCTION OF
//*     THE IMS/ESA DATA BASE RECOVERY CONTROL FEATURE.
//*
//*                JCL FOR IMAGE COPY.
//*
//STEPLIB   DD DSN=IMSVS.RESLIB,DISP=SHR
//SYSPRINT  DD SYSOUT=A
%ENDDEL
%DELETE  (%RCNDSN1 EQ '' | %SMSGRP GT '2')
//RECON1    DD DSN=%RCNDSN1,DISP=SHR
%ENDDEL
%DELETE  (%RCNDSN2 EQ '' | %SMSGRP GT '2')
//RECON2    DD DSN=%RCNDSN2,DISP=SHR
%ENDDEL
%DELETE  (%RCNDSN3 EQ '' | %SMSGRP GT '2')
//RECON3    DD DSN=%RCNDSN3,DISP=SHR
%ENDDEL
%DELETE  (%SMSGRP GT '2')
//IMS       DD DSN=IMS.DBDLIB,DISP=SHR
%ENDDEL
%DELETE  (%SMSGRP NE '')
//*-------------------------------------------------------------------
//*-----|  Single output data set if SMS and ONEDS
//*-------------------------------------------------------------------
%ENDDEL
%DELETE  (%SMSGRP GT '2' | %SMS1DS NE '1')
//%ICDDN1      DD DSN=%ICDSN1,UNIT=%ICUNIT1,
//             VOL=(PRIVATE,,,%ICVCNT1,SER=(%ICVOLS1)),
//             LABEL=(%ICFSEQ1,SL),
//             DISP=(NEW,KEEP)
%ENDDEL
%DELETE  (%COPIES LT '2' | %SMSGRP GT '2' | %SMS1DS NE '1')
//%ICDDN2      DD DSN=%ICDSN2,UNIT=%ICUNIT2,
//             VOL=(PRIVATE,,,%ICVCNT2,SER=(%ICVOLS2)),
//             LABEL=(%ICFSEQ2,SL),
//             DISP=(NEW,KEEP)
%ENDDEL
%DELETE  (%COPIES LT '3' |%SMSGRP GT '2' | %SMS1DS NE '1')
//%ICDDN3      DD DSN=%ICDSN3,UNIT=%ICUNIT3,
//             VOL=(PRIVATE,,,%ICVCNT3,SER=(%ICVOLS3)),
//             LABEL=(%ICFSEQ3,SL),
//             DISP=(NEW,KEEP)
%ENDDEL
%DELETE  (%COPIES LT '4' |%SMSGRP GT '2' | %SMS1DS NE '1')
//%ICDDN4      DD DSN=%ICDSN4,UNIT=%ICUNIT4,
//             VOL=(PRIVATE,,,%ICVCNT4,SER=(%ICVOLS4)),
//             LABEL=(%ICFSEQ4,SL),
//             DISP=(NEW,KEEP)
%ENDDEL
%DELETE  (%SMSGRP NE '')
//*-------------------------------------------------------------------
//*-----|  Section 2 is used once per DBDS (any value of %SMSGRP)
//*-------------------------------------------------------------------

%ENDDEL
```

```
%SELECT  DBDS((%DBNAME,%DBDDN))
%DELETE  (%DBADSAV NE 'AVAIL')
//%DBADDN   DD DSN=%DBDSN,DISP=%CICDISP
%ENDDEL
%DELETE  (%DBADSAV NE '' | %MDBNAME NE'')
//%DBDDN    DD DSN=%DBDSN,DISP=%CICDISP
%ENDDEL
%ENDSEL
%DELETE  (%SMSGRP NE '')
//*-------------------------------------------------------------------
//*-----|  Output data sets not included if SMS + Group + ONEDS
//*-------------------------------------------------------------------
%ENDDEL
%DELETE  (%SMS1DS EQ '1')
//%ICDDN1      DD DSN=%ICDSN1,UNIT=%ICUNIT1,
//             VOL=(PRIVATE,,,%ICVCNT1,SER=(%ICVOLS1)),
//             LABEL=(%ICFSEQ1,SL),
%ENDDEL
%DELETE (%SMS EQ '1' | %SMS1DS EQ '1')
//             DISP=(NEW,KEEP),DCB=BUFNO=10
%ENDDEL
%DELETE (%SMS NE '1' | %SMS1DS EQ '1')
//             DISP=(NEW,KEEP)
%ENDDEL
%DELETE  (%COPIES EQ '1' | %SMS EQ '1' | %SMS1DS EQ '1')
//%ICDDN2      DD DSN=%ICDSN2,UNIT=%ICUNIT2,
//             VOL=(PRIVATE,,,%ICVCNT2,SER=(%ICVOLS2)),
//             LABEL=(%ICFSEQ2,SL),
//             DISP=(NEW,KEEP),DCB=BUFNO=10
%ENDDEL
%DELETE  (%COPIES EQ '1' | %SMS NE '1' | %SMS1DS EQ '1')
//%ICDDN2      DD DSN=%ICDSN2,UNIT=%ICUNIT2,
//             VOL=(PRIVATE,,,%ICVCNT2,SER=(%ICVOLS2)),
//             LABEL=(%ICFSEQ2,SL),
//             DISP=(NEW,KEEP)
%ENDDEL
%DELETE  (%COPIES LE '2' | %SMS EQ '1' | %SMS1DS EQ '1')
//%ICDDN3      DD DSN=%ICDSN3,UNIT=%ICUNIT3,
//             VOL=(PRIVATE,,,%ICVCNT3,SER=(%ICVOLS3)),
//             LABEL=(%ICFSEQ3,SL),
//             DISP=(NEW,KEEP),DCB=BUFNO=10
%ENDDEL
%DELETE  (%COPIES LE '2' | %SMS NE '1' | %SMS1DS EQ '1')
//%ICDDN3      DD DSN=%ICDSN3,UNIT=%ICUNIT3,
//             VOL=(PRIVATE,,,%ICVCNT3,SER=(%ICVOLS3)),
//             LABEL=(%ICFSEQ3,SL),
//             DISP=(NEW,KEEP)
%ENDDEL
%DELETE  (%COPIES LE '3' | %SMS EQ '1' | %SMS1DS EQ '1')
//%ICDDN4      DD DSN=%ICDSN4,UNIT=%ICUNIT4,
//             VOL=(PRIVATE,,,%ICVCNT4,SER=(%ICVOLS4)),
//             LABEL=(%ICFSEQ4,SL),
//             DISP=(NEW,KEEP),DCB=BUFNO=10
%ENDDEL
%DELETE  (%COPIES LE '3' | %SMS NE '1' | %SMS1DS EQ '1')
//%ICDDN4      DD DSN=%ICDSN4,UNIT=%ICUNIT4,
//             VOL=(PRIVATE,,,%ICVCNT4,SER=(%ICVOLS4)),
//             LABEL=(%ICFSEQ4,SL),
//             DISP=(NEW,KEEP)
%ENDDEL
%DELETE  (%SMSGRP NE '')
//*-------------------------------------------------------------------
//*-----|  Section 3 is used once per step (%SMSGRP = 0, 1, or 4
//*-------------------------------------------------------------------
%ENDDEL
%DELETE  (%SMSGRP EQ '2'| %SMSGRP EQ '3')
```

```
//SYSIN     DD *
%ICSYSIN
/*
%ENDDEL
```

- This member may be processed more than once for each job step in the output JCL.
- All statements in the first part of the JCL, from the EXEC statement through the IMS DD statement, are in delete groups controlled by keyword %SMSGRP so that they are processed only once per job step.
- The first set of %ICDDn DD statements identify the single output data set for a group of image copies. It is in a delete group controlled by the %SMS1DS keyword so that it is output only when Sameds was specified in the GENJCL command.
- The second part of the JCL, from the %DBADDN DD statement through the %ICDDN4 DD statement, is processed once for each DBDS or area to be copied in a job step. The %ICDDNn DD statements are deleted if Sameds was specified on the GENJCL command.
- The third part of the JCL, the SYSIN DD statement and the %ICSYSIN statement, are in a delete group controlled by keyword %SMSGRP so that they are processed only for the last (or only) DBDS or area to be copied in a job step.

EXEC statement
The %STPNO keyword is replaced with the current step number; then the current step number is increased by 1.

STEPLIB DD statement
DBRC makes no changes to this statement.

SYSPRINT DD statement
DBRC makes no changes to this statement.

RECONn DD statements
The RECON DD statements identify the RECON data sets.

Each of these statements is within a delete group that is controlled by a %RCNDSN keyword. The %RCNDSN keyword values are set from the RECON names that are used when the GENJCL command is executed.

- If RECON data sets are allocated dynamically, the %RCNDSN keywords are set to null, and the RECONn DD statements are deleted.
- If RECON data sets are allocated with JCL, the %RCNDSN keywords are set to the name of the corresponding RECON in the GENJCL command.
- If a RECON data set is not used when the GENJCL command is executed (for example, no spare RECON exists), the keyword is set to null, and the DD statement is deleted.

IMS DD statement
DBRC makes no changes to this statement.

%DBADDN DD statement
This statement identifies the available ADS that is to be used. The %DBADDN keyword is replaced with the ddname of the ADS. The %DBDSN keyword is replaced with the ADS name.

%DBDNN DD statement
This statement identifies the DBDS that is to be copied. The %DBDDN keyword is replaced with the ddname of the DBDS. The %DBDSN keyword is replaced by the data set name of the DBDS.

%ICDDN1 DD statement

These statements identify the first image copy data set that is produced by an Image Copy utility. Other keywords relating to the image copy data set are replaced as follows:

%ICDSN1
Data set name

%ICVCNT1
Volume count

%ICVOLS1
Volume serial number list

%ICUNIT1
Unit type

%ICFSEQ1
File sequence number

%ICDDN2 | 3 | 4 DD statement

These statements identify the subsequent images that are produced by the Image Copy utility. This DD statement is within a delete group that is controlled by the %COPIES keyword. The %COPIES keyword is set to 1 if a single image copy data set is to be produced or to a 2, 3, or 4 if multiple image copy data sets are to be produced. If %COPIES is 1, the group is deleted.

The %ICDSNx, %ICVCNTx, %ICVOLSx, %ICUNITx, and %ICFSEQx keywords are replaced with the same type of information as is shown under the %D1OU0 DD statement just preceding. x can be either 2, 3 or 4.

SYSIN DD statement

DBRC makes no changes to this statement.

%ICSYSIN statement

The Image Copy utility control statement replaces the %ICSYSIN keyword.

The %ICSYSIN statement is required. If the %ICSYSIN statement is deleted, the GENJCL.IC command fails. Multiple control statements are generated if you are processing a group.

The following is sample JCL to use if your installation uses IMSPLEX and DBRCGRP EXEC parameters to identify the IMSplex and DBRC group that the DBRC instance should register with. This JCL enables the same skeletal member to be used for DFSUDMP0 and DFSUDMT0.

```
%DELETE  (%ICRTYPE NE 'BATCH' & %ICRTYPE NE 'CIC')
//IC%STPNO EXEC PGM=%PGMIC,REGION=0M,
//    PARM='%PARMX,IMSPLEX=%IMSPLEX,DBRCGRP=%DBRCGRP'
%ENDDEL
%DELETE  (%SMSGRP GT '2' | %ICRTYPE EQ 'BATCH' | %ICRTYPE EQ 'CIC')
//IC%STPNO EXEC PGM=%PGMIC,REGION=0M,
//    PARM=(%PARMX,,,,,                                              C
//    'IMSPLEX=%IMSPLEX,DBRCGRP=%DBRCGRP')
%ENDDEL
%DELETE  (%SMSGRP GT '2')
//*
//*
//*     THIS JCL ORIGINATES FROM THE USER'S 'JCLPDS' LIBRARY.
//*     KEYWORDS ARE REPLACED BY THE GENJCL FUNCTION OF
//*     THE IMS/ESA DATA BASE RECOVERY CONTROL FEATURE.
//*
//*                JCL FOR IMAGE COPY.
```

```
//*
//STEPLIB   DD DSN=IMSVS.RESLIB,DISP=SHR
//SYSPRINT  DD SYSOUT=A
%ENDDEL
```

Online Database Image Copy utility JCL (OICJCL)

The IBM-supplied skeletal JCL execution member for the Online Image Copy utility is named OICJCL. OICJCL is used when the GENJCL.OIC command is issued. You can specify an execution member other than OICJCL by using the OICJCL parameter on the INIT.DBDS or CHANGE.DBDS commands.

The following is a listing of OICJCL. A description of the statements in OICJCL follows it.

```
 //OIC%STPNO EXEC PGM=DFSRRC00,PARM='BMP,DFSUICP0,%PSB,,MASTER',
//             REGION=700K
//*     THIS JCL ORIGINATES FROM THE USER'S 'JCLPDS' LIBRARY.
//*     KEYWORDS ARE REPLACED BY THE GENJCL FUNCTION OF
//*     THE IMS/VS DATA BASE RECOVERY CONTROL FEATURE.
//*
//*           JCL FOR ONLINE IMAGE COPY.
//*
//STEPLIB   DD DSN=IMSVS.SDFSRESL,DISP=SHR
//SYSPRINT  DD SYSOUT=A
%DELETE  (%RCNDSN1 EQ '')
//RECON1    DD DSN=%RCNDSN1,DISP=SHR
%ENDDEL
%DELETE  (%RCNDSN2 EQ '')
//RECON2    DD DSN=%RCNDSN2,DISP=SHR
%ENDDEL
%DELETE  (%RCNDSN3 EQ '')
//RECON3    DD DSN=%RCNDSN3,DISP=SHR
%ENDDEL
//IMS       DD DSN=IMS.DBDLIB,DISP=SHR
//%ICDDN1      DD DSN=%ICDSN1,UNIT=%ICUNIT1,
//             VOL=(PRIVATE,,,%ICVCNT1,SER=(%ICVOLS1)),
//             LABEL=(%ICFSEQ1,SL),
//             DISP=(NEW,KEEP)
%DELETE  (%COPIES EQ '1')
//%ICDDN2      DD DSN=%ICDSN2,UNIT=%ICUNIT2,
//             VOL=(PRIVATE,,,%ICVCNT2,SER=(%ICVOLS2)),
//             LABEL=(%ICFSEQ2,SL),
//             DISP=(NEW,KEEP)
%ENDDEL
//DFSUCKPT  DD DSN=IMSVS.%DBNAME.%DBDDN.CHECKPT.IC%TIME,
//             UNIT=SYSDA,SPACE=(TRK,1),DISP=(NEW,CATLG)
//SYSIN     DD *
%ICSYSIN
/*
```

EXEC statement
The %STPNO keyword is replaced with the current step number; then the current step number is increased by 1. The PSB name that is specified on the GENJCL.OIC command replaces the %PSB keyword.

STEPLIB DD statement
DBRC makes no changes to this statement.

SYSPRINT DD statement
DBRC makes no changes to this statement.

RECONn DD statements
The RECON DD statements identify the RECON data sets.

Each of these statements is within a delete group that is controlled by a %RCNDSN keyword. The %RCNDSN keyword values are set from the RECON names that are used when the GENJCL command is executed.

- If RECON data sets are allocated dynamically, the %RCNDSN keywords are set to null, and the RECONn DD statements are deleted.
- If RECON data sets are allocated with JCL, the %RCNDSN keywords are set to the name of the corresponding RECON in the GENJCL command.
- If a RECON data set is not used when the GENJCL command is executed (for example, no spare RECON exists), the keyword is set to null, and the DD statement is deleted.

IMS DD statement
DBRC makes no changes to this statement.

%ICDDN1 DD statement
This statement identifies the first image copy data set that is produced by the Image Copy utility. Other keywords relating to the online image copy data set are replaced as follows:

%ICDSN1
Data set name

%ICVCNT1
Volume count

%ICVOLS1
Volume serial number list

%ICUNIT1
Unit type

%ICFSEQ1
File sequence number

%ICDDN2 DD statement
This statement identifies the duplicate image copy data set that produced by the Image Copy utility. This DD statement is within a delete group controlled by the %COPIES keyword. The %COPIES keyword is set to 1 if a single image copy data set is to be produced or to 2 if duplicate image copy data sets are to be produced. If %COPIES is 1, the group is deleted.

The %ICDSN2, %ICVCNT2, %ICVOLS2, %ICUNIT2, and %ICFSEQ1 keywords are replaced with the same type of information as is shown under the %D1OU0 DD statement just preceding.

DFSUCKPT DD statement
The DFSUCKPT DD statement identifies the optional online image copy checkpoint data set. Keywords relating to this optional data set are replaced as follows:

%DBNAME
The database name

%DBDDN The ddname

%TIME The current time of day (in the form hhmmss)

The volume serial number and device type for the checkpoint data set are not specified in the IBM-supplied skeletal JCL. You must supply these if checkpoint data sets are to be used.

The DFSUCKPT DD statement is optional. If checkpoint data sets are not to be used by the Online Image Copy utility, the statement can be deleted.

SYSIN DD statement
DBRC makes no changes to this statement.

%ICSYSIN statement
The Image Copy utility control statement replaces the %ICSYSIN keyword.

Database Recovery utility JCL-image copy receive-tracking site (ICRCVJCL)

The IBM-supplied skeletal JCL execution member for the Database Recovery utility (as used at the tracking site to receive the image copy) is named ICRCVJCL. ICRCVJCL is used when the GENJCL.RECEIVE command is issued. You can specify an execution member other than ICRCVJCL by using the RECVJCL parameter on the INIT.DBDS or CHANGE.DBDS commands.

The following is a listing of ICRCVJCL. A description of the statements in ICRCVJCL follows it.

```
//RCV%STPNO EXEC PGM=DFSRRC00,REGION=1300K,
//             PARM='UDR,DFSURDB0,%DBNAME',,,,,,,,,,,,,,,,,,,,,%GSGNAME'
//*
//*     THIS JCL ORIGINATES FROM THE USER'S 'JCLPDS' LIBRARY.
//*     KEYWORDS ARE REPLACED BY THE GENJCL FUNCTION OF
//*     THE IMS/VS DATA BASE RECOVERY CONTROL FEATURE.
//*
//*               JCL FOR IMAGE COPY RECEIVE
//*
//STEPLIB   DD DSN=IMSVS.RESLIB,DISP=SHR
//SYSPRINT  DD SYSOUT=A
%DELETE  (%RCNDSN1 EQ '')
//RECON1    DD DSN=%RCNDSN1,DISP=SHR
%ENDDEL
%DELETE  (%RCNDSN2 EQ '')
//RECON2    DD DSN=%RCNDSN2,DISP=SHR
%ENDDEL
%DELETE  (%RCNDSN3 EQ '')
//RECON3    DD DSN=%RCNDSN3,DISP=SHR
%ENDDEL
//IMS       DD DSN=IMS.DBDLIB,DISP=SHR
//%DBDDN    DD DSN=%DBDSN,
%DELETE  (%DBDSAM EQ 'VSAM')
//             UNIT=SYSDA,
//             VOL=SER=VOLSER,
//             SPACE=(CYL,(20,2)),
//             DISP=(NEW,KEEP),
//             DCB=BUFNO=10
%ENDDEL
%DELETE  (%DBDSAM NE 'VSAM')
//             DISP=OLD
%ENDDEL
//DFSUDUMP  DD DSN=%ICDSN,UNIT=%ICUNIT,
//             VOL=(PRIVATE,,,,SER=(%ICVOLS)),
//             LABEL=(%ICFSEQ,SL),
//             DISP=(OLD,KEEP),DCB=BUFNO=10
%DELETE    (%LOGSEL EQ 'YES')
//DFSULOG  DD  DUMMY
%ENDDEL
%DELETE    (%CADSN NE '')
//DFSUCUM  DD  DUMMY
%ENDDEL
//DFSVSAMP  DD *
1024,4
```

```
4096,4
//SYSIN     DD *
%RVSYSIN
/*
```

EXEC statement
The %STPNO keyword is replaced with the current step number; then the current step number is increased by 1. The %DBNAME keyword is replaced with the database name of the DBDS or area that is being received.

STEPLIB DD statement
DBRC makes no changes to this statement.

SYSPRINT DD statement
DBRC makes no changes to this statement.

RECONn DD statements
The RECON DD statements identify the RECON data sets.

Each of these statements is within a delete group that is controlled by a %RCNDSN keyword. The %RCNDSN keyword values are set from the RECON names that are used when the GENJCL command is executed.

- If RECON data sets are allocated dynamically, the %RCNDSN keywords are set to null, and the RECONn DD statements are deleted.
- If RECON data sets are allocated with JCL, the %RCNDSN keywords are set to the name of the corresponding RECON in the GENJCL command.
- If a RECON data set is not used when the GENJCL command is executed (for example, no spare RECON exists), the keyword is set to null, and the DD statement is deleted.

IMS DD statement
DBRC makes no changes to this statement.

%DBDDN DD statement
The %DBDDN keyword is replaced by the ddname of the DBDS that is being received. The %DBDSN keyword is replaced by the data set name of the DBDS or area.

Delete groups control the remainder of the %DBDDN DD statement. The access method of the DBDS controls the content of the delete groups. If the access method is VSAM, DISP=OLD is generated. Otherwise, the UNIT, VOL, SPACE, DISP, and DCB parameters are generated.

DFSUDUMP DD statement
This DD statement identifies the image copy data set that is to be received. The %ICDSN, %ICUNIT, %ICVOLS, and %ICFSEQ keywords are set from the appropriate fields in the image copy RECON record.

DFSUCUM DD statement
This DD statement is always listed as DUMMY at the tracking site.

DFSULOG DD statement
This DD statement is always listed as DUMMY at the tracking site.

DFSVSAMP DD statement
The DFSVSAMP DD statement identifies information that is required by the DL/I buffer handler. DBRC makes no changes to these statements.

SYSIN DD statement
This DD statement contains database recovery statements that control the processing.

%RVSYSIN statement

DBRC replaces the %RVSYSIN keyword.

Database Recovery utility JCL (RECOVJCL)

The IBM-supplied skeletal JCL execution member for the Database Recovery utility is named RECOVJCL. RECOVJCL is used when the GENJCL.RECOV command is issued. You can specify an execution member other than RECOVJCL by using the RECOVJCL parameter on the INIT.DBDS or CHANGE.DBDS commands.

The following is a listing of RECOVJCL. A description of the statements in RECOVJCL follows it.

```
%DELETE   (%MDBNAME NE '')
//RCV%STPNO EXEC PGM=DFSRRC00,REGION=1300K,
//             PARM='UDR,DFSURDB0,%DBNAME,,,,,,,,,,,,,,,,,,,%GSGNAME'
%ENDDEL
%DELETE   (%MDBNAME EQ  '')
//RCV%STPNO EXEC PGM=DFSRRC00,REGION=1300K,
//            PARM='UDR,DFSURDB0,
%ENDDEL
//*
//*
//*      THIS JCL ORIGINIATES FROM THE USER'S 'JCLPDS' LIBRARY.
//*      KEYWORDS ARE REPLACED BY THE GENJCL FUNCTION OF
//*      THE IMS/VS DATA BASE RECOVERY CONTROL FEATURE.
//*
//*              JCL FOR RECOVERY.
//*
//STEPLIB   DD DSN=IMSVS.SDFSRESL,DISP=SHR
//SYSPRINT  DD SYSOUT=A
%DELETE  (%RCNDSN1 EQ '')
//RECON1    DD DSN=%RCNDSN1,DISP=SHR
%ENDDEL
%DELETE  (%RCNDSN2 EQ '')
//RECON2    DD DSN=%RCNDSN2,DISP=SHR
%ENDDEL
%DELETE  (%RCNDSN3 EQ '')
//RECON3    DD DSN=%RCNDSN3,DISP=SHR
%ENDDEL
//IMS       DD DSN=IMS.DBDLIB,DISP=SHR
//%DBDDN    DD DSN=%DBDSN,
%DELETE  (%DBDSAM EQ 'VSAM' | %SMS EQ '1')
//             UNIT=SYSDA,
//             VOL=SER=VOLSER,
//             SPACE=(CYL,(20,2)),
//             DISP=(NEW,KEEP),
//             DCB=BUFNO=10
%ENDDEL
%DELETE  (%DBDSAM EQ 'VSAM' | %SMS EQ '0')
//             UNIT=SYSDA,
//             VOL=SER=VOLSER,
//             SPACE=(CYL,(20,2)),
//             DISP=(NEW,KEEP)
%ENDDEL
%DELETE  (%DBDSAM NE 'VSAM')
//             DISP=OLD
%ENDDEL
%DELETE  (%ICDSN EQ '')
//DFSUDUMP  DD DSN=%ICDSN,
%ENDDEL
%DELETE  (%ICCAT EQ 'YES')
//             UNIT=%ICUNIT,
//             VOL=(PRIVATE,,,,SER=(%ICVOLS)),
//             LABEL=(%ICFSEQ,SL),
```

```
%ENDDEL
%DELETE  (%ICDSN EQ '' | %SMS EQ '1')
//             DISP=(OLD,KEEP),DCB=BUFNO=10
%ENDDEL
%DELETE  (%ICDSN EQ '' | %SMS EQ '0')
//             DISP=(OLD,KEEP)
%ENDDEL
%DELETE  (%ICDSN NE '')
//DFSUDUMP  DD DUMMY
%ENDDEL
//DFSVDUMP  DD DUMMY
%DELETE  (%CADSN EQ '')
//DFSUCUM   DD DSN=%CADSN,UNIT=%CAUNIT,
//             VOL=(PRIVATE,,,,SER=(%CAVOLS)),
//             LABEL=(%CAFSEQ,SL),
//             DISP=(OLD,KEEP),DCB=BUFNO=10
%ENDDEL
%DELETE  (%CADSN NE '')
//DFSUCUM   DD DUMMY
%ENDDEL
%SELECT  RLDS((%DBNAME,%DBDDN),FROM(%DSLLGTM))
%DELETE (%LOGVOLS EQ '')
//DFSULOG   DD DSN=%LOGDSN,UNIT=%LOGUNIT,
//             VOL=(PRIVATE,,%LOGVSEQ,,SER=(%LOGVOLS)),
//             LABEL=(%LOGFSEQ,SL),
//             DCB=RECFM=VB,
//             DISP=OLD
%ENDDEL
%DELETE (%LOGVOLS NE '')
//DFSULOG   DD DSN=%LOGDSN,DISP=OLD
%ENDDEL
%ENDSEL
%DELETE  (%LOGSEL EQ 'YES')
//DFSULOG   DD DUMMY
%ENDDEL
%DELETE  (%TRACK EQ 'NO')
//DFSTRCV   DD DSN=??????,
//             DISP=OLD
%ENDDEL
//DFSVSAMP  DD *
1024,4
4096,4
//SYSIN     DD *
%RCSYSIN
/*
```

EXEC statement
The %STPNO keyword is replaced with the current step number; then the current step number is increased by 1. For HALDB, the HALDB master name, %MDBNAME, of the DBDs being recovered is used in the EXEC statement. For all others, the database name, %DBNAME, is used.

STEPLIB DD statement
DBRC makes no changes to this statement.

SYSPRINT DD statement
DBRC makes no changes to this statement.

RECONn DD statements
The RECON DD statements identify the RECON data sets.

Each of these statements is within a delete group that is controlled by a %RCNDSN keyword. The %RCNDSN keyword values are set from the RECON names that are used when the GENJCL command is executed.

- If RECON data sets are allocated dynamically, the %RCNDSN keywords are set to null, and the RECONn DD statements are deleted.

- If RECON data sets are allocated with JCL, the %RCNDSN keywords are set to the name of the corresponding RECON in the GENJCL command.
- If a RECON data set is not used when the GENJCL command is executed (for example, no spare RECON exists), the keyword is set to null, and the DD statement is deleted.

IMS DD statement
DBRC makes no changes to this statement.

%DBDDN DD statement
The %DBDDN keyword is replaced by the ddname of the DBDS that is being recovered. The %DBDSN keyword is replaced by the data set name of the DBDS.

Delete groups control the remainder of the %DBDDN DD statement. The access method of the DBDS controls the content of the delete groups. If the access method is VSAM, DISP=OLD is generated. Otherwise, the UNIT, VOL, SPACE, DISP, and DCB parameters are generated.

DFSUDUMP DD statement
This DD statement identifies the image copy data set, if any, that is to be used for recovery. Delete groups, which are controlled by the %ICDSN keyword, are used to generate this DD statement.

If the USEIC parameter was specified or if it was the default, on the GENJCL.RECOV command, the %ICDSN keyword is set to its data set name. Thus, the first delete group for DFSUDUMP is used, and the second delete group is deleted. Other keywords within the first delete group are unchanged.

If the USEDBDS or USEAREA keyword was specified on the GENJCL.RECOV command, the DFSUDUMP DD statement is generated as DUMMY.

DFSVDUMP DD statement
The DFSVDUMP DD statement is always generated as DUMMY.

DFSUCUM DD statement
This DD statement identifies the change accumulation data set, if any, to be used as input to recovery. Delete groups, which are controlled by the %CADSN keyword, are used to generate the DFSUCUM DD statement. If the DBDS belongs to a CA group, the %CADSN keyword is set to the data set name of the most-recent change accumulation data set. If the DBDS does not belong to a CA group or if no usable change accumulation data set exists, the %CADSN keyword is set to null.

- If the %CADSN keyword is null, the DFSUCUM DD statement is generated as DUMMY.
- If the %CADSN keyword is not null, the DFSUCUM DD statement identifies the change accumulation data set.

Other keywords relating to the change accumulation data set are replaced as follows:

%CAVOLS
Volume serial number list

%CAUNIT
Unit type

%CAFSEQ
File sequence number

DFSULOG DD statement
This DD statement identifies the log data sets that are to be used as input to the Database Recovery utility.

A select group selects the required log data sets. The %DBNAME and %DBDDN keywords identify the DBDS for which log data sets are to be selected. All log volumes that contain change records for the DBDS, that are not included in the change accumulation data sets, are selected. Other keywords relating to the log data sets are replaced as follows:

%LOGDSN
Log data set name

%LOGUNIT
Unit type

%LOGVSEQ
Volume sequence number

%LOGVOLS
Volume serial numbers

%LOGFSEQ
File sequence number

If any log data sets are selected, the value of the %LOGSEL keyword is YES, and the following delete group is deleted. Otherwise, the %LOGSEL keyword is NO, and a DD DUMMY statement is generated.

DFSTRCV DD statement
The DFSTRCV DD statement identifies the DBDS for which one or more tracks is being recovered. If the TRACK parameter was not specified on the GENJCL command, this statement does not appear in the generated JCL.

You must modify the DFSTRCV DD statement to include in it the appropriate data set name and unit information. You can modify it in either the skeletal JCL or generated JCL.

DFSVSAMP DD statement
The DFSVSAMP DD statement identifies information required by the DL/I buffer handler. DBRC makes no changes to these statements.

SYSIN DD statement
This DD statement contains database recovery statements that control the processing.

%RCSYSIN statement
DBRC replaces the %RCSYSIN keyword.

%RCVFULL
The %RCVFULL keyword indicates what type of recovery is being generated. It is set to NO when the RCVTIME parameter (time-stamp recovery) is specified on the GENJCL.RECOV command. It is set to YES to indicate full recoveries.

This keyword is useful if, for example, you want to turn ON the IC-NEEDED flag in the DBDS record following a time-stamp recovery. You could add the following JCL to the end of your RECOVJCL skeletal JCL member to accomplish this.

```
%DELETE (%RCVFULL EQ 'YES')
//RCV%STPNO EXEC PGM=DSPURX00
//STEPLIB   DD  DSN=IMS.SDFSRESL,DISP=SHR
%ENDDEL
%DELETE (%RCVFULL EQ 'YES' | %RCNDSN1 EQ '')
//RECON1    DD  DSN=%RCNDSN1,DISP=SHR
%ENDDEL
%DELETE (%RCVFULL EQ 'YES' | %RCNDSN2 EQ '')
//RECON2    DD  DSN=%RCNDSN2,DISP=SHR
%ENDDEL
```

```
%DELETE (%RCVFULL EQ 'YES' | %RCNDSN3 EQ '')
//RECON3    DD  DSN=%RCNDSN3,DISP=SHR
%ENDDEL
%DELETE (%RCVFULL EQ 'YES')
//SYSIN     DD *
CHANGE.DBDS DBD(%DBNAME) DDN(%DBDDN) ICON
/*
%ENDDEL
```

HALDB Index/ILDS Rebuild utility JCL (DSPUPJCL)

DFSPREC0 is used to rebuild the ILDS or primary index data sets (or both) of a HALDB partition.

The Database Recovery utility and the image copy utilities cannot be used on ILDS and primary index data sets. The GENJCL.RECOV, GENJCL.RECEIVE, GENJCL.IC, and GENJCL.OIC commands fail when attempted specifically for ILDS and primary index data sets. GENJCL commands on groups, either explicit (using the GROUP keyword) or implicit (DBD with no DDN keyword), do not generate any JCL for ILDS or primary index data sets. These data sets are ignored. GENJCL.USER can specify ILDS and primary index data sets. The ILDS and primary index data sets are not ignored for this command.

After a time-stamp recovery of a HALDB partition (the data DBDSs of the HALDB partition), the applicable ILDS or primary index data sets (or both) must be rebuilt. There is no specific GENJCL support for these data sets, but the GENJCL.USER command can be used. The IBM-supplied skeletal JCL execution member is named DSPUPJCL. Here is a suggestion for implementation:

```
GENJCL.USER MEMBER (DSPUPJCL) -
            USERKEYS ((%MDBNAME, 'DBHDOJ01'),(%DBNAME,'PART1'), -
                      (%RCVTYP, 'ILE'))
```

The following is a listing of DSPUPJCL. A description of the statements in DSPUPJCL follows it.

```
//UPREC%STPNO EXEC PGM=DFSRRC00,REGION=1300K,
//             PARM='ULU,DFSPREC0,%MDBNAME,,,,,,,,,,,,,,,,,,Y,N'
//*
//*********************************************************************
//*
//*     THIS JCL ORIGINATES FROM THE USER'S 'JCLPDS' LIBRARY.
//*     KEYWORDS ARE REPLACED BY THE GENJCL FUNCTION.
//*
//*              User JCL for rebuilding either the index or ILDS data set, or
//*              BOTH data sets for a HALDB partition.
//*
//SYSPRINT  DD SYSOUT=A
%DELETE  (%RCNDSN1 EQ '')
//RECON1    DD DSN=%RCNDSN1,DISP=SHR
%ENDDEL
%DELETE  (%RCNDSN2 EQ '')
//RECON2    DD DSN=%RCNDSN2,DISP=SHR
%ENDDEL
%DELETE  (%RCNDSN3 EQ '')
//RECON3    DD DSN=%RCNDSN3,DISP=SHR
%ENDDEL
//IMS       DD DSN=IMS.DBDLIB,DISP=SHR
//DFSVSAMP  DD DSN=IMS.VSAM.PARM(OPTIONS),DISP=SHR
//SYSIN     DD *
PARTITION=%DBNAME,RECOVTYP=%RCVTYP
/*
```

EXEC statement
The %STPNO keyword is replaced with the current step number; then the current step number is increased by 1. The %MDBNAME keyword is replaced with the HALDB master name of the DBDS that is being recovered.

SYSPRINT DD statement
DBRC makes no changes to this statement.

RECONn DD statements
The RECON DD statements identify the RECON data sets.

Each of these statements is within a delete group that is controlled by a %RCNDSN keyword. The %RCNDSN keyword values are set from the RECON names that are used when the GENJCL command is executed.

- If RECON data sets are allocated dynamically, the %RCNDSN keywords are set to null, and the RECONn DD statements are deleted.
- If RECON data sets are allocated with JCL, the %RCNDSN keywords are set to the name of the corresponding RECON in the GENJCL command.
- If a RECON data set is not used when the GENJCL command is executed (for example, no spare RECON exists), the keyword is set to null, and the DD statement is deleted.

IMS DD statement
DBRC makes no changes to this statement.

DFSVSAMP DD statement
The DFSVSAMP DD statement identifies information required by the DL/I buffer handler. DBRC makes no changes to these statements.

SYSIN DD statement
This DD statement contains database recovery statements that control the processing.

PARTITION=%DBNAME
This statement indicates the HALDB partition name of the ILDS or primary index data sets being rebuilt.

RECOVTYP=%RCVTYP
This statement indicates which type of data sets (ILDS, primary index, or both) the DFSPREC0 utility is rebuilding. It also indicates whether the utility is rebuilding the ILDS in VSAM load mode (to include free space) by providing the free space option parameter (ILEF or BOTHF).

Related reference:

Database Change Accumulation utility (DFSUCUM0) (Database Utilities)

Chapter 14. INIT commands

Use the INIT command to create various records in the RECON data set.

INIT.ADS command

Use the INIT.ADS command to create an entry in the RECON data set that defines an ADS (area data set) for an area. You can also use the command to define the data sets needed by the DEDB Alter utility. An area can include up to seven data sets and seven shadow data sets (for DEDB Alter).

For the DEDB Alter utility's ALTERAREA function, the INIT.ADS command is used with the SHADOW and IC options to define 1 to 6 shadow area data sets and 1 shadow image copy data set. The maximum total shadow data sets (shadow area data sets and shadow image copy data sets) is 7.

For the DEDB Alter utility's ADDAREA function, the INIT.ADS command is used without the SHADOW and IC options to define 1 to 7 active area data sets for shadow areas.

An area being added by the ADDAREA function is registered as a shadow area. The area data sets for the shadow area are registered as standard (active) area data stets. Shadow area data sets might be registered for the shadow area, but the ADDAREA function uses the active area data sets and ignores the shadow area data sets.

Before you issue the INIT.ADS command, you must create the area and database records in the RECON data set. Use the INIT.DBDS command to register the area.

Note: If the ADDN or ADSN names are not unique for this area, the INIT.ADS command fails.

When you register the area, the area status is set as “recovery needed” to prevent inadvertent use by the IMS online system before you have completed registration of the required ADSs.

While an area has a recovery needed status, you can create the ADS records in the RECON data set, but you must create them as unavailable by issuing the command INIT.ADS UNAVAIL. Shadow area data sets can be defined at any time, as long as the DEDB Alter utility is not currently running for the identified area.

However, you can create an ADS that is immediately available if you first remove the recovery needed status from the area with the CHANGE.DBDS NORECOV command and then create the ADS by issuing the command INIT.ADS AVAIL.

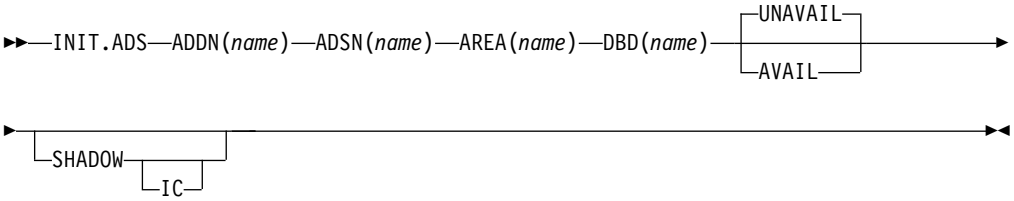
A separate INIT.ADS command must be issued for each area data set that is registered for the area.

Subsections:

- “Syntax” on page 284
- “Keywords” on page 284
- “Examples” on page 284

© Copyright IBM Corp. 1974, 2017

Syntax



Keywords

ADDN(*name*)
Required keyword you use to identify the ADS that is being identified to DBRC by its ddname.

ADSN(*name*)
Required keyword you use to identify the ADS that is being identified to DBRC by its data set name.

AREA(*name*)
Required keyword you use to identify the area name for which an ADS is being identified to DBRC.

DBD(*name*)
Required keyword you use to identify that area by database name for which an ADS is being identified to DBRC.

UNAVAIL | AVAIL
Mutually exclusive, optional keywords that you use to indicate whether the ADS record is available.

UNAVAIL
Makes the ADS status unavailable.

AVAIL
Makes the ADS status available. The INIT.ADS AVAIL command fails if you issue it when the area is in use or if the area needs to be recovered.

SHADOW
Optional keyword used to register 1-7 active area data sets in the RECON data set, to be used for the DEDB Alter utility.

IC Optional keyword that specifies that a shadow area data set is to be defined as a shadow image copy data set. This keyword is valid only when the SHADOW keyword is also specified.

Recommendation: Increasing the image copy data set GENMAX value by one or two before running the DEDB Alter utility can help accommodate the user image copy data set. You can decrease the image copy data set GENMAX value to restore it to its original value at a later time.

Examples

Here are some examples of using the INIT.ADS command.

Examples of creating a record that defines an ADS

In the following examples, a record that identifies an ADS is created in the RECON data set.

```
 //INITADS  JOB

⋮
 //SYSIN    DD   *
      INIT.ADS DBD(DBD03) AREA(AREA03)  -
               ADDN(AREADDN1) ADSN(AREADSN2)
 /*
```

Example of creating a record that defines MADSs

In this example, a record is created in the RECON data set that identifies multiple area data sets (MADSs) that are associated with the same DEDB area (DB21AR0).

```
INIT.DB DBD(DEDBJN21) SHARELVL(1) TYPEFP
INIT.DBDS DBD(DEDBJN21) AREA(DB21AR0) GENMAX(2) VSO PRELOAD
INIT.ADS DBD(DEDBJN21) AREA(DB21AR0) ADSN(IMSTESTL.DB21AR01) -
           ADDN(DB21AR01)
INIT.ADS DBD(DEDBJN21) AREA(DB21AR0) ADSN(IMSTESTL.DB21AR02) -
           ADDN(DB21AR02)
INIT.ADS DBD(DEDBJN21) AREA(DB21AR0) ADSN(IMSTESTL.DB21AR03) -
           ADDN(DB21AR03)
```

Examples of defining shadow area data sets

In the following example, records are created in the RECON data set that identify the shadow ADS and shadow image copy ADS that are used by the ALTRAREA function of the DEDB Alter utility.

```
//INITADS  JOB

 //SYSIN    DD   *
      INIT.ADS DBD(DEDBJ001) AREA(D0010001) ADDN(D001001S) -
               ADSN(IMSTESTS.D0010001.SHADOW) SHADOW
      INIT.ADS DBD(DEDBJ001) AREA(D0010001) ADDN(D001001I) -
               ADSN(IMSTESTS.D0010001.SHADOW.IC) SHADOW IC
 /*
```

INIT.CA command

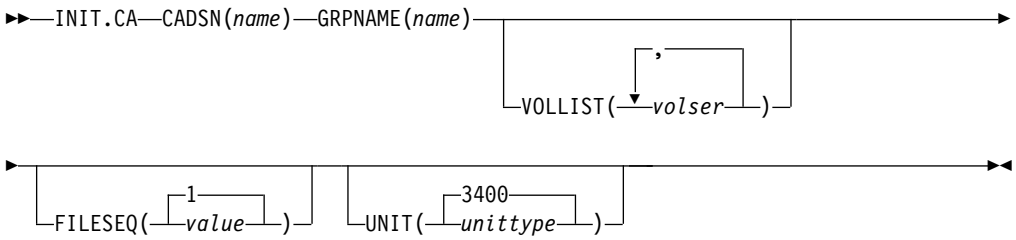
Use the INIT.CA command to create a record in the RECON data set that identifies a change accumulation data set. This data set can be used by the Change Accumulation utility to process a CA group.

You can create such change accumulation records only for those CA groups that have been defined with the REUSE option of the INIT.CAGRP command. You can create change accumulation records in the RECON data set up to the number specified in the GRPMAX keyword of the INIT.CAGRP command that you used to define the CA group.

Subsections:

- "Syntax"
- "Keywords" on page 286
- "Example of creating a record that defines a CA data set" on page 286

Syntax



Keywords

CADSN(*name*)
Required keyword you use to specify the name of the change accumulation data set for which you are creating a record in the RECON data set. The name you substitute in the variable field can be up to 44 characters. You can use the default naming convention for change accumulation data sets to assign this name.

GRPNAME(*name*)
Required keyword you use to specify the name of the CA group for which you are creating the record. The GRPNAME keyword must specify the name of a CA group that is already defined in the RECON data set.

VOLLIST(*volser*)
Optional keyword you use to specify the volume serial numbers of the volumes on which the change accumulation data set being defined is to reside. You can substitute from 1 to 255 volume serial numbers in the variable field. Each volume serial number can be up to 6 alphanumeric characters long, and must follow z/OS JCL conventions for volume serial numbers.

Attention: VOLLIST is required if the change accumulation data set is not to be treated as cataloged, as indicated in the RECON status record.

FILESEQ(1 | *value*)
Optional keyword you use to specify the file sequence number of the change accumulation data set that is being defined. If you specify the FILESEQ keyword with a value greater than 1, you must also specify the VOLLIST keyword.

value must be a decimal number from 1 to 9999.

UNIT(3400 | *unittype*)
Optional keyword you use to specify the unit type of the volumes on which the change accumulation data set is to reside. *unittype* can be up to 8 alphanumeric characters.

If you specify the UNIT keyword, you must also specify the VOLLIST keyword.

Example of creating a record that defines a CA data set

In this example, a record is created in the RECON data set that identifies a change accumulation data set (identified by the CADSN keyword). This change accumulation data set is being created for use by a subsequent run of the Change Accumulation utility for the CA group identified in the GRPNAME keyword. Creation of this record implies that the identified CA group was defined with a REUSE keyword.

```
//INITCA   JOB
⋮
//SYSIN    DD   *
     INIT.CA  GRPNAME(CAGRP1)  -
              CADSN(IMS.CAGRP1.CA.CA001)  -
              VOLLIST(VOL001)      FILESEQ(4)
/*
```

INIT.CAGRP command

Use the INIT.CAGRP command to specify the names of the DBDSs that will belong to a specified CA group.

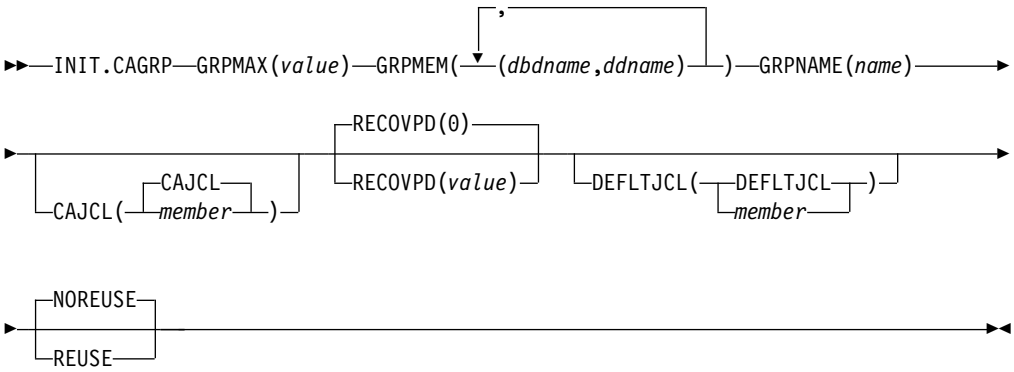
Before you assign a DBDS to a CA group, you must first use the INIT.DBDS command to create a record for the DBDS in the RECON data set. Each DBDS can belong to only one CA group.

Restriction: Index DBDSs and ILDS DBDSs are not recoverable, and changes to them are not logged. The INIT.CAGRP command does not support these data sets.

Subsections:

- "Syntax"
- "Keywords"
- "Example of creating a CA group" on page 289

Syntax



Keywords

GRPMAX(*value*)
Required keyword you use to specify the maximum number of change accumulation data sets that DBRC is to maintain for the specified CA group. *value* must be a decimal number from 2 to 1024.

When the number of times you run the Change Accumulation utility for the specified group exceeds the GRPMAX value, the record with the stop time of the earliest change accumulation execution record beyond the recovery period for the group is deleted if you specify the NOREUSE keyword for this CA group. The record with the stop time of the earliest change accumulation execution record beyond the recovery period is reused if you specify the REUSE keyword for this CA group.

GRPMEM(*dbdname*,*ddname*)
Required keyword you use to specify the names of the DBDSs that are to be members of the CA group you are defining.

A CA group can contain from 1 to 32 767 members. The names you substitute in the variable field must be pairs of names enclosed in parentheses, where *dbdname* is the database name of the DBDS, and *ddname* is the data set ddname or Fast Path area name. When using the INIT.CAGRP command for HALDB databases, *dbdname* specifies the name of a HALDB partition.

Restriction: User-recoverable databases cannot be added to a CA group.

GRPNAME(*name*)
Required keyword you use to specify the name of the CA group being created. *name* can be up to eight alphanumeric characters, and it must not be the same as the name of a CA group that already exists in the RECON data set.

CAJCL(CAJCL | *member*)
Optional keyword you use to specify the name of a member of a partitioned data set of skeletal JCL. You create this member to be used to generate the JCL required to run the Change Accumulation utility for the CA group being created.

RECOVPD(0 | *value*)
Optional keyword you use to set the recovery period for a specified CA group. The recovery period is the amount of time before the current date for which DBRC maintains change accumulation information in the RECON data set. For example, if the recovery period of a change accumulation group is 14 days, DBRC maintains sufficient change accumulation execution records for at least 14 days based on the stop time of the change accumulation execution record.

The recovery period is calculated as the current date minus the stop time date of the oldest change accumulation execution record. If the change accumulation execution records are dated within the days specified in RECOVPD(*value*), DBRC keeps them in the RECON data set.

For *value*, specify a decimal number from 0 to 999 that represents the number of days the change accumulation execution records are kept in the RECON data set. If you specify 0 (the default), there is no recovery period.

DEFLTJCL(DEFLTJCL | *member*)
Optional keyword you use to specify an implicit skeletal JCL default member for the CA group. The specified member is used by the GENJCL.CA command to resolve keywords you have defined.

NOREUSE | REUSE
Mutually exclusive, optional keywords that you use to specify whether the change accumulation data sets for the CA group being defined can be reused.

NOREUSE
Indicates that the change accumulation data sets in this group are not to be reused by the Change Accumulation utility.

REUSE
Indicates that the Change Accumulation utility is to reuse the oldest change accumulation data set and record (for the group being defined) when the GRPMAX value for the group is exceeded. Reuse means that the Change Accumulation utility uses the same physical space, volumes, data set name, and record in the RECON data set for the new change accumulation data set as were used for the oldest change accumulation data set in the group.

Example of creating a CA group

In this example, a CA group is being created. As indicated in the GRPMAX keyword, a maximum of 15 change accumulation data sets are to be maintained for this group. As indicated in the RECOVPD keyword, a retention period of 60 days is set for this group. The NOREUSE keyword indicates that change accumulation data sets for this group are not to be reused by the Change Accumulation utility when the GRPMAX value has been reached. That keyword also implies that empty change accumulation data sets cannot be defined for this group for use in future runs of the Change Accumulation utility.

```
//INITCAGP JOB
⋮
 //SYSIN    DD   *
      INIT.CAGRP  GRPNAME(CAGRP1)  GRPMAX(15)  RECOVPD(60) NOREUSE -
                  GRPMEM((DB1,DD1) (DB2,DD2) (DB3,DD3) -
                  (DB4,DD4) (DB5,DD5) (DB6,DD6) (DB7,DD7) -
                  (DB8,DD8) (DB9,DD9) (DB10,DD10)) –
 /*
```

Related concepts:

Recovery period of change accumulation data sets and GRPMAX (System Administration)

INIT.DB command

Use the INIT.DB command to register a database with DBRC and specify various characteristics of the database.

A database must be registered with DBRC before you can initialize a new DBDS, HALDB partition, or DEDB area with the INIT.DBDS command or INIT.PART.

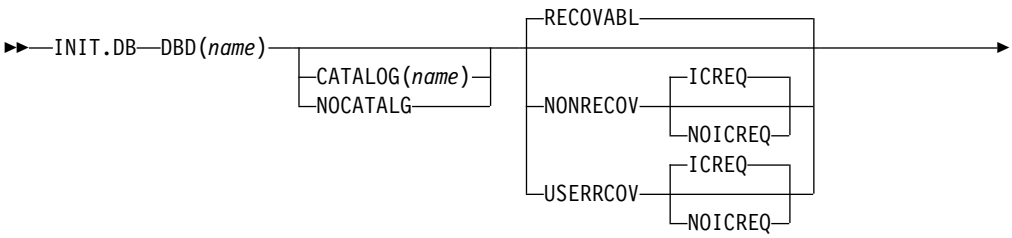
When you issue the INIT.DB command for a HALDB partition (TYPE=PART), the following fields are set as defaults:

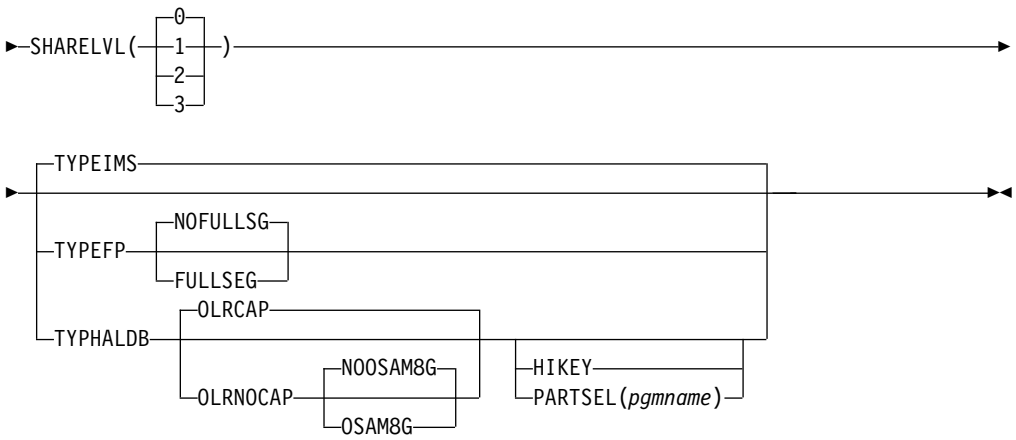
- OLREORG CURSOR ACTIVE =NO
- OLRIMSID=**NULL**
- ACTIVE DBDS=A-J

Subsections:

- "Syntax"
- "Keywords" on page 290
- "Example of creating a SHARELVL 1 DB record" on page 293

Syntax





Keywords

DBD(*name*)
Required keyword that you use to specify the database name of the database to be registered in the RECON data set.

Restriction for HALDB databases: For the INIT.DB command, *name* specifies the name of a HALDB master database.

CATALOG(*name*) | NOCATALG
Mutually exclusive, optional parameters that you use to indicate whether an IMS catalog is to be used during the INIT.DB command. If these parameters are omitted then IMS uses the IMS catalog that is named in the RECON header. If the IMS catalog name in the RECON is null, then DBRC uses the DBDLIB.

CATALOG(*name*)
Specifies that the INIT.DB command uses the specified IMS catalog instead of the DBD library.

Specify *name* as an eight-character alphanumeric string that matches the name of a HALDB IMS catalog database that is registered in the RECON data set. Any IMS catalog name that is less than eight-characters is padded with blanks.

The IMS catalog name is validated during the INIT.DB command. If the catalog does not exist or is not defined as a HALDB database, the command fails.

NOCATALG
Specifies that the INIT.DB command uses the DBD library instead of the IMS catalog.

RECOVABL | NONRECOV | USERRCOV
Mutually exclusive, optional keywords that are used to specify whether DBRC can record the updates in the RECON for the data sets of the specified databases.

RECOVABL
Specifies that the update allocations on the database are to be written in the RECON data set.

NONRECOV
Specifies that no record of the updates for the data sets of the specified database are to be kept in the RECON data set. The effect of the

NONRECOV keyword for full-function databases is different from that for Fast Path databases. When NONRECOV is specified for full-function databases, IMS logs backout information. When NONRECOV is specified for Fast Path databases, IMS does not log backout information.

ICREQ | NOICREQ
Mutually exclusive, optional keywords that are used to specify whether DBRC should enforce the requirement of taking an image copy of a nonrecoverable or user-recoverable database after it has been loaded or reorganized. This option affects all DBDSs within the DB.

ICREQ
Default value that turns on IC NEEDED in the DBDS record and increments the IMAGE COPY NEEDED COUNT in the DB record following initial load (PROCOPT=L) or reorganization of the DBDS.

NOICREQ
Turns off the IC NEEDED requirement. DBRC will not turn on IC NEEDED in the DBDS record or increment the IMAGE COPY NEEDED COUNT in the DB record following initial load or REORG of the DBDS. The IC NEEDED flag can still be set manually with the CHANGE.DBDS ICON command.

USERRCOV
Specifies that update allocations for the Fast Path database are not to be recorded in the RECON data set. Database updates are logged, however, DBRC will not be able to manage recovery for the database. Recovery is allowed with JCL manually generated by the user. This keyword is only valid for TYPEFP.

Notes:

1. USERRCOV should only be used if a critical need to reduce database open time exists and recoverability is required.
2. USERRCOV requires the user to provide their own manually created JCL for change accumulation and recovery.

ICREQ | NOICREQ
Mutually exclusive, optional keywords that are used to specify whether DBRC should enforce the requirement of taking an image copy of a nonrecoverable or user-recoverable database after it has been loaded or reorganized. This option affects all DBDSs within the DB.

ICREQ
Default value that turns on IC NEEDED in the DBDS record and increments the IMAGE COPY NEEDED COUNT in the DB record following initial load (PROCOPT=L) or reorganization of the DBDS.

NOICREQ
Turns off the IC NEEDED requirement. DBRC will not turn on IC NEEDED in the DBDS record or increment the IMAGE COPY NEEDED COUNT in the DB record following initial load or REORG of the DBDS. The IC NEEDED flag can still be set manually with the CHANGE.DBDS ICON command.

Restrictions:

- If the database is registered as RECOVABL or USERRCOV, VIO or DD DUMMY data sets cannot be used for the output log (IEFRDER) in any job

that updates the database. Temporary log data sets, such as VIO data sets, are deleted at job termination, so they are not usable for recovery.

- You cannot make concurrent image copies of nonrecoverable or user-recoverable databases.

SHARELVL(0 | 1 | 2 | 3)
Optional keyword that you use to specify the level of data sharing for which authorized subsystems can share a database.

Restrictions:

- You must specify a share level of 1, 2, or 3 for concurrent image copies.
- If you are using IRLM, and specified a share level 2 or 3, ensure that the VSAM SHAREOPTIONS (3 3) keyword is also specified in the database data set.

TYPEIMS | TYPEFP | TYPHALDB
Mutually exclusive, optional keywords that you use to specify whether the database is a full function, Fast Path DEDB, or a HALDB.

TYPEIMS
Specifies that the database is a DL/I database (non-HALDB).

TYPEFP
Specifies that the database is Fast Path DEDB.

NOFULLSG | FULLSEG

Mutually exclusive, optional keywords that specify whether the full segment image is logged in the X'5950' log record when the segment is updated by a Replace (REPL) call. These keywords are valid only for Fast Path DEDBs.This setting is saved in the database record for this DEDB and is used as the default setting for any areas that are registered later for this DEDB.

NOFULLSG
Indicates that only the updated portion of a segment is to be logged. This is the default.

FULLSEG
Indicates that the full segment image is to be logged.

TYPHALDB
Specifies that the database is a DL/I database (HALDB).

When INIT.DB TYPHALDB is issued, DBRC requires access to the database definition. If the IMS system manages ACBs, the definitions are retrieved from the IMS catalog and the name of the IMS catalog must be defined in the RECON header or the INIT.DB command itself. If the IMS system uses DBD, PSB, and ACB libraries, the IMS DBDLIB data set must be identified in the job stream for the Database Recovery Control utility (DSPURX00) with a ddname of IMS.

The following additional keywords can be specified only when TYPHALDB is specified:

OLRCAP | OLRNOCAP
Mutually exclusive, optional keywords that are used to indicate that HALDB OLR is allowed to run on that database.

OLRCAP
Specifies in the master database record and in the partition

database record that HALDB OLR is allowed to run on that database. This is the default (except for a PSINDEX, which is always OLRNOCAP).

OLRNOCAP
Specifies in the master database record and in the partition database record that HALDB OLR is not allowed to run on that database.

NOOSAM8G | OSAM8G
Mutually exclusive, optional keywords that specify in the master database record whether a HALDB PHDAM or PHIDAM database supports up to 4 or 8 gigabytes of data in OSAM database data sets. These keywords are valid only for HALDB master databases.

NOOSAM8G
Specifies that the maximum capacity of OSAM data sets is 4 gigabytes of data. This is the default.

OSAM8G
Specifies that the maximum capacity of OSAM data sets is 8 gigabytes of data. Only a database that is not OLR capable can be OSAM8G. When you specify OSAM8G, you must also specify OLRNOCAP.

HIKEY | PARTSEL(*pgmname*)
Mutually exclusive, optional keywords that you use to identify whether the HALDB uses a Partition Selection Exit or high key values. This setting determines whether the KEYSTRNG keyword on the INIT.PART command defines a partition selection exit string or a high key value. These keywords are valid only with TYPHALDB.

PARTSEL(*pgmname*)
Specifies the name of the Partition Selection Exit program to be used for the HALDB. The *pgmname* value is specified as a value up to 8 characters long that is a valid program name.

HIKEY
Specifies that this HALDB uses high key values.

If both PARTSEL and HIKEY are omitted, the HALDB will use the Partition Selection Exit module name (PSNAME) from the HALDB DBD. If no Partition Selection Exit module name was specified in the DBD, the HALDB will use high key values.

Example of creating a SHARELVL 1 DB record

In this example, a new database record is created in the RECON data set. This database has a share level of 1.

```
//INITDB    JOB
.
.
.
//SYSIN     DD   *
     INIT.DB  DBD(THISDB)  SHARELVL(1) TYPEFP
/*
```

Example using the DBDLIB instead of the default catalog name

```
 //INITDB JOB

⋮
 //IMS      DD   DSN=IMS.DBDLIB,DISP=SHR
 //SYSIN    DD   *
      INIT.DB  DBD(DBD002) NOCATALG
 /*
```

Example using the catalog name GREGCD00 instead of the default catalog name

```
//INITDB JOB

⋮
 //SYSIN    DD   *
      INIT.DB  DBD(DBD002) CATALOG(GREGCD00)
 /*
```

Related concepts:

Coordinating VSAM data set definitions with share options (System Administration)

Levels of data sharing (System Administration)

Nonrecoverable full-function databases (Database Administration)

Related reference:

/RMxxxxxx commands (Commands)

"INIT.PART command" on page 305

"INIT.DBDS command"

"LIST.DB command" on page 324

INIT.DBDS command

Use the INIT.DBDS command to register a DBDS or DEDB area.

The DBDS must exist for any of the other commands to work for a given DBDS or DEDB area. In order to register the DBDS, DBRC examines the IMS catalog or the IMS DBDLIB data set to:

- Verify that the DBDS or DEDB area exists.
- Obtain the data set identifier (DSID) of the DBDS, its data set organization, and its database organization.

If the IMS management of ACBs is enabled, DBRC examines the IMS catalog that is specified on the CATALOG keyword of the INIT.DBDS command or that is recorded in the RECON header as the default IMS catalog.

If your installation manages ACBs with an ACB library instead of the IMS catalog, DBRC examines the IMS DBDLIB data set. The IMS DBDLIB data set must be identified in the job stream for the Database Recovery Control utility (DSPURX00) with a ddname of IMS.

The INIT.DBDS command fails if you issue it while the database is in use.

When you issue the INIT.DBDS command, DBRC sets the image copy status in the DBDS record in the RECON to IC RECOMMENDED. After registering the DBDS, you can reset the image copy status in the RECON by performing either of the two following actions:

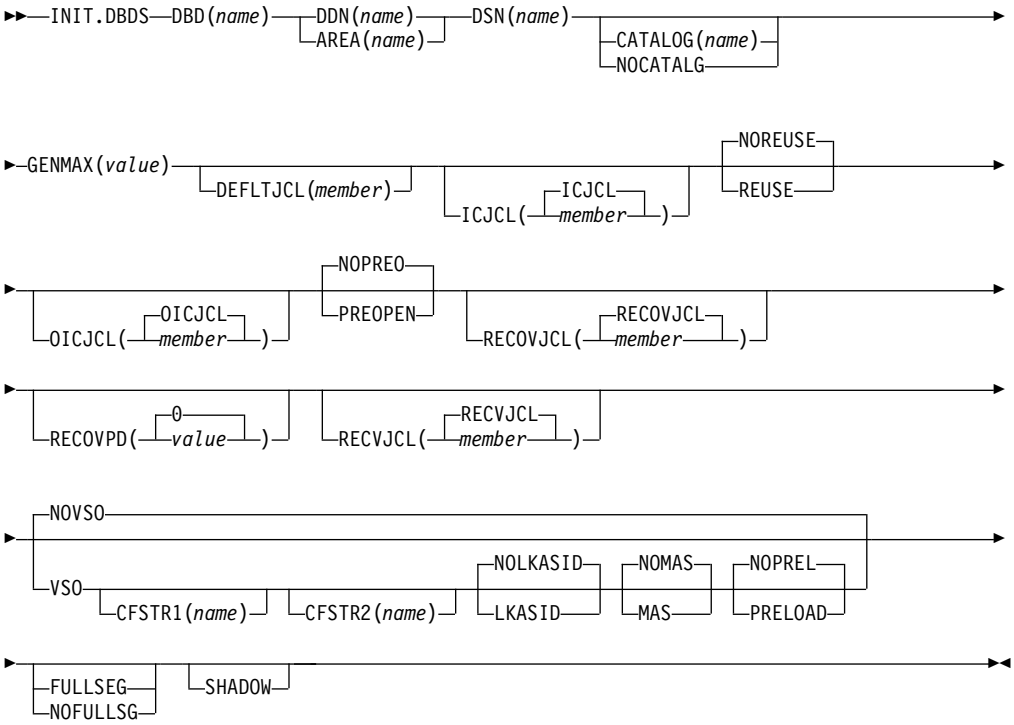
- Use the Image Copy utility to create a backup copy.
- Use the CHANGE.DBDS command to change the image copy status in the RECON data set.

Restriction for HALDB databases: You cannot use the INIT.DBDS command to register DBDSs of HALDB databases. You must instead use either the HALDB Partition Definition utility or the INIT.DB and INIT.PART commands.

Subsections:

- "Syntax"
- "Keywords"
- "Example of identifying the DBDS to initiate DBRC control over recovery" on page 300

Syntax



Keywords

DBD(*name*)
Required keyword you use to specify the database name of the DBDS or DEDB area being identified to DBRC.

DDN(*name*) | AREA(*name*)
Mutually exclusive, required keywords that you use to specify the ddname of the DBDS or DEDB area being identified to DBRC.

DSN(*name*)
Required keyword you use with the DDN(*name*) keyword to specify the data set name of the DBDS being identified to DBRC. You cannot use this keyword with a DEDB area.

If the DBDS is an ADS that is registered to DBRC, do not specify this keyword. Instead specify the data set name in the INIT.ADS command.

CATALOG(*name*) | NOCATALG
Mutually exclusive, optional parameters that you use to indicate whether a catalog is to be used during the INIT.DBDS command. If these parameters are omitted then IMS uses the default catalog name that is defined in the RECON header. If the default name is null then the DBDLIB is used.

For databases that are defined using Database Definition Language (DDL), DBRC must access a catalog.

NOCATALG
Specifies that theINIT.DBDS command uses the DBD library instead of the IMS catalog.

CATALOG(*name*)
Specifies that the INIT.DBDS command uses the specified IMS catalog instead of the DBD library.

Specify *name* as an eight-character alphanumeric string that matches the name of a HALDB IMS catalog database that is registered in the RECON data set. Any IMS catalog name that is less than eight-characters is padded with blanks.

The IMS catalog name is validated during the INIT.DBDS command. If the catalog does not exist or is not defined as a HALDB database, the command fails.

GENMAX(*value*)
Required keyword you use to specify the maximum number of image copies that DBRC is to maintain for the identified DBDS.

Each time you run the Database Image Copy utility for the DBDS that is being initiated, a new image copy is defined in the RECON data set. If you identified the DBDS with the NOREUSE keyword, the oldest image copy for the DBDS beyond the recovery period is deleted when the number of image copies exceeds the GENMAX value. If you identify the DBDS with the REUSE keyword, the oldest image copy beyond the recovery period is reused.

value must be a decimal number from 2 to 255.

DEFLTJCL(*member*)
Optional keyword you use to specify an implicit skeletal JCL default member for the DBDS. The specified member is used by the GENJCL.IC, GENJCL.OIC, and GENJCL.RECOV commands in order to resolve keywords you have defined.

ICJCL(ICJCL | *member*)
Optional keyword you use to specify the name of a member of a partitioned data set that contains skeletal JCL. When you issue a GENJCL.IC command, DBRC uses this member to generate the JCL to run the Database Image Copy utility for the DBDS or DEDB area being identified.

NOREUSE | REUSE
Mutually exclusive, optional keywords that you use to specify whether the supported image copy utilities are to reuse previously used image copy data sets.

NOREUSE
Prevents the automatic reuse of image copy data sets for this DBDS by the supported image copy utilities.

If the NOREUSE option is specified for the HISAM database, the image-copy-needed flag is not turned on at the end of the HISAM Reload utilities. The input data set that is used while the HISAM database is being reloaded is used as an image copy data set.

REUSE
Allows the GENJCL.IC command or the GENJCL.OIC command to generate a job that causes the supported image copy utilities to reuse the oldest image copy data set (for the DBDS being defined) when the GENMAX value for the DBDS is exceeded. REUSE requires that you create empty image copy data sets for future use by the supported image copy utilities. In addition, you must use INIT.IC commands to record their existence in the RECON data set. The NOREUSE keyword prohibits such actions.

If you want HSSP image copy processing, you must specify REUSE. Reuse means that the image copy job uses the same volumes, data set name, and record in the RECON data set for the new image copy data set as those of the oldest DBDS image copy data set.

OICJCL(OICJCL | *member*)
Optional keyword you use to specify the name of a member of a partitioned data set that contains skeletal JCL. You cannot use this keyword with a DEDB area. When you issue a GENJCL.OIC command, DBRC uses this member to generate the JCL to run the Online Database Image Copy utility for the DBDS being identified.

NOPREO | PREOPEN
Mutually exclusive, optional keywords that you use to specify whether an area is to be opened after the first checkpoint following the next control region initialization or when the next /START AREA command is processed. NOPREO is the default, except if you specify PRELOAD, in which case PREOPEN is the default.

NOPREO
Indicates that the area is not to be pre-opened the next time the control region is started or a /START AREA command is processed. You cannot specify NOPREO with PRELOAD.

PREOPEN
Indicates that the area is to be opened the next time the control region is started or a /STA AREA command is processed. This option is valid for both VSO and non-VSO areas.

RECOVJCL(RECOVJCL | *member*)
Optional keyword you use to specify the name of a member of a partitioned data set of skeletal JCL. When you issue the GENJCL.RECOV command, DBRC uses this member to generate the JCL that runs the Database Recovery utility for the DBDS or area being identified.

RECOVPD(0 | *value*)
Optional keyword you use to specify the recovery period for a specified DBDS or DEDB area.

The recovery period is the amount of time before the current date for which DBRC maintains recovery information in the RECON data set. For example, if the recovery period of a DBDS or area is 14 days, DBRC maintains sufficient recovery-generation information for at least 14 days.

The recovery period is calculated as the current date minus the date of the oldest image copy. If the image copies are dated within the days specified in the RECOVPD(*value*), DBRC keeps them in the RECON data set.

For *value*, specify a decimal number from 0 to 999 that represent the number of days the image copies are to be kept in the RECON data set. If you specify 0 (the default), there is no recovery period.

RECVJCL(RECVJCL | *member*)
Optional keyword you use to specify the name of the skeletal JCL member to be used by the GENJCL.RECEIVE command.

RECVJCL can be specified for DL/I DBDSs and Fast Path areas.

NOVSO | VSO
Mutually exclusive, optional keywords that you use to specify whether an area is to reside in virtual storage the next time the control region is initialized or when the next /STA AREA command is processed.

NOVSO
Indicates that this area is not to reside in virtual storage.

VSO
Indicates that the area is to reside in virtual storage. Areas that are defined with SHARELVL(0 | 1) are read into and written from a z/OS data space. Areas defined with SHARELVL(2 | 3) use the coupling facility to share data between connected subsystems.

CFSTR1(*name*)
Optional keyword you use to specify the name of the first coupling facility structure for the identified area. z/OS coupling facility structure naming conventions must be adhered to. This keyword is valid only for VSO areas of DEDBs that are defined with SHARELVL(2 | 3). The area name is the default if VSO is specified and the DEDB is SHARELVL(2 | 3).

CFSTR2(*name*)
Optional keyword you use to specify the name of the second coupling facility structure for the identified area. z/OS coupling facility structure naming conventions must be adhered to. This keyword is valid only for VSO area of DEDBs defined with SHARELVL(2 | 3). There is no default.

Restriction: The name cannot be the area name if the CFSTR1 keyword is not specified. CFSTR2 cannot be specified if MAS is specified.

NOLKASID | LKASID
Mutually exclusive optional keywords that you use to specify whether local data caching for the specified area is to be used for buffer lookaside on read requests. The LKASID option is valid only for SHARELVL(2 | 3) VSO areas.

NOLKASID
Indicates that buffer lookaside is not to be performed on read requests for this area. NOLKASID is the default.

LKASID
Indicates that buffer lookaside is to be performed on read requests for this area.

For VSO DEDB areas that use a single-area structure, the LKASID value that is defined in the RECON data set for the area takes precedence over the DFSVSMxx PROCLIB member's DEDB LKASID value.

For VSO DEDB areas that use a multi-area structure, LKASID or NOLKASID must be specified by using the DFSVSMxx PROCLIB member. The LKASID value that is defined in the RECON data set for the area is ignored.

NOMAS | MAS
Mutually exclusive optional keywords that you use to specify whether a shared VSO is to reside in a multi-area coupling facility structure.

NOMAS
Indicates that the coupling facility structure (or structures) should contain data only for this area. NOMAS is the default.

MAS
Indicates that the area is to reside in a multi-area coupling facility structure. If MAS is specified, a second coupling facility structure (CFSTR2) cannot be specified.

NOPREL | PRELOAD
Mutually exclusive, optional keywords that you use to specify whether a VSO area is to be loaded the next time it is opened.

NOPREL
Indicates that the area is not to be loaded into a data space or coupling facility structure the next time that it is opened. For VSO areas, CIs are copied into a data space when they are read for the first time.

PRELOAD
Indicates that the area is to be loaded into a data space or coupling facility structure the next time that it is opened. Selecting this option also causes the area to be pre-opened.

FULLSEG | NOFULLSG

Mutually exclusive, optional keywords that specify whether the full segment image is logged in the X'5950' log record when the segment is updated by a Replace (REPL) call. These keywords are valid only for Fast Path DEDBs.

If one of these keywords is specified, it overrides the setting of the INIT.DB command. If neither of these keywords is specified, the default value that is set in the database record for the DEDB is used. See the INIT.DB command.

FULLSEG
Indicates that the full segment image is to be logged.

NOFULLSG
Indicates that only the updated portion of a segment is to be logged.

SHADOW
Optional keyword used to specify whether to register a shadow area for a new DEDB area that is added to the end of a DEDB database by the ADDAREA function of the DEDB Alter utility. This keyword is only valid for Fast Path DEDBs.

The DEDB Alter function is required for shadow areas to be accessible to IMS systems.

Example of identifying the DBDS to initiate DBRC control over recovery

In this example, a DBDS is registered with DBRC. The IMS DD statement is required to enable access to the IMS DBDLIB data set to obtain the data set identifier, data set organization, and database organization of the DBDS. The DBDS is identified by the DBD, DDN, and DSN keywords and is accessed only by an IMS system. Image copy data sets (of the DBDS) can be reused and DBRC maintains a maximum number of 2 image copies. The ICJCL keyword specifies the member of the partitioned data set of skeletal JCL that is to be used for the generation of JCL for the Database Image Copy utility. The RECOVJCL keyword does the same for the Database Recovery utility.

```
 //INITDBDS JOB

.
.
.
 //IMS      DD   DSN=IMS.DBDLIB,DISP=SHR
 //SYSIN    DD   *
      INIT.DBDS  DBD(DBD002)  DDN(DDN003) GENMAX(2) REUSE -
                 ICJCL(ICJCLX) RECOVJCL(RECOVJCX) DSN(DSN003)
 /*
```

Example using the DBDLIB instead of the default catalog name

```
//INITDBDS JOB

.
.
.
 //IMS      DD   DSN=IMS.DBDLIB,DISP=SHR
 //SYSIN    DD   *
      INIT.DBDS  DBD(DBD002)  DDN(DDN003) GENMAX(2) REUSE -
                 ICJCL(ICJCLX) RECOVJCL(RECOVJCX) DSN(DSN003) –
                 NOCATALG
 /*

Example using the catalog name GREGCD00 instead of the default catalog name.

//INITDBDS JOB

.
.
.
//SYSIN    DD   *
      INIT.DBDS  DBD(DBD002)  DDN(DDN003) GENMAX(2) REUSE -
                 ICJCL(ICJCLX) RECOVJCL(RECOVJCX) DSN(DSN003) –
                 CATALOG(GREGCD00)
 /*
```

Related concepts:

Defining a VSO DEDB area (Database Administration)

Recovery period of image copy data sets and GENMAX (System Administration)

Related reference:

"INIT.PART command" on page 305

"LIST.DBDS command" on page 326

/RMxxxxxx commands (Commands)
"CHANGE.DBDS command" on page 84

INIT.DBDSGRP command

Use the INIT.DBDSGRP command to define a DBDS group (DBDSs or DEDB areas), a DB group (DL/I databases or DEDB areas), or a recovery group (DL/I databases or DEDB areas).

A DBDS group can be used anywhere that a DB group can be used, such as for the /DBR command, but this usage is inefficient. Define a separate DB group for such use.

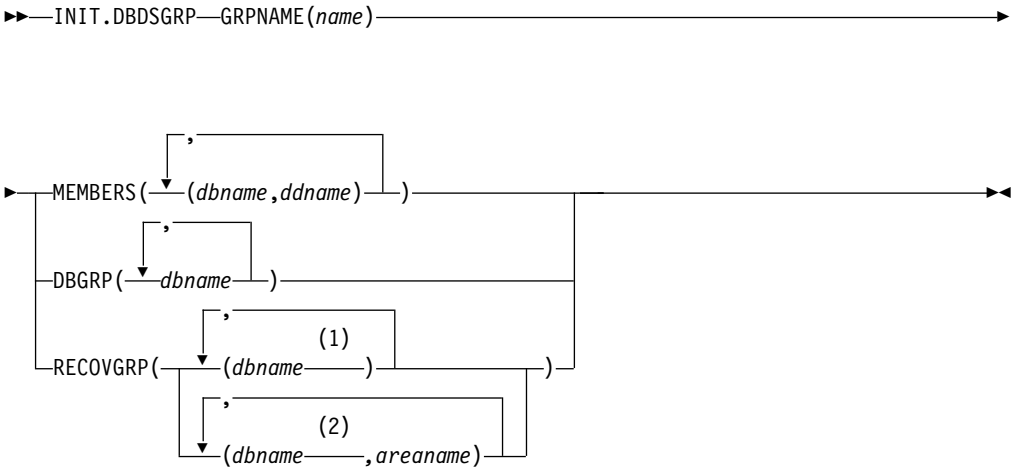
A recovery group is used with database recoveries. It can also be used anywhere that a DB group can be used.

If you define a DBDS group or a recovery group, you must register the members of the group in the RECON data set. Registering members of a DB group in the RECON data set is optional. To register members of a DBDS, recovery, or DB group, use the appropriate INIT command.

Subsections:

- "Syntax"
- "Keywords"
- "Example of creating a group of DBDSs" on page 302

Syntax



Notes:

1 Full-function database name

2 Fast Path DEDB name

Keywords

GRPNAME(*name*)
Required keyword you use to identify the DBDSGRP to be created. The name can be from one to 8 alphanumeric characters, and must not be the name of an existing DBDSGRP or CAGRP record.

MEMBERS(*dbname,ddname*) | DBGRP(*dbname*) | RECOVGRP(*dbname,areaname*)
Mutually exclusive, required keywords that identify the members to be included in the new group. A group can contain up to 32767 members.

MEMBERS(*dbname,ddname*)
Indicates that the group is a DBDS group. This keyword identifies one or more DBDSs or DEDB areas, each by a pair of names enclosed in parentheses, where *dbname* is the database name and *ddname* is the DD statement name or the DEDB area name.

Any member can belong to more than one DBDS group.

DBGRP(*dbname*)
Indicates that the group is a DB group, and identifies one or more databases or area names. Any member can belong to more than one DB group.

RECOVGRP(*dbname,areaname*)

Indicates that the group is a *recovery group*. A recovery group is a group of full-function databases or DEDB areas that you consider to be related.

DEDB areas can be a part of a recovery group, but DEDB databases cannot. HALDB databases can be a part of a recovery group, but individual HALDB partitions cannot.

If you perform a time-stamp recovery on one of the members of the group, some database recovery tools require you to recover all members of the group to the same time. A recovery group otherwise can be used like a DB group.

If you add a DEDB area to the recovery group, you must specify both the DEDB area and the DEDB database that the area belongs to. That is, to add a DEDB area to the recovery group, you must specify both *dbname* and *areaname*. You cannot specify only *areaname*.

If you add a full-function database to the recovery group, you do not need to specify *areaname* because *areaname* applies only to Fast Path databases.

A database or area can belong to only one recovery group. If any of the members specified by RECOVGRP already belongs to another recovery group, the command fails.

Restrictions for HALDB databases: For HALDB databases, *dbname* specifies the name of a HALDB partition or the HALDB master database. When specifying HALDB database names, use the INIT.DBDSGRP command only as defined in the following table.

Table 14. Supported dbname specifications for HALDB databases on INIT.DBDSGRP command keyword

INIT.DBDSGRP	*dbname* **= HALDB master**	*dbname* **= Partition**
MEMBERS	No	Yes
DBGRP	Yes	Yes
RECOVGRP	Yes	No

Example of creating a group of DBDSs

In this example, a group of DBDSs is defined.

```
//INITDBGRP   JOB
⋮
//SYSIN       DD  *
     INIT.DBDSGRP    GRPNAME(DBDSG1)   -
                     MEMBERS((DB1,DD1),(DB2,DD2),(DB3,DD3))
/*
```

Related reference:

"INIT.DB command" on page 289

"INIT.DBDS command" on page 294

INIT.IC command

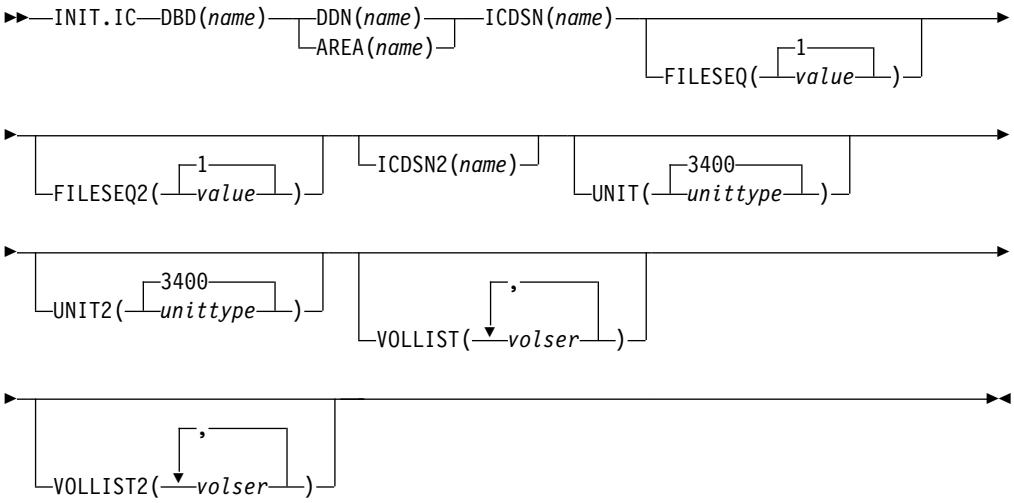
Use an INIT.IC command to create image copy records in the RECON data set. These image copy records define image copy data sets that are available for use during subsequent runs of the supported image copy utilities.

Each INIT.IC command creates one image copy record. You can define image copy data sets for subsequent use only if you have specified a REUSE keyword for the corresponding DBDS or DEDB area when it was identified in the RECON data set with an INIT.DBDS command. The maximum number of image copy records that are to be used for a given DBDS or DEDB area is determined by the value of GENMAX for the specified DBDS or DEDB area.

Subsections:

- "Syntax"
- "Keywords"
- "Example of creating a record that defines the ICDSN" on page 305

Syntax



Keywords

DBD(*name*)

Required keyword you use to identify the image copy data set being created by the database name of its related DBDS or DEDB area.

Restriction for HALDB databases: For the INIT.IC command, *name* specifies the name of a HALDB partition.

DDN(*name*) | AREA(*name*)
Mutually exclusive, required keywords that you use to identify the image copy data set being created; *name* is the data set ddname of the related DBDS or DEDB area.

ICDSN(*name*)
Required keyword you use to specify the name of the image copy data set for which the image copy record is being created. *name* can be up to 44 characters. You can use the default-naming convention for image copy data sets for this name.

FILESEQ(1 | *value*)
Optional keyword you use to specify the file sequence number of the image copy data set for which the image copy record is being created. You can specify this keyword only if you specify a VOLLIST keyword, and only if the file sequence number is not 1. *value* must be a decimal number from 1 to 9999.

FILESEQ2(1 | *value*)
Optional keyword you use to specify the file-sequence number of the duplicate image copy data set for which the image copy record is being created. You can specify this keyword only if you are creating a duplicate image copy data set, if you specify a VOLLIST2 keyword, and if the file-sequence number is not 1. The value you substitute in the variable field must be a decimal number from 1 to 9999.

ICDSN2(*name*)
Optional keyword you use to specify the name of the duplicate image copy data set for which the image copy record is being created. *name* can be up to 44 characters. You can use the default naming convention for duplicate image copy data sets for this name.

UNIT(3400 | *unittype*)
Optional keyword you use to specify the unit type of the image copy data set being defined. The unit type can be up to eight alphanumeric characters long.

If you specify the UNIT keyword, you must also specify the VOLLIST keyword.

UNIT2(3400 | *unittype*)
Optional keyword you use to specify the unit type of the duplicate image copy data set being defined. The unit type can be up to eight alphanumeric characters.

VOLLIST(*volser*)
Optional keyword you use to specify the volume serial numbers of the volumes on which the image copy data set resides. You can substitute from 1 to 255 volume serial numbers in the variable field. Each volume serial number can be up to 6 alphanumeric characters, and they must follow z/OS® JCL conventions for volume serial numbers. VOLLIST is required if the image copy data set is not to be treated as cataloged, as indicated in the RECON status record.

VOLLIST2(*volser*)
Optional keyword you use to specify the volume serial numbers of the volumes on which the duplicate image copy data set resides. You can substitute from 1 to 255 volume serial numbers in the variable field. Each volume serial number can be up to 6 alphanumeric characters, and they must follow z/OS® JCL conventions for volume serial numbers. VOLLIST2 is only allowed if ICDSN2 is specified. VOLLIST2 is required if the image copy data set is not to be treated as cataloged, as indicated in the RECON status record.

Example of creating a record that defines the ICDSN

In this example, a record is created in the RECON data set that defines an image copy data set that is to be used for creating an image copy of the DBDS. The name of the image copy data set is specified in the ICDSN keyword; in this example, the default naming convention is used to generate the fully qualified data set name. The volume on which the image copy data set is to reside is specified by the VOLLIST keyword and its file-sequence number is specified by the FILESEQ keyword.

```
//INITIC   JOB
:
//SYSIN    DD   *
     INIT.IC     DBD(DB1)  DDN(DD1)  -
                 ICDSN(IMS.*.ICDSN2)  -
                 VOLLIST(VOL003)   FILESEQ(5)
/*
```

INIT.PART command

Use the INIT.PART command to register a HALDB partition.

This command creates the RECON HALDB partition structure: a PART record, the partition DB record, and one or more DBDS records, according to the DBD specification.

The INIT.PART command fails if the HALDB is being used by the HALDB Partition Definition utility.

For the INIT.PART command, DBRC references the database definitions in either the IMS catalog or the IMS DBDLIB.

If the IMS management of ACBs is enabled, DBRC references the IMS catalog. The name of the IMS catalog must be specified on the CATALOG keyword of the INIT.PART command or the default IMS catalog name must be recorded in the RECON header.

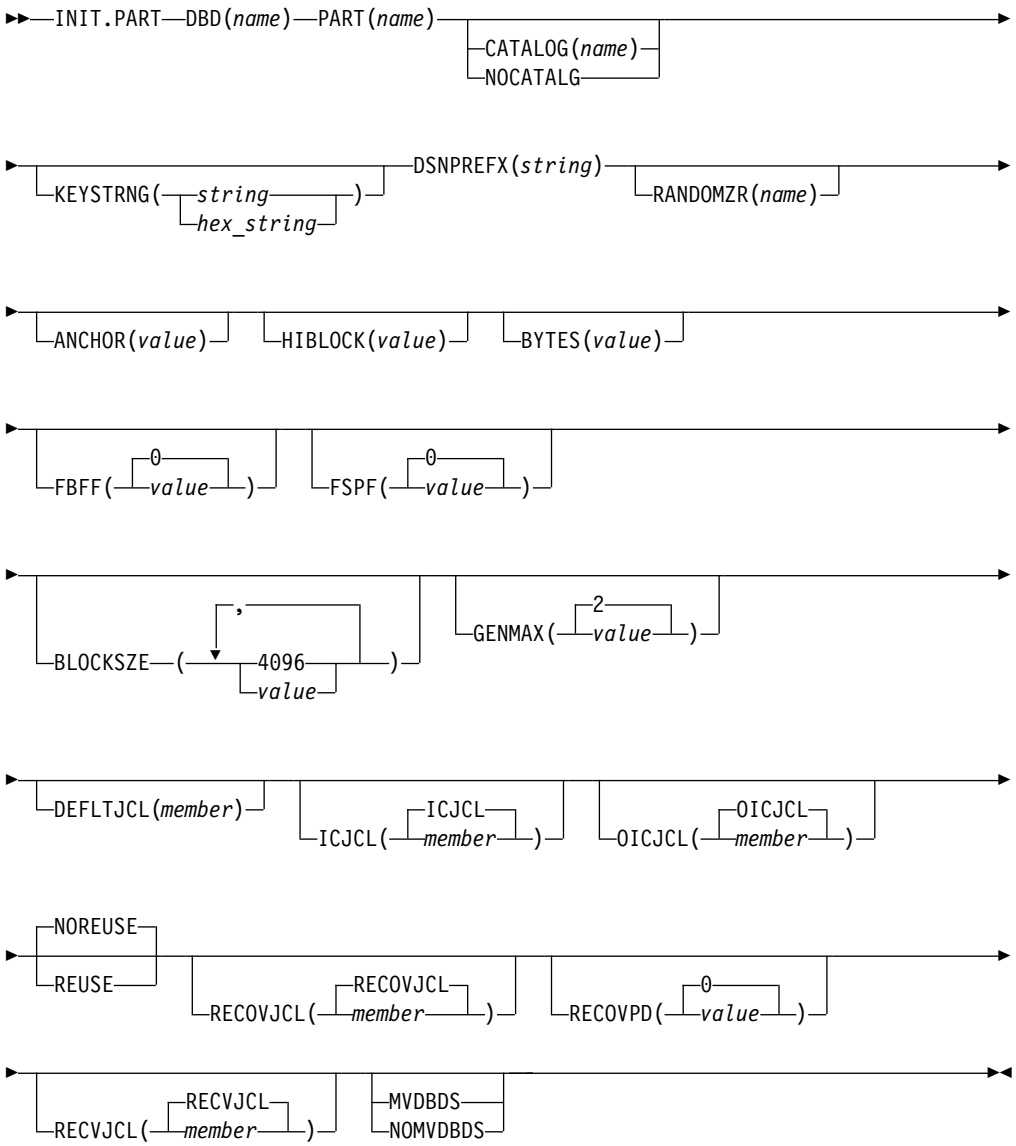
If your installation manages ACBs with an ACB library instead of the IMS catalog, DBRC references the IMS DBDLIB data set. The IMS DBDLIB data set must be identified in the job stream for the Database Recovery Control utility (DSPURX00) with a ddname of IMS.

Some keywords for this command apply to all the partition DBDSs that are created as a result of this command. This differs from the HALDB Partition Definition utility, for which the keywords can be specified separately for each partition DBDS that is created. These keywords can later be changed individually with the CHANGE.DBDS command.

Subsections:

- "Syntax"
- "Keywords" on page 306
- "Example of creating a partition" on page 310

Syntax



Keywords

DBD(*name*)
Required keyword used to identify the HALDB for which the partition is to be defined.

PART(*name*)
Required keyword used to identify a HALDB partition name. Specified as an alphanumeric value, up to 7 characters long, with the first character being alphabetic.

CATALOG(*name*) | NOCATALG
Mutually exclusive, optional parameters that you use to indicate whether a catalog is to be used during the INIT.PART command. If these parameters are omitted then IMS will use the default catalog name defined in the RECON header. If the default name is null then the DBDLIB will be used.

For databases that are defined using Database Definition Language (DDL), DBRC must access a catalog.

CATALOG(*name*)
Specifies that the INIT.PART command uses the specified IMS catalog instead of the DBD library.

Specify *name* as an eight-character alphanumeric string that matches the name of a HALDB IMS catalog database that is registered in the RECON data set. Any catalog name that is less than eight-characters is padded with blanks.

The catalog name is validated during the INIT.PART command. If the catalog does not exist or is not defined as a HALDB database, the command fails.

NOCATALG
Specifies that the INIT.PART command uses the DBD library instead of the IMS catalog.

KEYSTRNG(*string*) | KEYSTRNG(*hex_string*)
Optional keyword that you use to specify a HALDB partition high key value or a selection string for use by a partition selection exit. Specified as a character value up to 256 characters long or a hexadecimal value up to 512 characters long. Character values must be alphanumeric (with no embedded blanks or commas unless the string is enclosed by single quotation marks). Unless enclosed by single quotation marks, the character string will be folded to uppercase. Hexadecimal values must be enclosed by single quotation marks and preceded by the letter X, for example: KEYSTRNG(X'D7C1D9E3D2C5E8').

If no partition selection routine was specified in the HALDB master definition, KEYSTRNG defines the Partition high key and is required. The high key length cannot be longer than the root key length. If the high key length is less than the defined root key length, the high key value is padded with X'FF's up to the defined root key length. The partition high key values must be unique for each partition within a HALDB.

If a partition selection routine was specified in the HALDB master definition, KEYSTRNG defines a Partition Selection String which is passed to the partition selection routine. Your installation partition selection routine may or may not require a Partition Selection String. If required, the content of the string is determined by your installation. It can be up to 256 bytes long and consist of simple character information. If it contains non-printable characters, it must be identified using hex notation. A hex character string is enclosed by single quotation marks and prefixed with an X.

DSNPREFX(*string*)
Required keyword you use to specify the data set name prefix for the partition data sets contained in a HALDB. Specified as a value, up to 37 characters long, that is a valid JCL data set name.

RANDOMZR(*name*)
Optional keyword used to specify the name of the randomizing module for HALDB PHDAM databases only. If RANDOMZR is omitted, the name of the randomizing module is obtained from the DBD. A randomizing module controls root segment placement in, or retrieval from, the PHDAM HALDB.

ANCHOR(*value*)
Optional keyword used to specify the number of root anchor points (RAPs) desired in each control interval or block in the root addressable area of a PHDAM HALDB. This keyword is for PHDAM HALDB databases only.

The value specified must be between 1 and 255. Typical values are from 1 to 5. If ANCHOR is omitted, the value is obtained from the DBD.

HIBLOCK(*value*)
Optional keyword used to specify the maximum relative block number value that you want to enable a randomizing module to produce for this HALDB. This keyword is for PHDAM HALDB databases only.

The HIBLOCK value determines the number of control intervals or blocks in the root addressable area of an PHDAM HALDB. The value can range between 1 and 16,777,215 (2**24-1). If HIBLOCK is omitted, the value is obtained from the DBD.

BYTES(*value*)
Optional keyword used to specify the maximum number of bytes of a HALDB database record that can be stored into the root addressable area in a series of inserts unbroken by a call to another HALDB database record. The value can range between 0 and 16,777,215 (2**24-1). If BYTES is omitted, the value is obtained from the DBD. This keyword is for PHDAM HALDB databases only.

A value of 0 (zero) means that all bytes are addressable. Setting BYTES to 0 is equivalent to omitting BYTES on the DBD definition.

FBFF(0 | *value*)
Optional keyword used to specify the free block frequency factor (fbff) which specifies that every *n*th control interval or block in this data set group is left as free space during database load or reorganization (where FBFF=*n*). The range of FBFF includes all integer values from 0 to 100 except 1. The default value for FBFF is 0.

FSPF(0 | *value*)
Optional keyword used to specify the free space percentage factor. It specifies the minimum percentage of each control interval or block that is to be left as free space in this data set group. Value may be any number between 0 and 99. The default value for FSPF is 0.

BLOCKSZE(4096 | *value*)
Optional keyword that you use to specify the block size for OSAM data sets. Specify an even number no greater than 32,766. The block size value is used for OSAM only. The default is 4096. You may specify up to 10 values, one for each data set group defined in the DBD.

GENMAX(2 | *value*)
Optional keyword that you use to specify the maximum number of image copies that DBRC is to maintain for the partition DBDSs. If you identify a partition DBDS with the NOREUSE keyword, the oldest image copy beyond the recovery period is deleted when the number of image copies exceeds the GENMAX value. If you identify it with the REUSE keyword, the oldest image copy beyond the recovery period is reused. Specified as a numeric value from 2 to 255. All partition DBDSs will be created with this GENMAX value. The CHANGE.DBDS command can be used to change this for individual partition DBDSs. The default value for GENMAX is 2.

DEFLTJCL(*member*)
Optional keyword that you use to specify an implicit skeletal JCL default member for a HALDB Partition DBDS. The specified member is used by the GENJCL.IC, GENJCL.OIC, and GENJCL.RECOV commands in order to resolve keywords you have defined. All partition DBDSs will be created with this DEFLTJCL member. The CHANGE.DBDS command can be used to change this for individual partition DBDSs.

ICJCL(ICJCL | *member*)
Optional keyword that you use to specify the name of a member of a

partitioned data set that contains skeletal JCL. When you issue a GENJCL.IC command, DBRC uses this member to generate the JCL to run the Database Image Copy utility (or the Database Image Copy 2 utility) for the partition DBDS specified on the GENJCL command. All partition DBDSs will be created with this ICJCL member. The CHANGE.DBDS command can be used to change this for individual partition DBDSs.

NOREUSE | REUSE
Mutually exclusive, optional keywords that you use to specify whether the supported image copy utilities are to reuse previously used image copy data sets.

REUSE
REUSE enables the GENJCL.IC command or the GENJCL.OIC command to generate a job that causes the supported image copy utilities to reuse the oldest image copy data set (for the DBDS specified on the GENJCL command) when the GENMAX value for it is exceeded. REUSE requires that you create empty image copy data sets for future use by the supported image copy utilities. In addition you must use an INIT.IC command to record their existence in the RECON data set.

NOREUSE
NOREUSE keyword prohibits such actions. All partition DBDSs will be created with the keyword specified. The CHANGE.DBDS command can be used to change this for individual partition DBDSs.

OICJCL(OICJCL | *member*)
Optional keyword that you use to specify the name of a member of a partitioned data set that contains skeletal JCL. When you issue a GENJCL.OIC command, DBRC uses this member to generate the JCL to run the Online Database Image Copy utility for the partition DBDS specified on the GENJCL command. All partition DBDSs will be created with this OICJCL member. The CHANGE.DBDS command can be used to change this for individual partition DBDSs.

RECOVJCL(RECOVJCL | *member*)
Optional keyword that you use to specify the name of a member of a partitioned data set that contains skeletal JCL. When you issue the GENJCL.RECOV command, DBRC uses this member to generate the JCL to run the Database Recovery utility for the partition DBDS specified on the GENJCL command. All partition DBDSs will be created with this RECOVJCL member. The CHANGE.DBDS command can be used to change this for individual partition DBDSs.

RECOVPD(0 | *value*)
Optional keyword that you use to specify the recovery period for a specified partition DBDS. Specify a numeric value from 0 to 999 that represents the number of days you want DBRC to maintain sufficient recovery-generation information in the RECON data set. The default is 0 which means there is no recovery period. All partition DBDSs will be created with this RECOVPD value. The CHANGE.DBDS command can be used to change this for individual partition DBDSs.

RECVJCL(RECVJCL | *member*)
Optional keyword that you use to specify the name of the skeletal JCL member to be used by the GENJCL.RECEIVE command.

All partition DBDSs will be created with this RECVJCL member. The CHANGE.DBDS command can be used to change this for individual partition DBDSs.

MVDBDS | NOMVDBDS
Optional keyword that you use to allow the partner DBDSs (M-V,Y) to be created in the RECON data set for HALDB databases capable of running HALDB online reorganization.

MVDBDS
Optional parameter that you use to create the M-V, Y data sets if the HALDB is defined as OLRCAP. If the parameter is not included on an INIT.PART command, the M-V, Y data sets are not created until HALDB online reorganization is run the first time.

NOMVDBDS
Optional parameter that you use to indicate that the M-V, Y data sets are not to be created in the RECON data set for HALDB databases defined as OLRCAP. This is the default for HALDB databases that are OLRCAP.

Example of creating a partition

This example creates a partition for the HALDB identified by the DBD keyword. The partition name is identified by the PART keyword. The HALDB uses a partition selection exit. The KEYSTRNG keyword defines the selection string used by the partition selection exit. The string is 149 characters long. The character string must be enclosed in quotation marks because it contains blanks. The ending minus sign ("-") indicates the line is continued. Note that the selection string also contains minus signs, so the first line of the string ends with two minus signs (the first, being part of the selection string, and the second, being the continuation character). The first DSGROUP (DBDS) is assigned a block size of 32 000, the second defaults to 4096, the third is 4000, and the remaining DSGROUPs default to 4096. The partition is marked as needing partition initialization when it is created.

```
//INITPART JOB
.
.
.
//IMS      DD  DSN=IMS.DBDLIB,DISP=SHR
//SYSIN    DD  *
  INIT.PART DBD(DBHDOK01) PART(PDHDOKE) KEYSTRNG('ABCDEFGHIJKL--

MNOPQRSTUVWXYZ0123456789 ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJ-
KLMNOPQRSTUVWXYZ0123456789- ABCD--

EFGHIJKLMNOPQRSTUVWXYZ0123456789') -
  DSNPREFX(TESTSYS1.PROJECT) -
  BLOCKSZE(32000,,4000)
/*
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7..
```

Example using the DBDLIB instead of the default catalog name

```
//INITPART JOB
.
.
.
//IMS      DD  DSN=IMS.DBDLIB,DISP=SHR
//SYSIN    DD  *
  INIT.PART DBD(DBHDOK01) PART(PDHDOKE) KEYSTRNG('ABCDEFGHIJKL--

MNOPQRSTUVWXYZ0123456789 ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJ-
KLMNOPQRSTUVWXYZ0123456789- ABCD--

EFGHIJKLMNOPQRSTUVWXYZ0123456789') -
```

```
  DSNPREFX(TESTSYS1.PROJECT) -
BLOCKSZE(32000,,4000) –
NOCATALG
/*
```

Example using the catalog name GREGCD00 instead of the default catalog name

```
//INITPART JOB
.
.
.
//SYSIN    DD  *
  INIT.PART DBD(DBHDOK01) PART(PDHDOKE) KEYSTRNG('ABCDEFGHIJKL--

MNOPQRSTUVWXYZ0123456789 ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789ABCDEFGHIJ-
KLMNOPQRSTUVWXYZ0123456789- ABCD--

EFGHIJKLMNOPQRSTUVWXYZ0123456789') -
  DSNPREFX(TESTSYS1.PROJECT) -
BLOCKSZE(32000,,4000) –
CATALOG(GREGCD00)
/*
```

Related tasks:

Database design and implementation (Database Administration)

Related reference:

DATASET statements (System Utilities)

INIT.RECON command

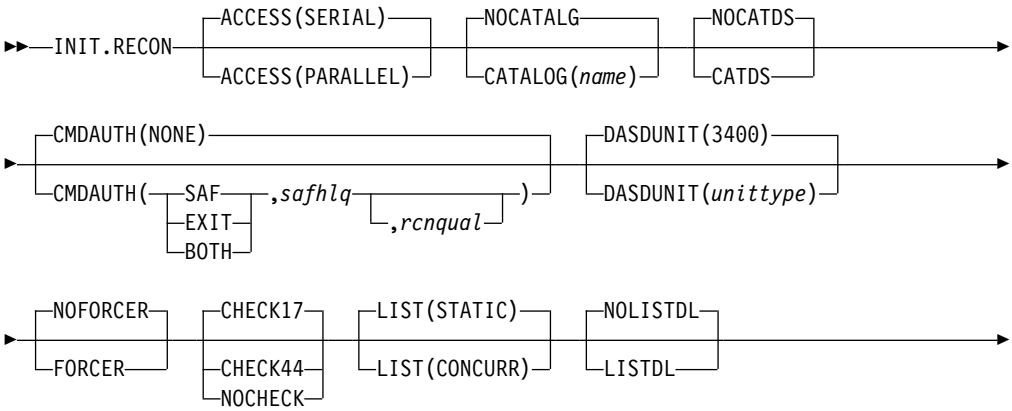
Use the INIT.RECON command to initialize the RECON data set for use by DBRC.

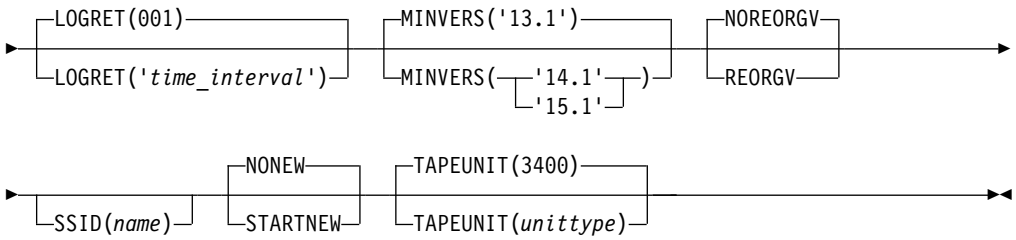
The RECON data sets must first be created using the DFSMS command AMS DEFINE CLUSTER, and must be empty.

Subsections:

- "Syntax"
- "Keywords" on page 312
- "Examples of initializing the RECON data set" on page 318

Syntax





Keywords

ACCESS()
Optional keyword you use to disable or enable parallel RECON access.

SERIAL
Specifies access to a RECON data set in serial mode. Only one DBRC request can be processed at a time. Access to the RECON data set is serialized at the data set level using hardware reserve.

PARALLEL
Specifies access to a RECON data set in parallel mode. With parallel RECON access, DBRC requests from multiple IMS systems, batch jobs, and utility jobs can be processed concurrently. the RECON data set is accessed in transactional VSAM mode. Access is serialized at the VSAM record level using coupling facility locking.

To use parallel RECON access, the following conditions must be met:

- All three RECON data sets must be available.
- The DBRC instance processing the command must register with SCI (that is, an IMSplex name must be specified through the DBRC SCI Registration exit routine (DSPSCIX0) or the IMSPLEX execution keyword).

NOCATALG | CATALOG
Mutually exclusive, optional parameters that you use to indicate whether DBRC retrieves the database definitions from the IMS catalog or the DBDLIB data set when the following commands are run:

CHANGE.DBDS
CHANGE.PART
INIT.DB
INIT.DBDS
INIT.PART
NOTIFY.REORG

For databases that are defined using Database Definition Language (DDL), DBRC must access an IMS catalog.

NOCATALG
Specifies that the DBDLIB data set is used.

CATALOG(*name*)
Specifies that an IMS catalog is used.

name is an eight-character alphanumeric string that identifies an IMS catalog HALDB database that is registered in the RECON data set. Any catalog name that is less than eight-characters is padded with blanks.

The catalog name is not validated when the INIT.RECON command is run because the catalog cannot be defined in the RECON data set until after the RECON data set is initialized.

NOCATDS | CATDS
Mutually exclusive, optional keywords that you use to indicate whether image copy, change accumulation, and log data sets are cataloged.

NOCATDS
Specifies that these data sets, regardless of their cataloged status, are not to be treated as cataloged.

DBRC verifies that the volume serial and file sequence numbers that appear in the job file control block are the same as the information recorded in the RECON data set.

CATDS
Specifies that these data sets are cataloged or SMS-managed.

DBRC does not verify the volume serial and file sequence numbers for the data set.

In order for the CATDS option to be effective, the data set must be cataloged, and volume serial information for the data set must be omitted from the JCL. If the data set is cataloged, CATDS is specified, and volume serial information is included in the JCL, DBRC ignores CATDS and allocates the data set by the JCL. Normal volume serial and file sequence checking occurs.

If the data set is not cataloged, CATDS is not effective, and DBRC allocates the data set by the JCL, with volume serial and file sequence checking.

If log data sets are SMS-managed, select the CATDS option and remove the %LOGVOLS keyword from skeletal JCL member CAJCL.

Attention: The CATDS option affects restart of IMS from SLDS data sets. Because the CATDS option indicates the SLDSs are under the control of a catalog management system, the VOLSER is not passed back to IMS for data set allocation. If the SLDS data sets are not cataloged, IMS restart fails.

CMDAUTH()
Optional keyword you use to specify whether DBRC batch command authorization is to be performed.

NONE | SAF | EXIT | BOTH
Mutually exclusive subparameters that you use to specify your DBRC batch command security option.

NONE
Specifies that command authorization is to be skipped. This is the default.

SAF
Specifies that a security product is to be called for command authorization.

EXIT
Specifies that the DBRC Command Authorization exit (DSPDCAX0) is to be called for command authorization.

BOTH
Specifies that a security product and the DBRC Command

Authorization Exit (DSPDCAX0) are to be called for command authorization. The security product is called first, then DSPDCAX0.

safhlq
High-level qualifier of the resource name defined to the z/OS System Authorization Facility (SAF) product (RACF) to define command authorization. *safhlq* must be specified with SAF, EXIT, or BOTH. *safhlq* is a name up to 8 alphanumeric characters in length.

rcnqual
Optional parameter you use for setting a string value in the RECON data set that will be used in DBRC command processing on the RECON to determine whether the data set is a copy and whether command authorization should be enforced. The *rcnqual* parameter can be specified only with SAF, EXIT, or BOTH.

When setting *rcnqual*, the string must be a substring of the RECON COPY1 data set name in order to prevent inadvertently disabling security for the production RECON data set. If *rcnqual* is specified with an asterisk (*) at the end and within quotation marks, the RECON COPY1 data set name must start with *rcnqual* or DBRC command authorization will not be enforced.

For command authorization to be enforced, the *rcnqual* value must be a substring of the RECON COPY1 data set name. Ideally it should be a substring that is common to all three RECON data set names in order to prevent any problems if RECON reconfiguration takes place. *rcnqual* can be up to 44 characters long and must follow DBRC data set naming conventions.

DASDUNIT()
Optional keyword you use to specify the unit type of the DASD device holding the records for log data sets. The unit type can be up to eight alphanumeric characters long.

If you do not use this keyword to specify a DASD device, then the INIT.RECON command defaults to DASD unit type 3400.

DBRC stores the value specified for DASDUNIT in the unit field of various RECON records. Multiple utilities use this value as the default value for the output device type.

NOFORCER | FORCER
Mutually exclusive, optional keywords that you use to specify whether all IMS databases must be registered in the RECON data set.

NOFORCER
Specifies that databases do not have to be registered.

FORCER
Specifies that all databases must be registered. If you specify FORCER and a job attempts to access an unregistered database, the database authorization call from IMS to DBRC fails.

CHECK17 | CHECK44 | NOCHECK
Mutually exclusive, optional keywords that you use to change the type of log data set name comparison that DBRC does.

CHECK17
Verifies that the last 17 characters of a log data set name are consistent with the RECON data set. If the name in the RECON data set does not match the names on the appropriate ddname, the utility stops.

CHECK44
Verifies that the 44-character log data set name is consistent with the RECON data set. If the name in the RECON data set does not match the name on the appropriate log ddname, the utility stops.

NOCHECK
Used if the data set name specified as input to the Database Recovery utility is longer than 17 characters and has a new high-level qualifier. DBRC does not compare the log data set name recorded in the RECON data set with the name on the appropriate ddname.

LIST()
Optional keyword that you use to set a default of STATIC or CONCURR for LIST commands. The default only applies to LIST commands processed when the RECON data set is in parallel access (or TVS) mode.

If the LIST keyword is not specified on the INIT.RECON command, STATIC is set as the default.

STATIC
Specifies that if the RECON data set is in parallel access mode and a LIST command is processed that did not specify the STATIC or CONCURR keyword, the default for list processing is STATIC.

CONCURR
Specifies that if the RECON data set is in parallel access mode and a LIST command is processed that did not specify the STATIC or CONCURR keyword, the default for list processing is CONCURR.

Recommendation: Because some LIST commands with the STATIC option are likely to be disruptive, set CONCURR as the default. STATIC can then be specified on specific commands, for example, when trying to resolve problems. (The default if the LIST keyword is not specified on INIT.RECON is STATIC rather than CONCURR only because LIST commands with serial RECON access always produce consistent output.)

LOGRET(001 | '*time_interval*')
Optional keyword you use to specify the retention period for log data sets.

The retention period is the minimum amount of time in which a log becomes inactive after it is opened. (It is then eligible to be deleted.)

The *time_interval* is a partial, punctuated time stamp representing a time interval (days, hours, minutes, and seconds) rather than a date and time. The time stamp has the following format:

ddd|*hh*|*mm*|*ss*|*t*

where:

ddd
Number of days (000 to 365)

hh Number of hours (0 to 23)

mm Number of minutes (0 to 59)

ss Number of seconds (0 to 59)

t Tenths of a second (0 to 9)

The punctuation for the time stamp (shown in the above format as a vertical bar (|)) can be any non-numeric character, such as a period (.) or a comma (,). The time stamp must be enclosed in single quotation marks (') if it contains any blanks or special characters. The number of days must include any leading

zeros, but you can omit trailing zeros. Valid intervals range from a tenth of a second to 365 days. The default value, 001, is 24 hours.

Because the time interval is treated as a time stamp, message DSP1048I might be issued for incorrect values. Some examples of valid time intervals include:

```
LOGRET(365)
LOGRET('030 12.00')
LOGRET('000 00:00:08.0')
LOGRET('000 00,00,00,1')
```

Two different valid formats for equivalent time stamp specifications are shown.

```
LOGRET(030)          LOGRET('030')       = 30 days
LOGRET('010 12,30')  LOGRET('010 12:30')  = 10 days, 12 hours, 30 minutes
```

For information about inactive logs and how they are determined by use of the log retention period and other conditions, see the INACTIVE keyword under the DELETE.LOG (for RLDS and SLDS) command.

If you do not use this keyword to specify a retention period, then the INIT.RECON command defaults to a period of 001 (24 hours).

Recommendation: Ensure that an appropriate log retention period is specified so that log information is kept in the RECON data set for a long enough period of time. Certain IMS functions depend on log information in the RECON data set, and these functions might fail if that information is deleted too soon. For example, one or more of the following functions might fail:

- Dynamic backout
- Batch message processing (BMP) extended restart (XRST)
- Unit of work table entry (UOWE) resynchronization

NOLISTDL | LISTDL

Mutually exclusive, optional keywords that you use to specify whether data set names deleted from the RECON (by the DELETE.LOG command or by an archive job log compression) are listed in the job output. The setting specified on this command can be overridden by the DELETE.LOG command. There is no way to override the setting for log compression during an archive job.

LISTDL

Specifies that names of deleted data sets are to be listed in the job output.

NOLISTDL

Specifies that names of deleted data sets are not to be listed in the job output.

MINVERS('13.1' | '14.1' | '15.1')

Optional keyword that specifies the lowest version of IMS that is allowed to sign on to DBRC. The default is MINVERS('13.1').

The format for the MINVERS keyword includes a period. DBRC considers this a special character and requires the value to be enclosed in single quotation marks.

Requirement:

- To run the DEDB Alter utility with the ALTERAREA function for a DEDB area with sequential dependent (SDEP) segments, the **MINVERS** value must be '14.1'.
- To specify that an HALDB database can have up to 8 gigabytes of data in an OSAM PHDAM or PHIDAM database data set, the **MINVERS** value must be '14.1' or it must be '13.1' with a CDSLID value of 2. When **MINVERS** is set to '14.1', it cannot be lowered if an HALDB database is defined as using 8-GB

OSAM data sets. The CHANGE.RECON () fails and the DSP1256E message is issued. A CDSLID value of 2 indicates that the SPE APAR that enables the support of 8-GB OSAM data sets has been applied to all IMS Version 13 resident libraries that access the RECON data set.

NOREORGV | REORGV
Mutually exclusive, optional parameters you use to modify the HALDB partition reorganization number verification. The default is NOREORGV.

NOREORGV
Specifies that HALDB partition reorganization number verification is disabled. Any requests to store, update or verify the reorganization number will be ignored.

REORGV
Specifies that HALDB partition reorganization number verification is enabled. DBRC stores the reorganization number for each partition in the partition record in the RECON data set to prevent the reorganization number from being regressed.

SSID(*name*)
Optional keyword you use to specify the name of the IMS subsystem that is to be used as the default subsystem ID for the following commands:

CHANGE.PRILOG
CHANGE.SECLOG
DELETE.LOG
GENJCL.ARCHIVE
GENJCL.CLOSE
NOTIFY.PRILOG
NOTIFY.SECLOG

name is an 8-character alphanumeric string that identifies a valid IMS subsystem ID.

NONEW | STARTNEW
Mutually exclusive, optional keywords that you use to specify whether new jobs are to be started when only one RECON data set is available.

NONEW
Specifies that new jobs are not to be started.

STARTNEW
Specifies that new jobs are to be started. If parallel RECON access is enabled, new jobs cannot start with a single available RECON data set if another DBRC instance is running with two active RECON data sets.

TAPEUNIT(*unittype*)
Optional keyword you use to specify the unit type of the tape device that is holding the records for log data sets. The unit type can be up to 8 alphanumeric characters long.

If you do not use this keyword to specify a tape device, then the INIT.RECON command defaults to unit type 3400.

DBRC stores the value specified for TAPEUNIT in the unit field of various RECON records. Multiple utilities use this value as the default value for the output device type.

Usage notes

DBRC accesses the database definitions stored in the DBDLIB while processing the NOTFY.REORG, CHANGE.PART, CHANGE.DBDS, INIT.PART, INIT.DB and INIT.DBDS commands. DBRC needs to access the IMS catalog when the directory is enabled instead of the DBDLIB because it may not contain the database definitions for a database defined with DDL.

The RECON will contain a new catalog name. This name will be defined indicating the DBRC is using the directory instead of the DBDLIB for commands that need to reference the database definition. This catalog name can be set on the INIT.RECON command as the default. The catalog name is not required; if it is not present then DBRC will continue to expect the IMS DD to contain the DBDLIB.

The catalog name defined during INIT.RECON is not validated because the catalog cannot be defined to the RECON yet.

Examples of initializing the RECON data set

Here are some examples of using the INIT.RECON command.

Example 1 for INIT.RECON command

```
 //INITRCON JOB
⋮
 //RECON1   DD  DSN=RECON7,DISP=SHR
 //RECON2   DD  DSN=RECON8,DISP=SHR
 //SYSIN    DD   *
      INIT.RECON  NOCHECK SSID(IMSB) LOGRET('007 00:00:30.0')
 /*
```

In this example, the RECON data sets are identified by the RECON1 and RECON2 DD statements.

Example 2 for INIT.RECON command

```
 //INITRCON JOB
⋮
 //SYSIN    DD   *
      INIT.RECON CMDAUTH(SAF,SAFHLQ1,IMSTESTS.DSHR)
 /*
```

In this example, if the RECON COPY1 data set name is IMSTESTS.DSHR.RECON1, DBRC command authorization security will remain enforced for any future access attempts on the RECON data set, because the *rcnqual* value of “IMSTESTS.DSHR” is a substring of the RECON COPY1 data set name IMSTESTS.DSHR.RECON1.

If you copy this RECON data set into a data set whose name has COPYDSHR as the second level qualifier, DBRC command authorization security will be disabled for any future access attempts on the RECON data set, because when the copy is made, the RECON COPY1 data set name is changed to IMSTESTS.COPYDSHR.RECON1, which no longer contains the *rcnqual* string value of “IMSTESTS.DSHR”.

Example 3 for INIT.RECON command

```
//INITRCON JOB
.
.
.
 //RECON1   DD  DSN=ILOVE.IMS.SAIDPAT.RECON1,DISP=SHR
 //RECON2   DD  DSN=ILOVE.IMS.SAIDPAT.RECON2,DISP=SHR
 //SYSIN    DD   *
      INIT.RECON  SSID(GREG) LOGRET('007 00:00:30.0') CATALOG(DFSCD000)
 /*
```

Related reference:

"DELETE.LOG command (for RLDS and SLDS)" on page 169

"DBRC time stamps" on page 52

Chapter 15. LIST commands

Use the LIST commands to produce a formatted printout of all or selected information contained in a RECON data set.

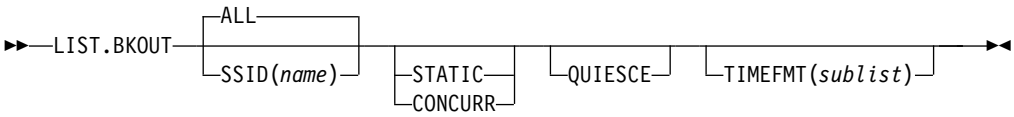
LIST.BKOUT command

Use the LIST.BKOUT command to list information about the backout record for the selected subsystem or to list all backout records in the RECON data set.

Subsections:

- "Syntax"
- "Keywords"
- "Example of listing information about a backout record" on page 322

Syntax



Keywords

ALL | SSID(*name*)
Mutually exclusive, optional keywords that identify the backout records that are to be displayed.

ALL
Specifies that all the backout records in the RECON data set are to be displayed.

SSID(*name*)
Specifies that only one backout record is to be displayed. *name* is an eight-character alphanumeric string that identifies a valid subsystem ID.

QUIESCE
Optional keyword that you use to indicate that, if the RECON access is parallel and the STATIC keyword is specified or defaulted to, the RECON data set is to be quiesced to process the command. QUIESCE is ignored if the CONCURR keyword is specified or defaulted to.

STATIC | CONCURR
Mutually exclusive, optional keywords that you use to control how the list output is produced when parallel RECON access is in effect. If specified when parallel access is not in effect, the keyword is ignored.

If neither STATIC nor CONCURR is specified, the default setting from the RECON header record is used. (The default is set by the INIT.RECON command, by a CHANGE.RECON LIST command, or by upgrade processing.)

STATIC
Specifies that the list output is to be consistent as of a point in time during list processing. The RECON records accessed by list processing cannot be updated by other DBRC instances until list processing has completed.

© Copyright IBM Corp. 1974, 2017

CONCURR
Specifies that list processing can occur concurrent with RECON update activity. In this case, the list output may not reflect the state of the RECON data at any point in time.

TIMEFMT(*sublist*)
Optional keyword that you use to define the form in which time stamps appear in messages, displays, and listings from DBRC.

The LIST commands get the TIMEFMT default from what is specified in the RECON header record.

Example of listing information about a backout record

The following example shows how to use the LIST.BKOUT command to list information about the backout record for subsystem IMS1.

```
//LISTBKOUT  JOB
⋮
//SYSIN      DD  *
     LIST.BKOUT SSID(IMS1)
/*
```

Related reference:

Chapter 18, “Sample listings from the RECON data set,” on page 393

“DBRC time stamps” on page 52

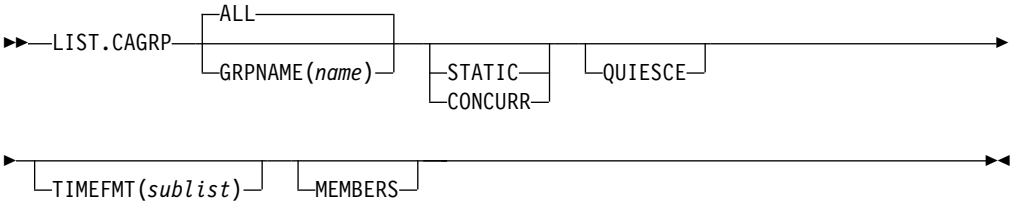
LIST.CAGRP command

Use the LIST.CAGRP command to list information in the Copy1 RECON data set about either a specified CA group or all CA groups.

Subsections:

- “Syntax”
- “Keywords”
- “Example of specifying the CA group and CA records using GRPNAME” on page 323

Syntax



Keywords

ALL | GRPNAME(*name*)
Mutually exclusive, optional keywords you use to specify the name of the CA group for which information is to be displayed.

ALL
Produces a list of the CA group record and corresponding change accumulation run records for each CA group in the RECON data set.

GRPNAME(*name*)
Produces a list of the CA group record and the change accumulation run records for the group that you request in *name*.

MEMBERS
Optional keyword you use to only list information about the CA group definition. CA execution information is not listed.

QUIESCE
Optional keyword to indicate that, if the RECON access is parallel and the STATIC keyword is specified or defaulted to, the RECON data set is quiesced to process the command. QUIESCE is ignored if the CONCURR keyword is specified or defaulted to.

STATIC | CONCURR
Mutually exclusive, optional keywords that you use to control how the list output is produced when parallel RECON access is in effect. If specified when parallel access is not in effect, the keyword is ignored.

If neither STATIC nor CONCURR is specified, the default setting from the RECON header record is used. (The default is set by the INIT.RECON command, by a CHANGE.RECON LIST command, or by upgrade processing.)

STATIC
Specifies that the list output is to be consistent as of a point in time during list processing. The RECON records accessed by list processing cannot be updated by other DBRC instances until list processing has completed.

CONCURR
Specifies that list processing can occur concurrent with RECON update activity. In this case, the list output may not reflect the state of the RECON data at any point in time.

TIMEFMT(*sublist*)
Optional keyword you use to define the form in which time stamps appear in messages, displays, and listings from DBRC. The five values are positional. Each is optional and can be omitted by including only the comma.

The TIMEFMT default for LIST commands is obtained from what is specified in the RECON header record.

Example of specifying the CA group and CA records using GRPNAME

In this example, the CA group record and the associated CA records are specified by the GRPNAME keyword.

```
//LISTCAGP JOB
⋮
//SYSIN    DD   *
     LIST.CAGRP GRPNAME (MYGROUP)
/*
```

Related reference:

Chapter 18, “Sample listings from the RECON data set,” on page 393

“DBRC time stamps” on page 52

LIST.DB command

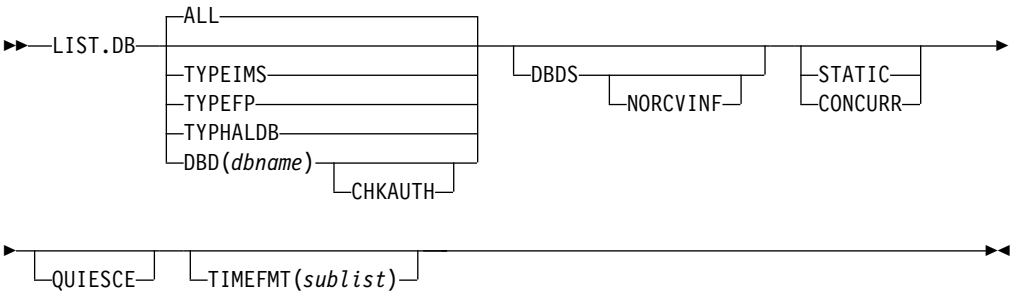
Use the LIST.DB command to receive a list of databases that are registered in the RECON data set. You can list one or all database records, with or without their associated DBDS records.

The LIST.DB command displays the recoverable or nonrecoverable status of the database.

Subsections:

- "Syntax"
- "Keywords"
- "Example of displaying a database without its DBDS records" on page 326

Syntax



Keywords

ALL | TYPEIMS | TYPEFP | TYPHALDB | DBD(*dbname*)
Mutually exclusive, optional keywords you use to specify which databases in the RECON data set are to be displayed.

ALL
Specifies that all database records in the RECON data set are to be displayed. For HALDB databases, the partition database records are listed under the master record.

TYPEIMS
Specifies that all database records in the RECON data set that describe a DL/I database are to be displayed.

TYPEFP
Specifies that all database records in the RECON data set that describe a Fast Path DEDB are to be displayed.

TYPHALDB
Specifies that all database records that represent HALDB databases are to be displayed, including the HALDB master database records (TYPE=HALDB) along with their associated HALDB partition database records (TYPE=PART).

DBD(*dbname*)
Displays a specific database record or recovery group for a database.

Restriction for HALDB databases: For HALDB databases, specifying the HALDB master name lists the HALDB master record and all of its partition database records. Specifying the partition database name lists only the partition database record.

CHKAUTH
Optional keyword you use to check the authorization status of the specified DBD. DBD can be either a full-function database, a Fast Path database, a HALDB master, or a HALDB partition. Nothing will be listed. DSP0044I and RC=04 will be received if the DBD is authorized to a subsystem. Only RC=00 will be received if the DBD is not authorized.

Only the DBD keyword is valid with the CHKAUTH keyword. If DBD is not specified, the command fails. If any other keywords are specified, they are ignored.

DBDS
Optional keyword you use to display those DBDSs or areas in the RECON data set that are associated with the specified database. DBDS information includes recovery-related records (ALLOC, IC, RECOV, REORG). If you do not specify this keyword, no DBDS records or area records are displayed.

NORCVINF
Suppresses recovery-related records (ALLOC, IC, RECOV, REORG) for those DBDSs or areas in the RECON data set that are associated with the specified database.

If the LIST.DB command output is truncated and message DSP0057I is returned, you can specify the NORCVINF keyword to minimize the size of the output.

QUIESCE
Optional keyword to indicate that, if the RECON access is parallel and the STATIC keyword is specified or defaulted to, the RECON data set is to be quiesced to process the command. QUIESCE is ignored if the CONCURR keyword is specified or defaulted to.

STATIC | CONCURR
Mutually exclusive, optional keywords that you use to control how the list output is produced when parallel RECON access is in effect. If specified when parallel access is not in effect, the keyword is ignored.

If neither STATIC nor CONCURR is specified, the default setting from the RECON header record is used. (The default is set by the INIT.RECON command, by a CHANGE.RECON LIST command, or by upgrade processing.)

STATIC
Specifies that the list output is to be consistent as of a point in time during list processing. The RECON records accessed by list processing cannot be updated by other DBRC instances until list processing has completed.

CONCURR
Specifies that list processing can occur concurrent with RECON update activity. In this case, the list output might not reflect the state of the RECON data at any point in time.

TIMEFMT(*sublist*)
Optional keyword you use to define the form in which time stamps appear in messages, displays, and listings from DBRC. The five values are positional. Each is optional and can be omitted by including only the comma.

The TIMEFMT default for LIST commands is obtained from what is specified in the RECON header record.

Example of displaying a database without its DBDS records

In this example, database DEDBAB12, without its associated DBDS records, is displayed.

Input:

```
 //LISTDB
⋮
 //SYSIN    DD   *
      LIST.DB DBD(DEDBAB12)
 /*
```

Response:

```
LIST.DB DBD(DEDBAB12)

-------------------------------------------------------------------------
 DB
  DBD=DEDBAB12                              DMB#=8       TYPE=FP
  SHARE LEVEL=3
  FLAGS:                                COUNTERS:
                                          RECOVERY NEEDED COUNT   =3
                                          IMAGE COPY NEEDED COUNT =0
    PROHIBIT AUTHORIZATION=OFF            AUTHORIZED AREAS        =1
    RECOVERABLE           =YES            EEQE COUNT              =0
    FULLSEG DEFAULT       =NO             ALTER COUNT             =1
                                          ALTER COMPLETE COUNT    =1
-------------------------------------------------------------------------
```

Related concepts:

Overview of recovery of databases (Database Administration)

Related reference:

Chapter 18, “Sample listings from the RECON data set,” on page 393

“DBRC time stamps” on page 52

/RMxxxxxx commands (Commands)

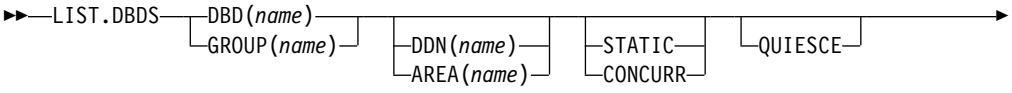
LIST.DBDS command

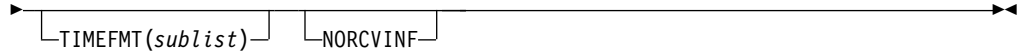
Use the LIST.DBDS command to display a list of all records in the RECON data set that contain information about a specific DBDS or DEDB area.

Subsections:

- “Syntax”
- “Keywords” on page 327
- “Example of displaying a list of records” on page 328

Syntax





Keywords

DBD(*name*) | GROUP(*name*)
Mutually exclusive, required keywords you use to identify the DBDS or DEDB area being listed.

DBD(*name*)
Specifies the database name of the DBDS or DEDB area being displayed.

Restrictions for HALDB databases: For HALDB databases, you can specify either a HALDB master database name or a HALDB partition name when you do not specify the DDN keyword. When you specify the DDN keyword, you can specify only a HALDB partition name.

GROUP(*name*)
Specifies that all DBDSs or DEDB areas of the named DBDS group are to be displayed. If GROUP is specified, the LIST.DBDS command is executed for each member of the identified group.

DDN(*name*) | AREA(*name*)
Mutually exclusive, optional keywords you use to identify the DBDS or DEDB area to be displayed. You specify one of these keywords only when you specify the DBD keyword.

DDN(*name*)
Specifies the name of the DBDS to display.

Restrictions for HALDB databases: For HALDB databases, you must specify a HALDB partition name (not a HALDB master name) with the DBD keyword in order to use the DDN keyword. The DDN keyword value is the HALDB partition DDN. The LIST.DBDS command is performed for the identified DBDS of the partition. The LIST.DBDS command fails if DDN does not identify a DBDS in the partition.

AREA(*name*)
Specifies the name of the DEDB area to display.

If neither DDN nor AREA is specified, the LIST.DBDS command is executed for each DBDS or DEDB area of the specified database.

Restrictions for HALDB databases: If you specify a HALDB master name, the LIST.DBDS command is performed for each DBDS for each HALDB partition in the HALDB master. If you specify a HALDB partition name, the LIST.DBDS command is performed for each DBDS of the identified partition.

NORCVINF
Suppresses recovery-related records (ALLOC, IC, RECOV, REORG) for all of the displayed DBDSs and DEDB areas. If you do not specify this keyword, you get a list of all records in the RECON data set that contain information about the displayed DBDSs and DEDB areas.

If the LIST.DBDS command output is truncated and message DSP0057I is returned, you can specify the NORCVINF keyword to minimize the size of the output.

QUIESCE
Optional keyword to indicate that, if the RECON access is parallel and the

STATIC keyword is specified or defaulted to, the RECON data set is to be quiesced to process the command. QUIESCE is ignored if the CONCURR keyword is specified or defaulted to.

STATIC | CONCURR
Mutually exclusive, optional keywords that you use to control how the list output is produced when parallel RECON access is in effect. If specified when parallel access is not in effect, the keyword is ignored.

If neither STATIC nor CONCURR is specified, the default setting from the RECON header record is used. (The default is set by the INIT.RECON command, by a CHANGE.RECON LIST command, or by upgrade processing.)

STATIC
Specifies that the list output is to be consistent as of a point in time during list processing. The RECON records accessed by list processing cannot be updated by other DBRC instances until list processing has completed.

CONCURR
Specifies that list processing can occur concurrent with RECON update activity. In this case, the list output might not reflect the state of the RECON data at any point in time.

TIMEFMT(*sublist*)
Optional keyword you use to define the form in which time stamps appear in messages, displays, and listings from DBRC. The five values are positional. Each is optional and can be omitted by including only the comma.

The TIMEFMT default for LIST commands is obtained from what is specified in the RECON header record.

Example of displaying a list of records

In this example, a list of all records in the RECON data set that contain information about DEDB area AREA01 is displayed.

Input:

```
 //LISTDBDS JOB
⋮
 //SYSIN    DD   *
      LIST.DBDS DBD(FPEDBD02) AREA(AREA01)
 /*
```

Response:

```
LIST.DBDS DBD(FPEDBD02) AREA(AREA01)
  ---------------------------------------------------------------------
   DBDS
    DBD=FPEDBD02  AREA=AREA01                                  TYPE=FP
    SHARE LEVEL=3             DSID=00001 DBORG=DEDB   DSORG=VSAM
  GSNAME=**NULL**              USID=0000000002
 AUTHORIZED USID=0000000002  RECEIVE USID=0000000002 HARD USID=0000000002
 RECEIVE NEEDED USID=0000000000
 CAGRP=**NULL**  GENMAX=2     IC AVAIL=0     IC USED=0     DSSN=00000001
 NOREUSE         RECOVPD=0      NOVSO  NOPREOPEN  NOPRELOAD  NOFULLSG
 CFSTR1=**NULL**         CFSTR2=**NULL**          NOLKASID  NOMAS
 DEFLTJCL=**NULL**  ICJCL=ICJCL    RECVJCL=ICRCVJCL  RECOVJCL=RECOVJCL
 DBRCVGRP=**NULL**
 FLAGS:                            COUNTERS:
   PROHIBIT AUTHORIZATION=OFF        AUTHORIZED SUBSYSTEMS   =1
                                     HELD AUTHORIZATION STATE=6
   IC NEEDED             =OFF        ADS AVAIL #             =1
```

```
  IC RECOMMENDED        =ON
  RECOV NEEDED          =OFF        REGISTERED ADS #        =1
                                    EEQE COUNT              =0
  RECEIVE NEEDED        =OFF        SHADOW ADS AVAIL #      =2
  OFR REQUIRED          =NO         REGISTERED SHADOW ADS # =2
  TRACKING SUSPENDED    =NO
  HSSP CIC IN PROGRESS  =NO
  QUIESCE IN PROGRESS   =NO
  QUIESCE HELD          =NO
  ALTER IN PROGRESS     =YES

ADS LIST:
                                                          CREATE
  -ADS DDN--ADS DSN-                                   -STAT- -RUNNING-
  DB21AR0  IMSTESTS.DB21AR0                            AVAIL     NO

SHADOW ADS LIST:
                                                          CREATE
  -ADS DDN--ADS DSN-                                   -STAT- -RUNNING-
  DB21AS01 IMSTESTS.DB21AR0.SHADOW                     AVAIL     NO
  DB21AS02 IMSTESTS.DB21AR0.SHADOW.IC                  AVAIL     NO  IC

ASSOCIATED SUBSYSTEM INFORMATION:
                            ENCODED
   -SSID-   -ACCESS INTENT-  -STATE-  -SS ROLE-
  IMS1         UPDATE          6      ACTIVE
```

Related reference:

Chapter 18, “Sample listings from the RECON data set,” on page 393

“DBRC time stamps” on page 52

/RMxxxxxx commands (Commands)

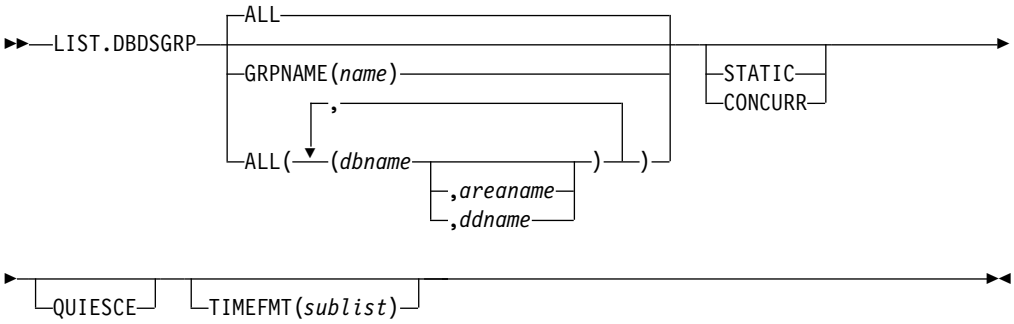
LIST.DBDSGRP command

Use the LIST.DBDSGRP command to display a list of all three kinds of data group records (DB groups, DBDS groups, and recovery groups) in the RECON data set, a list of the members of a single data group record, or a list of all data group records containing a specified member or members.

Subsections:

- “Syntax”
- “Keywords” on page 330
- “Example of displaying the members of a DBDS group” on page 330

Syntax



Keywords

ALL | GRPNAME(*name*) | ALL(*dbname,areaname | ddname*)
Mutually exclusive, optional keywords you use to specify the groups to be listed.

ALL
Produces a list of all the DBDS, DB, and recovery groups identified in the RECON data set. All is the default.

GRPNAME
Produces a list of the members of the specified group. The named group must exist in the RECON data set.

ALL(*dbname,areaname,ddname*)
Produces a limited list of the DBDS, DB, and recovery groups identified in the RECON data set. Use the keywords (*dbname,areaname,ddname*) to limit the number of records listed. A group is listed only if it contains one or more of the databases, DBDSs, or areas specified.

Restriction for HALDB databases: For the LIST.DBDSGRP command, *dbname* can specify either the name of a HALDB partition or a HALDB master database.

QUIESCE
Optional keyword to indicate that, if the RECON access is parallel and the STATIC keyword is specified or defaulted to, the RECON data set is to be quiesced to process the command. QUIESCE is ignored if the CONCURR keyword is specified or defaulted to.

STATIC | CONCURR
Mutually exclusive, optional keywords that you use to control how the list output is produced when parallel RECON access is in effect. If specified when parallel access is not in effect, the keyword is ignored.

If neither STATIC nor CONCURR is specified, the default setting from the RECON header record is used. (The default is set by the INIT.RECON command, by the CHANGE.RECON command, or by upgrade processing.)

STATIC
Specifies that the list output is to be consistent as of a point in time during list processing. The RECON records accessed by list processing cannot be updated by other DBRC instances until list processing has completed.

CONCURR
Specifies that list processing can occur concurrent with RECON update activity. In this case, the list output may not reflect the state of the RECON data at any point in time.

TIMEFMT(*sublist*)
Optional keyword you use to define the form in which time stamps appear in messages, displays, and listings from DBRC. The five values are positional. Each is optional and can be omitted by including only the comma.

The TIMEFMT default for LIST commands is obtained from what is specified in the RECON header record.

Example of displaying the members of a DBDS group

In this example, the members of a specified DBDS group are displayed.

```
//LISTDBGP   JOB
⋮
//SYSIN      DD  *
     LIST.DBDSGRP   GRPNAME(DBDSG1)
/*
```

Related reference:

"DBRC time stamps" on page 52

LIST.HISTORY command

Use the LIST.HISTORY command to produce a history-of-activity listing for DBDSs or DEDB areas.

The output of the LIST.HISTORY command consists of:

- The database record listing.
- The DBDS record listing. (For a DEDB area, the area authorization and area recovery records are combined to form a DBDS record listing.)
- The following records (if present), listed in time sequence order under each DBDS or area:
  - ALLOC records.
  - IMAGE records.
  - CA execution records, showing the purge time only for the current DBDS or area.
  - RECOV records.
  - REORG records.
- The PRILOG records associated with all ALLOC records listed.
- A timeline summary that interrelates all the events represented by the records listed above.

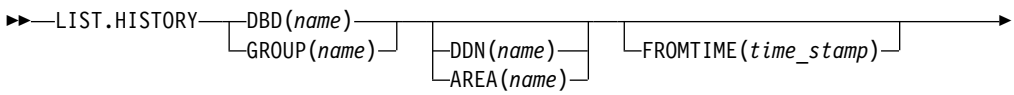
You can use the FROMTIME and TOTIME keywords to define a time range that excludes these records:

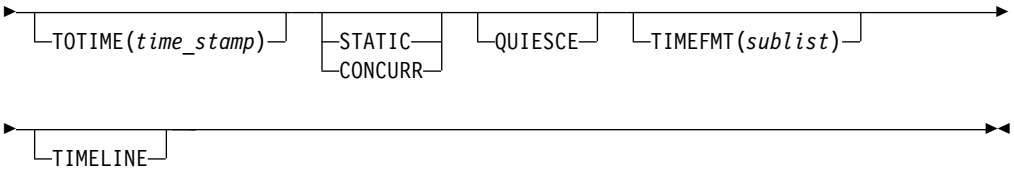
- ALLOC records for USIDs that are not active within the range. If any ALLOC record is active within the time range, all ALLOCs for the same USID are listed.
- IMAGE records with RUN times (or for CICs, an effective purge time) outside the range.
- CA execution records with STOP and PURGE times outside the range.
- RECOV records with RUN and RECOV TO times outside the range.
- REORG records with RUN times outside the range.

Subsections:

- "Syntax"
- "Keywords" on page 332
- "Example of displaying the activity history of a DBDS" on page 334

Syntax





Keywords

DBD(*name*) | GROUP(*name*)
Mutually exclusive, required keywords you use to identify the DBDS or DEDB area to list.

DBD(*name*)
Specifies the database name of the DBDS or DEDB area to be listed.

Restrictions for HALDB databases: For HALDB databases, you can specify either a HALDB master database name or a HALDB partition name when you do not specify the DDN keyword. When you specify the DDN keyword, you can only specify a HALDB partition name.

GROUP(*name*)
Specifies that all the DBDSs or DEDB areas of a DBDS group or CA group are to be listed. If GROUP is specified, the LIST.HISTORY command is executed for each member of the identified group.

DDN(*name*) | AREA(*name*)
Mutually exclusive, optional keywords you use to identify the DBDS or DEDB area to be listed. You specify one of these keywords only when you specify the DBD keyword.

DDN(*name*)
Specifies the name of the DBDS to list.

Restrictions for HALDB databases: For HALDB databases, you must specify a partition database name with the DBD keyword in order to use the DDN keyword. The DDN keyword value is the partition DDN. The LIST.HISTORY command is performed for the identified DBDS of the partition. The LIST.HISTORY command fails if DDN does not identify a DBDS in the partition.

AREA(*name*)
Specifies the name of the DEDB area to list.

If neither DDN nor AREA is specified, the LIST.HISTORY command is executed for each DBDS or DEDB area of the specified database.

Restrictions for HALDB databases: If you specify a HALDB master name, the LIST.HISTORY command is performed for each DBDS for each partition in the HALDB master. If you specify a HALDB partition name, the LIST.HISTORY command is performed for each DBDS of the identified partition.

FROMTIME(*time_stamp*)
Optional keyword you use to specify the time stamps of the DBDS or DEDB area records that are to be listed in time sequence order. The time stamp must be in standard form. Those records that are not listed in time-sequence order are listed regardless of whether FROMTIME or TOTIME are specified.

FROMTIME specifies the time stamp of the oldest record to be listed. If you specify only FROMTIME, all subsequent, pertinent records in the RECON data set are listed.

You can combine the FROMTIME and TOTIME keywords in order to specify a range of records to display.

If you specify neither FROMTIME nor TOTIME, all the records that exist in the RECON data set for the specified DBDSs or DEDB areas are listed.

QUIESCE
Optional keyword to indicate that, if the RECON access is parallel and the STATIC keyword is specified or defaulted to, the RECON data set is to be quiesced to process the command. QUIESCE is ignored if the CONCURR keyword is specified or defaulted to.

STATIC | CONCURR
Mutually exclusive, optional keywords that you use to control how the list output is produced when parallel RECON access is in effect. If specified when parallel access is not in effect, the keyword is ignored.

If neither STATIC nor CONCURR is specified, the default setting from the RECON header record is used. (The default is set by the INIT.RECON command, by a CHANGE.RECON LIST command, or by upgrade processing.)

STATIC
Specifies that the list output is to be consistent as of a point in time during list processing. The RECON records accessed by list processing cannot be updated by other DBRC instances until list processing has completed.

CONCURR
Specifies that list processing can occur concurrent with RECON update activity. In this case, the list output may not reflect the state of the RECON data at any point in time.

TOTIME(*time_stamp*)
Optional keyword you use to specify the time stamps of the DBDS or DEDB area records to be listed in time-sequence order. The time stamp must be in standard form. Those records not listed in time-sequence order are listed regardless of whether FROMTIME or TOTIME are specified. TOTIME specifies the time stamp of the last record to be listed. If you specify only TOTIME, that record plus all prior, pertinent records in the RECON data set are listed.

You can combine the FROMTIME and TOTIME keywords in order to specify a range of records to display.

If you specify neither FROMTIME nor TOTIME, all the records that exist in the RECON data set for the specified DBDSs or DEDB areas are listed.

TIMEFMT(*sublist*)
Optional keyword you use to define the form in which time stamps appear in messages, displays, and listings from DBRC. The five values are positional. Each is optional and can be omitted by including only the comma.

The TIMEFMT default for LIST commands is obtained from what is specified in the RECON header record.

TIMELINE
Optional keyword you use to specify that only the timeline summary should be listed in the output.

Example of displaying the activity history of a DBDS

In this example, the activity history of a specified DBDS is displayed.

```
//LISTHIST    JOB
⋮
//SYSIN       DD  *
     LIST.HISTORY   DBD(DB1)  DDN(NAME1)
/*
```

Related reference:

"DBRC time stamps" on page 52

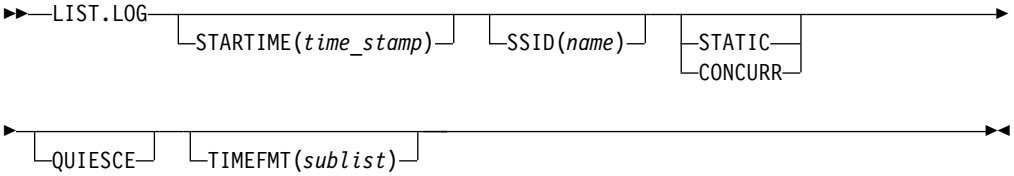
LIST.LOG command (for a PRILOG family)

Use the LIST.LOG command to display the PRILOG record, as well as any LOGALL, SECLOG, PRISLD, or SECSLD records that have the specified start time.

Subsections:

- "Syntax"
- "Keywords"
- "Example of listing a PRILOG family of records" on page 335

Syntax



Keywords

STARTIME(*time_stamp*)
Optional keyword you use to specify the start time of the records that you want displayed.

SSID(*name*)
Optional keyword that limits the display of log records or OLDS entries to those associated with the specified subsystem.

QUIESCE
Optional keyword to indicate that, if the RECON access is parallel and the STATIC keyword is specified or defaulted to, the RECON data set is to be quiesced to process the command. QUIESCE is ignored if the CONCURR keyword is specified or defaulted to.

STATIC | CONCURR
Mutually exclusive, optional keywords that you use to control how the list output is produced when parallel RECON access is in effect. If specified when parallel access is not in effect, the keyword is ignored.

If neither STATIC nor CONCURR is specified, the default setting from the RECON header record is used. (The default is set by the INIT.RECON command, by a CHANGE.RECON LIST command, or by upgrade processing.)

STATIC
Specifies that the list output is to be consistent as of a point in time during

list processing. The RECON records accessed by list processing cannot be updated by other DBRC instances until list processing has completed.

CONCURR
Specifies that list processing can occur concurrent with RECON update activity. In this case, the list output may not reflect the state of the RECON data at any point in time.

TIMEFMT(*sublist*)
Optional keyword you use to define the form in which time stamps appear in messages, displays, and listings from DBRC. The five values are positional. Each is optional and can be omitted by including only the comma.

The TIMEFMT default for LIST commands is obtained from what is specified in the RECON header record.

Attention: If the ERROR, OPEN, or UNARCH keywords are coded, message DSP0141I is issued and the command fails.

If the FROMTIME, TOTIME, INTERIM, or TRACKING keywords are coded, STARTIME is ignored and the command defaults to ALL processing as described under the next section.

Example of listing a PRILOG family of records

In the example, all PRILOG, SECLOG, PRISLD, SECSLD, and LOGALL records that have the specified start time are listed.

```
//LISTLOG  JOB
⋮
//SYSIN DD *
LIST.LOG STARTIME('07.023 12:12:12.123456 PST')
/*
```

Related reference:

"DBRC time stamps" on page 52

LIST.LOG command (for a category of records)

Use the LIST.LOG command to display the PRILOG record, as well as any LOGALL, SECLOG, PRISLD, or SECSLD records that have the specified start time. The syntax of this command is different depending on the list of the records it displays.

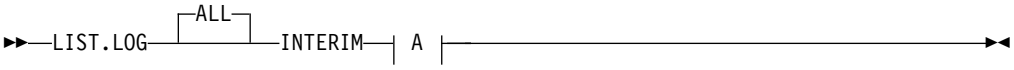
Subsections:

- "Syntax"
- "Keywords" on page 336
- "Examples" on page 338

Syntax

Syntax for PRILOG, LOGALL, SECLOG, PRISLD, SECSLD, PRIOLD, and SECOLD records

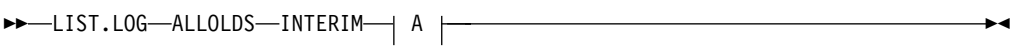
Syntax for IPRI, ISEC, IPRISL, ISECSL, IPRIOL, and ISECOL records



Syntax for PRIOLD and SECOLD records

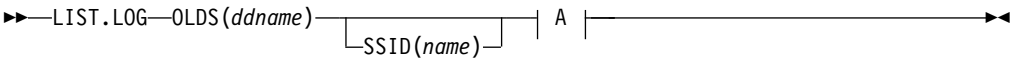
- PRIOLD
- SECOLD

Syntax for IPRIOL and ISECOL records



- IPRIOL
- ISECOL

Syntax for PRIOLD and SECOLD records with matching DD names and subsystem names

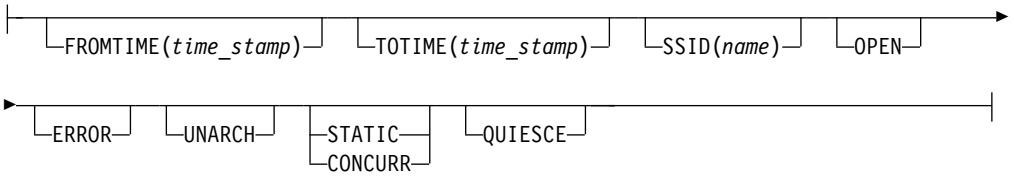


LIST.LOG OLDS displays only the data set entries with matching DD names and subsystem names. If SSID is omitted, processing is the same as for ALLOLDS.

The LIST.LOG command can be further qualified with one or more of the following optional keywords. For example, combining SSID and OPEN limits the display to logs and OLDS entries that belong to a specified subsystem and to those that are not closed.

The optional keywords for LIST.LOG ALL, ALLOLDS, and OLDS are:

A:



Keywords

FROMTIME(*time_stamp*)
Optional keyword that limits the display to log records or OLDS entries starting at, or after, this time. The time stamp must be in standard form.

FROMTIME specifies the time stamp of the oldest record to be listed. If you specify only FROMTIME, all subsequent, pertinent records in the RECON data set are listed.

You can specify both FROMTIME and TOTIME keywords in order to specify a range of records to display.

For PRILOG, SECLOG, PRISLD, SECSLD, IPRI, ISEC, IPRISL, ISECSL records, the FROMTIME time stamp is compared against the time the subsystem started (START =). This start time has an asterisk (*) beside it when the record is listed.

For PRIOLD, SECOLD, IPRIOL, and ISECOL records, the FROMTIME time stamp is compared against the time the subsystem started (PRILOG TIME =).

QUIESCE
Optional keyword to indicate that, if the RECON access is parallel and the STATIC keyword is specified or defaulted to, the RECON data set is to be quiesced to process the command. QUIESCE is ignored if the CONCURR keyword is specified or defaulted to.

STATIC | CONCURR
Mutually exclusive, optional keywords that you use to control how the list output is produced when parallel RECON access is in effect. If specified when parallel access is not in effect, the keyword is ignored.

If neither STATIC nor CONCURR is specified, the default setting from the RECON header record is used. (The default is set by the INIT.RECON command, by a CHANGE.RECON LIST command, or by upgrade processing.)

STATIC
Specifies that the list output is to be consistent as of a point in time during list processing. The RECON records accessed by list processing cannot be updated by other DBRC instances until list processing has completed.

CONCURR
Specifies that list processing can occur concurrent with RECON update activity. In this case, the list output may not reflect the state of the RECON data at any point in time.

TOTIME(*time_stamp*)
Optional keyword that limits the display to log records or OLDS entries starting at, or before, this time. The time stamp must be in standard form.

TOTIME specifies the time stamp of the last record to be listed. If you specify only TOTIME, that record plus all prior pertinent records in the RECON data set are listed.

You can specify both FROMTIME and TOTIME keywords in order to specify a range of records to display.

For PRILOG, SECLOG, PRISLD, SECSLD, IPRI, ISEC, IPRISL, ISECSL records, the TOTIME time stamp is compared against the time the subsystem started (START =). This start time has an asterisk (*) beside it when the record is listed.

For PRIOLD, SECOLD, IPRIOL, and ISECOL records, the TOTIME time stamp is compared against the time the subsystem started (PRILOG TIME =).

SSID(*name*)
Optional keyword that limits the display to log records or OLDS entries associated with the specified subsystem.

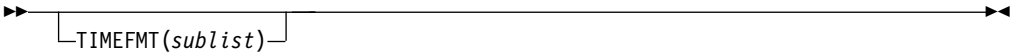
OPEN
Optional keyword that limits the display to log records or OLDS entries that are not closed.

ERROR
Optional keyword that limits the display to log records having one or more data set entries marked in error, and to OLDS entries marked in error.

UNARCH
Optional keyword that limits the display of OLDS entries to those that are not archived.

Note: Specifying the UNARCH keyword without the ERROR or OPEN keywords causes the LIST.LOG ALL command to be processed like the LIST.LOG ALLOLDS command; that is, no log records are listed, only unarchived OLDS entries are listed.

The following keyword can be used on all forms of the command.



TIMEFMT(*sublist*)
Optional keyword you use to define the form in which time stamps appear in messages, displays, and listings from DBRC. The five values are positional. Each is optional and can be omitted by including only the comma.

The TIMEFMT default for LIST commands is obtained from what is specified in the RECON header record.

Examples

Example of displaying RECON records specified by STARTIME

In this example, the RLDSs, SLDS, and corresponding LOGALL record with the time stamp specified in the STARTIME keyword are to be displayed.

```
//LISTRCON JOB
⋮
//SYSIN    DD   *
    LIST.LOG  STARTIME(07031131313023456)
/*
```

Example of displaying a subsystem's OLDS records

In this example, the OLDS records from subsystem IMSA are to be displayed.

```
//LISTRCON JOB
⋮
//SYSIN    DD   *
    LIST.LOG  ALLOLDS SSID(IMSA)
/*
```

Related reference:
"DBRC time stamps" on page 52

LIST.RECON command

Use the LIST.RECON command to display the current status of the RECON data sets.

RECON status information includes the following items:

- The name of the IMS catalog that DBRC commands use for database definitions or CATALOG=**NULL** if DBRC commands use a DBD library for database definitions.
- The contents of the Time Zone Label Table
- The TIMEZIN and TIMEFMT settings
- The minimum IMS release level to coexist with this RECON
- The status of each of the three RECON data sets as follows:

  RECON status
  Meaning

  COPY1
  PRIMARY ACTIVE RECON

  COPY2
  SECONDARY ACTIVE RECON

  SPARE
  AVAILABLE RECON

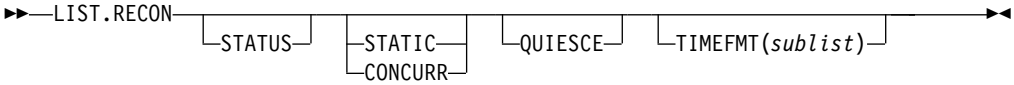
  UNAVAILABLE
  UNAVAILABLE RECON

  DISCARDED
  UNUSABLE RECON

The RECON data set is unavailable when the resource is allocated in another system.

Subsections:
- "Syntax"
- "Keywords"
- "Examples" on page 340

Syntax



Keywords

STATUS
Optional keyword you use to request the RECON header record information and the status of all RECON data sets. If you specify this keyword the listing of the remainder of the records is suppressed.

QUIESCE
Optional keyword to indicate that, if the RECON access is parallel and STATIC is specified or defaulted to, the RECON data set is to be quiesced to process the command. QUIESCE is ignored if the CONCURR keyword is specified or defaulted to.

STATIC | CONCURR
Mutually exclusive, optional keywords that you use to control how the list output is produced when parallel RECON access is in effect. If specified when parallel access is not in effect, the keyword is ignored.

If neither STATIC nor CONCURR is specified, the default setting from the RECON header record is used. (The default is set by the INIT.RECON command, by a CHANGE.RECON LIST command, or by upgrade processing.)

STATIC
Specifies that the list output is to be consistent as of a point in time during list processing. The RECON records accessed by list processing cannot be updated by other DBRC instances until list processing has completed.

CONCURR
Specifies that list processing can occur concurrent with RECON update activity. In this case, the list output may not reflect the state of the RECON data at any point in time.

TIMEFMT(*sublist*)
Optional keyword you use to define the form in which time stamps appear in messages, displays, and listings from DBRC. The five values are positional. Each is optional and can be omitted by including only the comma.

The TIMEFMT default for LIST commands is obtained from what is specified in the RECON header record.

Usage notes

DBRC accesses the database definitions stored in the DBDLIB while processing the NOTIFY.REORG, CHANGE.PART, CHANGE.DBDS, INIT.PART, INIT.DB and INIT.DBDS commands. DBRC needs to access the IMS catalog when the directory is enabled, instead of the DBDLIB, because it may not contain the database definitions for a database defined with DDL.

You will need to define a catalog name indicating that DBRC is using the directory instead of the DBDLIB for commands that need to reference the database definition. The catalog name is displayed on the LIST.RECON command.

Examples

Here are some examples of using the LIST.RECON command.

Example of displaying the RECON data sets

In this example, the status and contents of the RECON data set are displayed.

```
//LISTRCON JOB
.
.
.
//SYSIN    DD   *
     LIST.RECON
/*
```

Example of displaying RECON header and status information

In this example, only RECON header status information is displayed.

```
//LISTRCON JOB
.
.
.
//SYSIN    DD   *
   LIST.RECON STATUS
/*
```

Only the first segment of output shown in the Chapter 18, "Sample listings from the RECON data set," on page 393 is produced in this case.

Example using the DBDLIB

```
LIST.RECON STATUS
15.318 10:47:08.595032              LISTING OF RECON                  PAGE 0002
-------------------------------------------------------------------------------
 RECON
  RECOVERY CONTROL DATA SET, IMS V14R1
  DMB#=9                              INIT TOKEN=15318F1857505F
  NOFORCER  LOG DSN CHECK=CHECK17    STARTNEW=NO
  TAPE UNIT=3400      DASD UNIT=SYSDA     TRACEOFF   SSID=**NULL**
  LIST DLOG=YES                CA/IC/LOG DATA SETS CATALOGED=NO
  MINIMUM VERSION = 12.1       CROSS DBRC SERVICE LEVEL ID= 00001
  REORG NUMBER VERIFICATION=NO
  LOG RETENTION PERIOD=00.000 00:15:00.0
  COMMAND AUTH=NONE  HLQ=**NULL**
RCNQUAL=**NULL**
CATALOG=**NULL**
  ACCESS=SERIAL      LIST=STATIC
  SIZALERT DSNUM=15      VOLNUM=16     PERCENT= 95
  LOGALERT DSNUM=3       VOLNUM=16

  TIME STAMP INFORMATION:

    TIMEZIN = %SYS             -LABEL- -OFFSET-
                                PDT     -07:00
                                PST     -08:00

    OUTPUT FORMAT:  DEFAULT = LOCORG NONE   PUNC YY
                    CURRENT = LOCORG NONE   PUNC YY


  IMSPLEX = ** NONE **    GROUP ID = ** NONE **

 -DDNAME-      -STATUS-       -DATA SET NAME-
  RECON1        COPY1          ILOVE.IMS.SAIDPAT.RECON1
  RECON2        COPY2          ILOVE.IMS.SAIDPAT.RECON2
  RECON3        SPARE          ILOVE.IMS.SAIDPAT.RECON3

  NUMBER OF REGISTERED DATABASES =        9
```

Example using the default catalog name PATSCD00

```
LIST.RECON STATUS
15.318 10:47:08.595032              LISTING OF RECON                  PAGE 0002
-------------------------------------------------------------------------------
 RECON
  RECOVERY CONTROL DATA SET, IMS V14R1
  DMB#=9                              INIT TOKEN=15318F1857505F
  NOFORCER  LOG DSN CHECK=CHECK17    STARTNEW=NO
  TAPE UNIT=3400      DASD UNIT=SYSDA     TRACEOFF   SSID=**NULL**
  LIST DLOG=YES                CA/IC/LOG DATA SETS CATALOGED=NO
  MINIMUM VERSION = 12.1       CROSS DBRC SERVICE LEVEL ID= 00001
  REORG NUMBER VERIFICATION=NO
  LOG RETENTION PERIOD=00.000 00:15:00.0
  COMMAND AUTH=NONE  HLQ=**NULL**
RCNQUAL=**NULL**
CATALOG=PATSCD00
  ACCESS=SERIAL      LIST=STATIC
  SIZALERT DSNUM=15      VOLNUM=16     PERCENT= 95
  LOGALERT DSNUM=3       VOLNUM=16

  TIME STAMP INFORMATION:

    TIMEZIN = %SYS             -LABEL- -OFFSET-
```

```
                                PDT     -07:00
                                PST     -08:00

   OUTPUT FORMAT:  DEFAULT = LOCORG NONE   PUNC YY
                   CURRENT = LOCORG NONE   PUNC YY


  IMSPLEX = ** NONE **    GROUP ID = ** NONE **

 -DDNAME-      -STATUS-       -DATA SET NAME-
  RECON1        COPY1          ILOVE.IMS.SAIDPAT.RECON1
  RECON2        COPY2          ILOVE.IMS.SAIDPAT.RECON2
  RECON3        SPARE          ILOVE.IMS.SAIDPAT.RECON3

  NUMBER OF REGISTERED DATABASES =        9
```

Related reference:

Chapter 18, “Sample listings from the RECON data set,” on page 393

“DBRC time stamps” on page 52

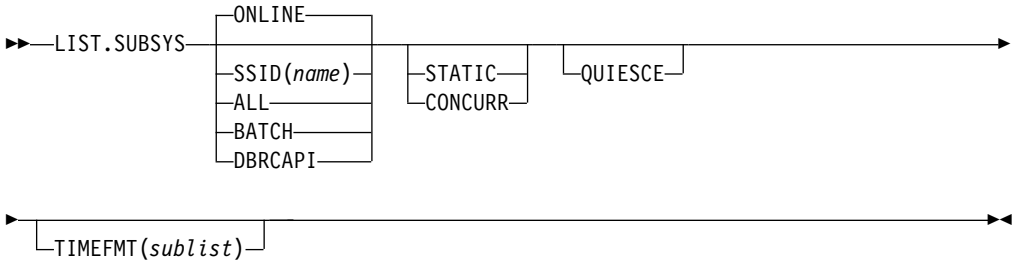
LIST.SUBSYS command

Use the LIST.SUBSYS command to receive a formatted list of the subsystems that are registered in the RECON data set.

Subsections:

- “Syntax”
- “Keywords”
- “Example of displaying all online subsystem records” on page 343

Syntax



Keywords

ONLINE | SSID(*name*) | ALL | BATCH | DBRCAPI

Mutually exclusive, optional keywords you use to specify which subsystem information is to be displayed.

ONLINE

Specifies that all online subsystem information is to be displayed.

SSID(*name*)

Specifies the name of the subsystem for which information is to be displayed.

ALL

Specifies that all subsystem information, both batch and online, is to be displayed.

BATCH
Specifies that all batch subsystem information is to be displayed.

DBRCAPI
Specifies that all DBRCAPI subsystem information is to be displayed.

QUIESCE
Optional keyword to indicate that, if the RECON access is parallel and the STATIC keyword is specified or defaulted to, the RECON data set is to be quiesced to process the command. QUIESCE is ignored if the CONCURR keyword is specified or defaulted to.

STATIC | CONCURR
Mutually exclusive, optional keywords that you use to control how the list output is produced when parallel RECON access is in effect. If specified when parallel access is not in effect, the keyword is ignored.

If neither STATIC nor CONCURR is specified, the default setting from the RECON header record is used. (The default is set by the INIT.RECON command, by a CHANGE.RECON LIST command, or by upgrade processing.)

STATIC
Specifies that the list output is to be consistent as of a point in time during list processing. The RECON records accessed by list processing cannot be updated by other DBRC instances until list processing has completed.

CONCURR
Specifies that list processing can occur concurrent with RECON update activity. In this case, the list output may not reflect the state of the RECON data at any point in time.

TIMEFMT(*sublist*)
Optional keyword you use to define the form in which time stamps appear in messages, displays, and listings from DBRC. The five values are positional. Each is optional and can be omitted by including only the comma.

The TIMEFMT default for LIST commands is obtained from what is specified in the RECON header record.

Example of displaying all online subsystem records

In this example, all online subsystem records are displayed.

```
//LISTSS   JOB

⋮
//SYSIN    DD   *
   LIST.SUBSYS
/*
```

Related reference:

Chapter 18, “Sample listings from the RECON data set,” on page 393

“DBRC time stamps” on page 52

Chapter 16. NOTIFY commands

Use the NOTIFY commands to add information to a RECON data set that is normally written there automatically.

NOTIFY.ALLOC command

Use the NOTIFY.ALLOC command to add information to the RECON data set about either a specific database allocation or a specific database deallocation of a DBDS or DEDB area.

This addition of information is required only when the RECON data set was not updated during a run of IMS that resulted in an allocation of the DBDS or DEDB area for updates. Under normal operating conditions, you will not need to use this command.

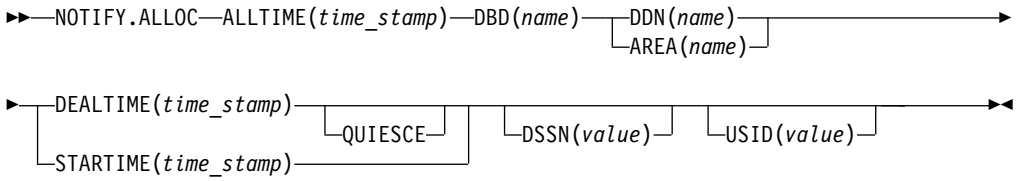
The NOTIFY.ALLOC command fails if the DBDS or DEDB area is nonrecoverable, user-recoverable, or in use.

Restriction for HALDB databases: This command is not allowed for ILDS or index DBDSs of HALDB partitions.

Subsections:

- "Syntax"
- "Keywords"
- "Example of adding allocation information to the RECON data set" on page 346

Syntax



Keywords

ALLTIME(*time_stamp*)
Required keyword you use to specify the time stamp of the allocation of the database that contains the DBDS or DEDB area that is specified in this command. The time stamp must be in standard form.

When used with the STARTIME keyword, ALLTIME causes a new allocation record to be written in the RECON data set. When used with a DEALTIME keyword, it identifies the allocation record in the RECON data set for which a deallocation time is being added.

DBD(*name*)
Required keyword you use to specify the database name of the DBDS or DEDB area for which you are adding allocation information to the RECON data set.

© Copyright IBM Corp. 1974, 2017

Restriction for HALDB databases: When using the NOTIFY.ALLOC command for HALDB databases, *name* must specify the name of a HALDB partition.

DDN(*name***) | AREA(***name***)**
Mutually exclusive, required keywords you use to specify the data set ddname of the DBDS or DEDB area for which you are adding allocation information to the RECON data set.

DEALTIME(*time_stamp***) | STARTIME(***time_stamp***)**
Mutually exclusive, required keywords. The time stamp must be in standard form.

DEALTIME
Specifies the time stamp of the deallocation of the database for the specified DBDS or DEDB area. This addition to the RECON data set is required only if the database is allocated for updates and explicitly deallocated before the end of an IMS run.

STARTIME
Specifies the starting time stamp of the log data set that was active at the time of the allocation specified in the ALLTIME keyword.

DSSN(*value***)**
Optional keyword you use to specify which data set sequence number is placed in the allocation record to be created. If you do not specify the DSSN keyword, the data set sequence number for the new allocation record is 0, indicating no data sharing. If you are using data sharing, you must specify the appropriate DSSN. You use this keyword for log-merge processing.

QUIESCE
Optional keyword you use to specify that the DEALLOC time stamp specified was caused by a database quiesce command. This keyword is valid only when DEALTIME is specified.

USID(*value***)**
Optional keyword you use to specify the update set identifier of the database or area when the update occurred.

The database or area cannot be authorized if the specified USID value is greater than the current USID for the database or area. Also, this value cannot be more than one greater than the current USID.

Example of adding allocation information to the RECON data set

In this example, information about an allocation of a specified DBDS is to be added to the RECON data set. The ALLTIME keyword specifies the time stamp of the allocation of the DBDS or DEDB area; the STARTIME keyword specifies the time stamp of the start of the log data set that was active at the time of the allocation.

```
//NFYALLOC JOB
⋮
//SYSIN    DD   *
     NOTIFY.ALLOC   DBD(DB1)  DDN(DD1)  -
                    STARTIME(07067020101023456)  -
                    ALLTIME(070670308200)
/*
```

Related reference:

"DBRC time stamps" on page 52

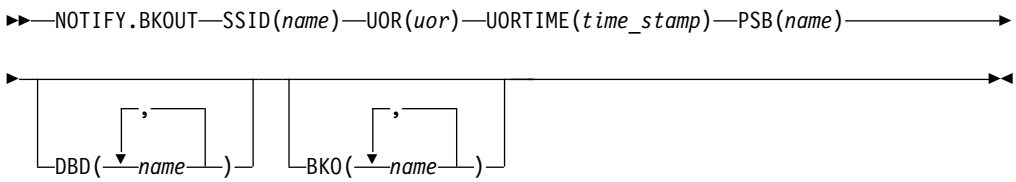
NOTIFY.BKOUT command

Use the NOTIFY.BKOUT command to create a backout record for a specified subsystem and to add a single unit of recovery (UOR) entry to the record that is created. Additional UOR entries can be added to the backout record by using the CHANGE.BKOUT command.

Subsections:

- "Syntax"
- "Keywords"
- "Example of adding a backout record to the RECON data set" on page 348

Syntax



Keywords

SSID(*name***)**
Required keyword you use to specify the subsystem for which the backout record is to be created. The name is an eight-character, alphanumeric string that represents any valid subsystem name.

UOR(*uor***)**
Required keyword you use in conjunction with the UORTIME keyword to identify a unit of recovery in the backout record. The recovery token (*uor*) is a 16-byte field that describes a specific UOR that is to be included with the backout record. *uor* must be 32 hexadecimal digits expressed as a character string; for example, `UOR(E2E8E2F340404040000000000600000003)`.

The recovery token is intended to be a unique identifier, but it can be duplicated across restarts. When you include UORTIME, you eliminate the problem of possible duplication.

UORTIME(*time_stamp***)**
Required keyword you use to specify the time of the UOR to be added to the backout record. The value is the time stamp of the beginning of the UOR (found in the X'5607' log record). The time stamp must be in standard form.

PSB(*name***)**
Required keyword you use to identify the PSB associated with the UOR.

DBD(*name***)**
Optional keyword you use to identify up to eight databases having changes associated with the unit of recovery that require backout.

Restriction for HALDB databases: When using the `NOTIFY.BKOUT` command for HALDB databases, *name* must specify the name of a HALDB partition.

BKO(*name***)**
Optional keyword you use to identify up to eight databases having changes associated with the unit of recovery that have already been backed out.

Use BKO to identify databases that have already been backed out from this UOR-UORTIME combination. You can specify either the BKO keyword, the DBD keyword, or both. A database name can appear in either BKO or DBD, but not both. (A database cannot both be backed out and require a backout at the same time.)

Example of adding a backout record to the RECON data set

In this example, a backout record for subsystem SYS3 is added to the RECON data set.

```
 //NFYBKOUT JOB
⋮
 //SYSIN    DD  *
      NOTIFY.BKOUT  SSID(SYS3)
                    UOR(E2E8E2F340404040000000600000003)
                    UORTIME(070931345027) PSB(APPL34)
                    DBD(DATA1,DATA2,DATA3C)
                    BKO(DATA4,DATA5,DATA3A)
 /*
```

See "Standard time stamp format" on page 53.

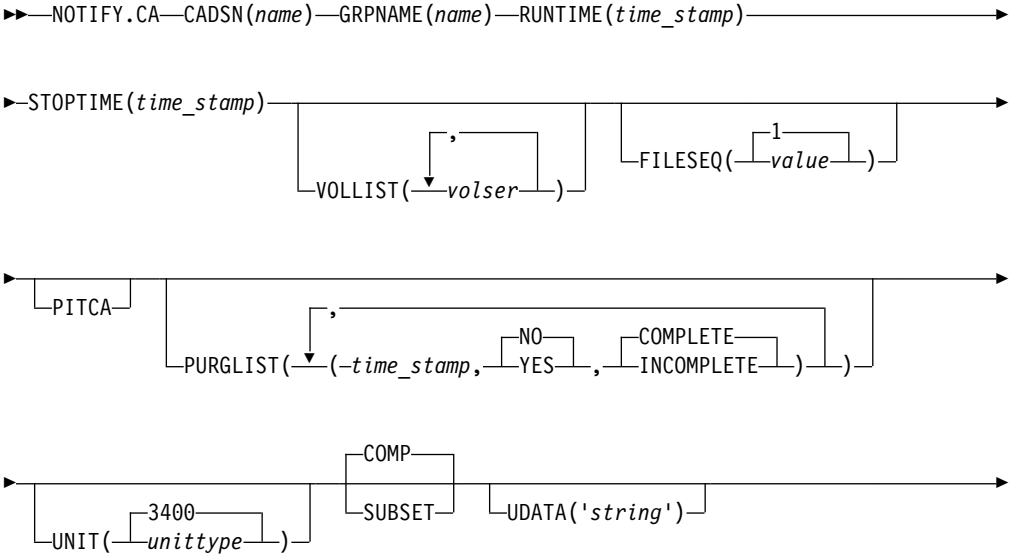
NOTIFY.CA command

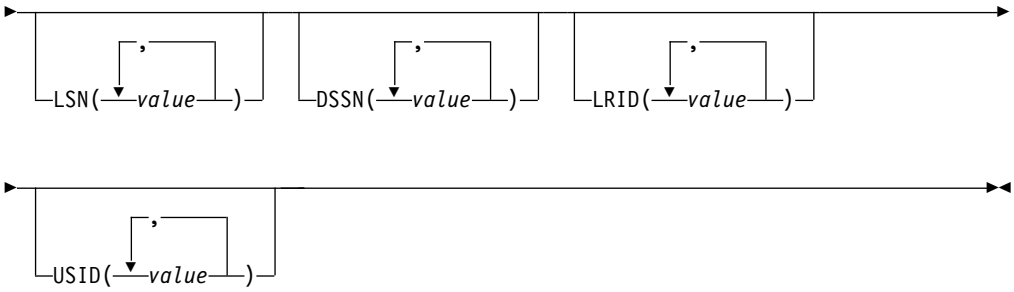
Use the NOTIFY.CA command to add information to the RECON data set about a run of the Database Change Accumulation utility for a specified CA group.

Subsections:

- "Syntax"
- "Keywords" on page 349
- "Example of adding CADSN information to the RECON data set" on page 352

Syntax





Keywords

CADSN(*name***)**
Required keyword that you use to specify the data set name of the change accumulation data set in the identified record. If the CA group is defined as reusable, the data set name must be unique. DBRC does not check for duplicate data set names.

GRPNAME(*name***)**
Required keyword that you use to specify the name of the CA group for which information is to be added.

RUNTIME(*time_stamp***)**
Required keyword that you use to identify the specific change accumulation run record to be added. The time stamp represents the time at which the Database Change Accumulation utility was run, and it must be in standard form.

STOPTIME(*time_stamp***)**
Required keyword that you use to specify the time stamp of the change accumulation run record for which information is to be added. The time stamp is the stop time of the last log volume that was processed by the specified run of the Change Accumulation utility, and it must be in standard form.

VOLLIST(*volser***)**
Optional keyword that you use to specify the volume serial numbers of the volumes on which the change accumulation data set being defined resides. You can specify a maximum of 255 volume serial numbers for *volser*. Each volume serial number can be a maximum of 6 alphanumeric characters, and they must follow z/OS JCL conventions for volume serial numbers. VOLLIST is required if the change accumulation data set is not to be treated as cataloged, as indicated in the RECON status record.

FILESEQ(1 | *value*)
Optional keyword that you use to specify the file sequence number of the change accumulation data set that is being defined. If you specify the FILESEQ keyword with a value greater than 1, you must also specify the VOLLIST keyword. *value* must be a decimal number from 1 to 9999.

PITCA
Optional keyword that you use to add a CA record that was generated outside of IMS. The record added will have the PITCA and ERR flags ON. No other DBRC functions such as GENJCL.CA or GENJCL.RECOV will use this CA.

PURGLIST(*time_stamp***,NO | YES,COMPLETE | INCOMPLETE)**
Optional keyword that you use to specify the purge time, which is the point in time in the input log records where change accumulation started and to specify whether the logs form a complete subset.

The time stamp must be in standard form. If you do not specify a time stamp, the time is set to 0.

If you are using the accumulated changes as input to recovery, you must choose a purge time that satisfies the DBRC input requirements for recovery. Recovery first chooses an image copy and then uses a change accumulation whose purge time for that DBDS matches the run time of the image copy.

NO | YES
Mutually exclusive subparameters you use to specify whether any changes for the corresponding DBDS have been accumulated.

NO Specifies that no changes have been accumulated for the corresponding DBDS.

YES
Specifies that some changes have been accumulated for the corresponding DBDS.

COMPLETE | INCOMPLETE
Mutually exclusive subparameters you use to specify whether the logs form a complete subset. To determine whether a log subset is complete, use the LIST.CAGRP command.

COMPLETE
Specifies that the logs form a complete subset. When you specify COMPLETE, the time stamp of the STOPTIME keyword is the stop time of the last log input to the Change Accumulation utility.

INCOMPLETE
Specifies that the logs form an incomplete subset. When you specify INCOMPLETE, the time stamp of the STOPTIME keyword is the start time of the earliest unselected (open) log volume. This volume should be the first one that is selected at a later run.

If you specify the PURGLIST keyword, the order of the time stamp and the change indicator in the purge list corresponds to the order of the DBDS names specified in the GRPMEM keyword of the INIT.CAGRP command. For example, the third purge time and change indicator is the purge time for the third DBDS that is specified in the GRPMEM keyword of the INIT.CAGRP command.

If you specify fewer subparameters with the PURGLIST keyword than you specified with the GRPMEM keyword of the INIT.CAGRP command, DBRC uses the defaults of NO and COMPLETE for each DBDS that you omit. Similarly, if you do not specify the PURGLIST keyword, DBRC uses the defaults of NO and COMPLETE for each DBDS specified with the GRPMEM keyword of the INIT.CAGRP command. To use a default of NO for certain DBDSs, use commas to indicate which DBDSs are subject to the default.

UNIT(3400 | *unittype*)
Optional keyword that specifies the unit type of the volumes on which the change accumulation data set resides. *unittype* can be up to 8 alphanumeric characters. If you specify the UNIT keyword, you must also specify the VOLLIST keyword.

COMP | SUBSET
Mutually exclusive, optional keywords you use to indicate that the change accumulation record's stop time is a log volume start time.

COMP
Indicates that when the CA was created, a complete set of logs was processed and that the CA's stop time is the stop time of the last log volume processed.

SUBSET
Indicates that when the CA was created, a subset of logs was processed and the CA's stop time is the start time of the first unprocessed log volume. Specifying INCOMPLETE in the PURGLIST keyword does not automatically cause SUBSET to be set.

You do not need to use this keyword under normal conditions. Checking is not done to verify that the use of this keyword is consistent with the value of the CA stop time. This keyword value is used by the GENJCL.CA and GENJCL.RECOV processes. Incorrect use of this keyword can result in invalid generated JCL.

UDATA('*string*')
Optional keyword that you use to specify up to 80 bytes of information about the identified change accumulation data set. You can use the variable field of this keyword to describe how the change accumulation data set was created. The *string* value must be enclosed in single quotation marks if blanks or any other special characters are included.

LSN(*value*)
Optional keyword that you use to specify the lock sequence number of the last change that is accumulated for the DBDS.

The lock sequence number can be one of the following:

- A hexadecimal number of 1 to 12 characters, which is enclosed in single quotation marks and preceded by the letter, X. For example: `LSN(X'10B9C')`
- A decimal number from 0 to (2**48) - 1, specified without delimiters. For example: `LSN(68508)`

DSSN(*value*)
Optional keyword that you use to specify the data set sequence number of the last change that is accumulated for the DBDS.

The data set sequence number can be a decimal number from 0 to (2**32) - 1, specified without delimiters. For example: `DSSN(68508)`

LRID(*value*)
Optional keyword that you use to specify the last log record ID of the last change that is accumulated for the DBDS.

The last log record ID can be one of the following:

- A hexadecimal number of 1 to 16 characters, which is enclosed in single quotation marks and preceded by the letter, X. For example: `LRID(X'10B9C')`
- A decimal number from 0 to (2**64) - 1, specified without delimiters. For example: `LRID(68508)`

USID(*value*)
Optional keyword that you use to specify the last update set ID of the last change that is accumulated for the DBDS.

The last update set ID can be a decimal number from 0 to (2**32) - 1, specified without delimiters. For example: `USID(68508)`

Example of adding CADSN information to the RECON data set

In this example, information about a change accumulation data set is to be added to the RECON data set.

```
//NFYCA    JOB

.
.
.
//SYSIN    DD   *
     NOTIFY.CA   GRPNAME(CAGRP2) -
                 STOPTIME(070240202020) -
                 RUNTIME(070250305029) CADSN(CADSN06) -
                 VOLLIST(VOL005) -
                 PURGLIST((070240302005,YES),,(070250420256,))
/*
```

Related reference:

"DBRC time stamps" on page 52

NOTIFY.IC command

Use the NOTIFY.IC command to add information to the RECON data set about an image copy.

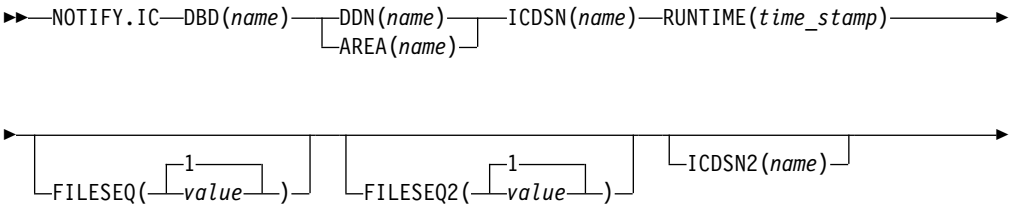
Restrictions for HALDB databases:

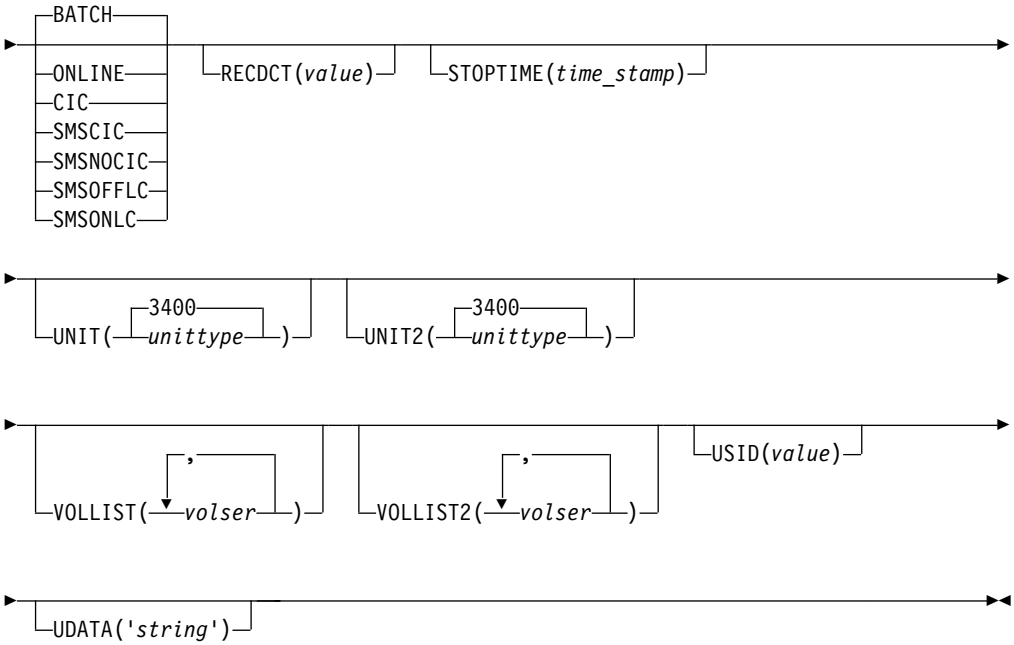
- The NOTIFY.IC command is not allowed for ILDS or Index DBDSs of HALDB partitions.
- The NOTIFY.IC command will fail if it is attempting to record an image copy of a HALDB that has an online reorganization in progress. This is because an image copy of a HALDB cannot be taken while the HALDB is being reorganized online and the reorganization has not completed (either it is in progress or was terminated).

Subsections:

- "Syntax"
- "Keywords" on page 353
- "Example of notifying DBRC of concurrent image copy completion" on page 355

Syntax





Keywords

DBD(*name*)
Required keyword you use to specify the database name of the DBDS or area for which an image copy run record is to be added.

Restriction for HALDB databases: When using the NOTIFY.IC command for HALDB databases, *name* must specify the name of a HALDB partition.

DDN(*name*) | AREA(*name*)
Mutually exclusive, required keywords you use to specify the data set ddname of the DBDS (use DDN) or DEDB area (use AREA) for which an image copy run record is to be added.

ICDSN(*name*)
Required keyword you use to specify the data set name of the image copy data set that contains the image copy whose run record is being added. *name* can be a maximum of 44 characters.

RUNTIME(*time_stamp*)
Required keyword you use to specify the time the Image Copy utility was run. The time stamp must be in standard form.

FILESEQ(1 | *value*)
Optional keyword you use to specify the file sequence number of the identified image copy data set. *value* must be a decimal number from 1 to 9999.

FILESEQ2(1 | *value*)
Optional keyword you use to specify the file sequence number of the identified duplicate image copy data set. *value* can be a decimal number from 1 to 9999.

You can specify this keyword only if you specify the VOLLIST2 keyword. If the VOLLIST2 keyword is specified, then FILESEQ2(1) is the default for this keyword.

ICDSN2(*name*)
Optional keyword you use to specify the data set name of the duplicate image

copy data set that is to contain the image copy whose run record is being added. *name* can be a maximum of 44 characters.

BATCH| ONLINE | CIC | SMSCIC | SMSNOCIC | SMSONLC | SMSOFFLC
Mutually exclusive, optional keywords you use to specify the type of image copy that the data set contains.

Restriction: CIC, SMSCIC, and ONLINE cannot be used for user-recoverable databases.

BATCH
Indicates that the Database Image Copy (DFSUDMP0) utility was used to create the image copy while the database was unavailable for update processing (the CIC keyword was not specified). BATCH can also be specified to record the output of the HISAM Reorganization Unload utility image copy.

ONLINE
Specifies that the image copy data set was obtained by executing the Online Database Image Copy utility. You must use the STOPTIME keyword when you specify ONLINE.

CIC
Indicates that a concurrent image copy was taken. A concurrent image copy is a "fuzzy" copy, so the data set uses logs in order to complete the image. STOPTIME must be used if CIC is specified. CIC cannot be used to copy VSAM KSDS databases.

SMSCIC
Indicates that the Database Image Copy 2 was used to create the image copy while the database was available for update processing ('S' was specified on the utility control statement). The image copy is in DFSMS dump format. The image copy is a "fuzzy" copy so logs must be applied to recover the data set to a usable state. The STOPTIME keyword must be specified when you specify SMSCIC.

SMSNOCIC
Indicates that the Database Image Copy 2 utility was used to create the image copy while the database was unavailable for update processing ('X' was specified on the utility control statement). The image copy is in DFSMS dump format.

SMSOFFLC
Indicates that the Database Image Copy 2 utility invoked DFSMS Fast Replication to take an image copy while the database was unavailable for update processing. The image copy was created by a DFSMS COPY command.

SMSONLC
Indicates that the Database Image Copy 2 utility invoked DFSMS Fast Replication to take an image copy concurrently with update processing. The image copy was created by a DFSMS COPY command format. The image copy is a "fuzzy" copy, so logs must be applied to recover the data set to a usable state. The STOPTIME keyword must be specified when you specify SMSONLC.

RECDCT(*value*)
Optional keyword you use to specify the count of the records in the image copy data set. *value* must be a decimal number from 1 to 2147483647.

STOPTIME(*time_stamp*)
Optional keyword you use to specify the stop time of the online or concurrent image copy. You must specify this keyword when online, CIC, SMSCIC, or SMSONLC is specified. The time stamp must be in standard form.

UNIT(3400 | *unittype*)
Optional keyword you use to specify the unit type of the image copy data set. The unit type can be up to 8 alphanumeric characters.

UNIT2(3400 | *unittype*)
Optional keyword you use to specify the unit type of the duplicate image copy data set. The unit type can be up to eight alphanumeric characters long.

VOLLIST(*volser*)
Optional keyword you use to specify the volume serial numbers of the volumes on which the image copy data set resides. You can substitute from 1 to 255 volume serial numbers in the variable field. Each volume serial number can be up to 6 alphanumeric characters long, and must follow z/OS JCL conventions for volume serial numbers.

Attention: VOLLIST is required if the image copy data set is not treated as cataloged, as indicated in the RECON status record, and if this data set is not an SMS image copy.

VOLLIST2(*volser*)
Optional keyword you use to specify the volume serial numbers of the volumes on which the duplicate image copy data set resides. You can substitute from 1 to 255 volume serial numbers in the variable field. Each variable can be up to 6 alphanumeric characters long, and must follow z/OS JCL conventions for volume serial numbers. VOLLIST2 is allowed only if the ICDSN2 keyword is also specified.

Attention: VOLLIST2 is required if the image copy data set is not treated as cataloged, as indicated in the RECON status record, and if this data set is not an SMS image copy.

USID(*value*)
Optional keyword you use to specify the update set identifier of the database or area when the reorganization occurred.

UDATA('*string*')
Optional keyword you use to specify up to 80 bytes of information about the identified, standard image copy data set. You can use the variable field of this keyword to describe how the standard image copy data set was created. The *string* value must be enclosed in single quotation marks if blanks or any other special characters are included.

Example of notifying DBRC of concurrent image copy completion

In this example, DBRC is notified of the successful completion of a concurrent image copy for the area specified. RUNTIME refers to the time the image copy started. STOPTIME refers to the time the image copy ended.

```
 //NFYIC    JOB

⋮
 //SYSIN    DD   *
      NOTIFY.IC   DBD(DBD001) AREA(AREA1)                              -
                  RUNTIME(0720002020)                                  -
```

```
                STOPTIME(0720004040)                              -
                ICDSN(IC0005) CIC                                 -
                UDATA('FIRST CIC OF THE WEEK')
 /*
```

Related reference:

"DBRC time stamps" on page 52

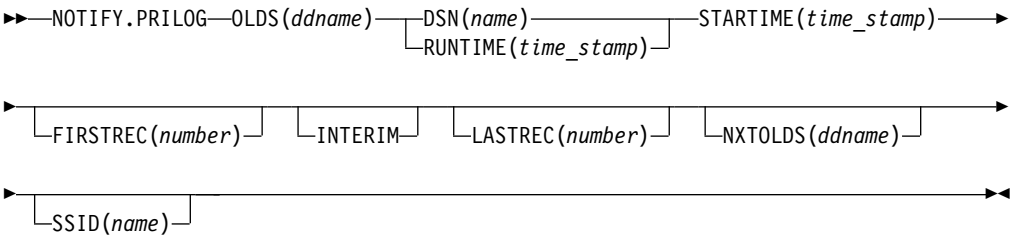
NOTIFY.PRILOG command (for OLDS)

Use a NOTIFY.PRILOG command to add information about a primary OLDS to the RECON data set and to manually create interim PRILOG in the RECON data set. You would do this in a case where the log processing exit routines of the IMS system failed to do so.

Subsections:

- "Syntax"
- "Keywords"
- "Examples" on page 358

Syntax



Keywords

OLDS(*ddname***)**
Required keyword that specifies that a record will be created in the RECON data set for an OLDS.

ddname
Specifies the name of the DD statement that the IMS online control region used when it used the OLDS.

Note: Failure to specify the OLDS keyword results in an attempt to create or update an RLDS record.

DSN(*name***) | RUNTIME(***time_stamp***)**
Mutually exclusive, required keywords.

DSN
Specifies the data set name of the primary OLDS for which a log record is being created in the RECON data set.

RUNTIME
Specifies the time stamp of a close operation for the specified primary OLDS. The time stamp must be in standard form.

These two keywords are used in conjunction with the STARTIME, FIRSTREC, and NXTOLDS keywords to identify what type of primary OLDS entry is to be

added to the RECON data set. The following table indicates which keyword combinations are required for each type of primary OLDS entry.

Definitions:

OLDS Open
Starting to use an OLDS and inserting a starting time.

OLDS Switch
Stopping the use of the current OLDS (inserting a stop time) and starting to use the next OLDS (inserting a starting time into the new OLDS).

OLDS Close
Stopping the use of an OLDS (inserting a stop time)

Table 15. Keywords of NOTIFY.PRILOG command (for OLDS) for open, switch, and close

Type of log entry	Required keywords
OLDS Open	STARTIME, DSN, FIRSTREC
OLDS Switch	STARTIME, DSN, FIRSTREC, NXTOLDS
OLDS Close	LASTREC, STARTIME, RUNTIME

For each primary OLDS, you must issue a separate NOTIFY.PRILOG command for open, switch, and close.

STARTIME(*time_stamp***)**
Required keyword you use to specify the starting time of a primary OLDS. The time stamp must be in standard form. For a description of the use of this keyword with other keywords in the `NOTIFY.PRILOG` command, see Table 15.

FIRSTREC(*number***)**
Optional keyword you use to specify the log record sequence number of the first log record of the OLDS. For the first OLDS of the PRILOG, it corresponds to the first log record that was written during initialization of the IMS subsystem.

FIRSTREC is required for OLDS OPEN and SWITCH commands. It specifies the first log record sequence number on the OLDS that is being opened. It is invalid for a CLOSE command.

The log record sequence number can be one of the following:

- A hexadecimal number

  This number is 1 to 16 characters, enclosed in single quotation marks and preceded by the letter, X. For example: `FIRSTREC(X'10B9C')`.
- A decimal number

  This number is a decimal number from 0 to (2**64)-1, without delimiters. For example: `FIRSTREC(68508)`.

In either case, leading zeros can be omitted.

INTERIM
Optional keyword you use to specify that an interim log data set record is to be created.

Before you specify NOTIFY.PRILOG INTERIM, a corresponding primary log record must exist.

LASTREC(*number***)**
Optional keyword you use to specify the log record sequence number of the last log record of the OLDS.

LASTREC is required for the OLDS CLOSE command. It is optional for the SWITCH command; if it is omitted, the FIRSTREC value minus 1, is recorded for the OLDS that is being closed. It is invalid for an OPEN command.

The log record sequence number can be one of the following:

- A hexadecimal number

  This number is 1 to 16 characters, enclosed in single quotation marks and preceded by the letter, X. For example: `LASTREC(X'10B9C')`.
- A decimal number

  This number is a decimal number from 0 to (2**64)-1, without delimiters. For example: `LASTREC(68508)`.

In either case, leading zeros can be omitted.

NXTOLDS(*ddname***)**
Optional keyword you use when the RECON data set is to be updated to reflect an OLDS switch. The current OLDS is closed and an IMS online control region opens a new OLDS. *ddname* is the DD statement of the OLDS being opened. You specify the OLDS being closed with the OLDS(*ddname*) keyword. Use the DSN(*name*) keyword to specify the data set name of the OLDS being opened. Use the STARTIME(*time_stamp*) keyword to specify the close time of the OLDS being closed and the open time of the OLDS being opened.

SSID(*name***)**
Optional keyword you use to specify the name of the IMS subsystem that created the log data set.

The SSID is an eight-character, alphanumeric string that represents a valid IMS subsystem identification name. If you do not specify SSID, DBRC uses the default subsystem identifier in the RECON header record. You use the INIT.RECON or CHANGE.RECON command to set the default subsystem identifier in the RECON header record. If you have not specified a default in the RECON header record, you must specify SSID.

Examples

Here are some examples of using the `NOTIFY.PRILOG (for OLDS)` command.

Example of creating a PRIOLDS for an online subsystem OLDS

In this example, you create a PRIOLDS for an OLDS that belongs to IMS online subsystem IMSA.

```
//NFYPRILG JOB
.
.
.
//SYSIN    DD   *
     NOTIFY.PRILOG   STARTIME(071230554321)  -
                     DSN(IMS.OLDSP13) OLDS(DFSOLP13) -
                     FIRSTREC(001) -
                     SSID(IMSA)
/*
```

Example of adding information about two primary OLDSs to the RECON data set

In this example, you create a PRIOLDS for two OLDSs that belong to the IMS online subsystem IMSA. Both OLDSs are closed. The first STARTIME keyword specifies the time stamp of the opening of the primary OLDS. The DSN keyword indicates that information that is added relates to the opening of the OLDS. NXTOLDS indicates an OLDS switch. The second STARTIME keyword and second

DSN indicate the start time and DSN of the next OLDS. The third STARTIME keyword indicates the start time of the OLDS to be closed. The RUNTIME keyword is the time stamp of the closing volume.

```
NOTIFY.PRILOG SSID(IMSA) STARTIME(072171212120) OLDS(DFSOLP01) -
       DSN(IMS.OLDP01) LASTREC(4999)
NOTIFY.PRILOG SSID(IMSA) STARTIME(072181212120) OLDS(DFSOLP01) -
       DSN(IMS.OLDP02) NXTOLDS(DFSOLP02)
NOTIFY.PRILOG SSID(IMSA) STARTIME(072181212120) OLDS(DFSOLP02) -
       RUNTIME(072191010101)
/*
```

Related reference:

"DBRC time stamps" on page 52

NOTIFY.PRILOG command (for RLDS)

Use a NOTIFY.PRILOG command to add information about a primary RLDS (or a primary SLDS that a batch subsystem created) to the RECON data set and to manually create interim-primary log data set records in the RECON data set.

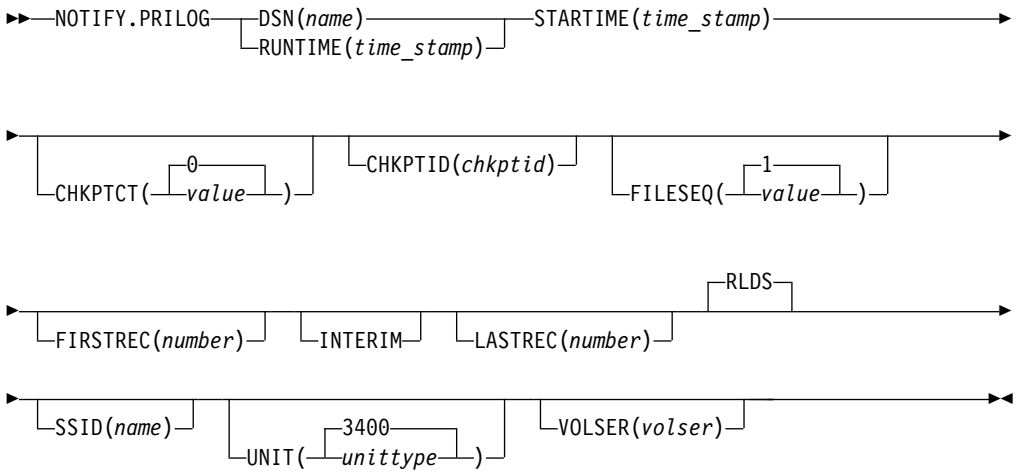
This is information that could not be added to the RECON data set from the IMS system log processing exit routines. If you are processing DBDSs with IMS, you should not need to use this command under normal operating conditions. You must specify a NOTIFY.ALLOC command for each DBDS for which change records might exist on the primary RLDS being added.

This command adds or completes a data set entry in a PRILOG record. If you are modifying an existing completed data set entry, you should use the CHANGE.PRILOG (RLDS) command.

Subsections:

- "Syntax"
- "Keywords" on page 360
- "Examples" on page 362

Syntax



Keywords

DSN(*name***) | RUNTIME(***time_stamp***)**
Mutually exclusive, required keywords.

DSN
Specifies the data set name of the primary RLDS for which a log record is being created in the RECON data set.

RUNTIME
Specifies the time stamp of a close or end-of-volume (EOV) operation for the specified primary RLDS. The time stamp must be in standard form.

These two keywords are used in conjunction with the STARTIME and VOLSER keywords to identify what type of primary-recovery-log-data set entry is to be added to the RECON data set.

The following table indicates which keyword combinations are required for each type of primary-recovery-log-data-set entry.

Definitions:

RLDS Open
Starting to use an RLDS and inserting a starting time.

RLDS EOV
RLDS end of volume (EOV) notification from the operating system. This is analogous to an OLDS Switch.

RLDS Close
Stopping the use of an RLDS (inserting a stop time)

Table 16. Keywords of NOTIFY.PRILOG command (for RLDS) for open, EOV, and close

Type of log entry	Required keywords
RLDS Open	STARTIME, DSN, VOLSER, FIRSTREC
RLDS EOV	STARTIME, VOLSER, RUNTIME
RLDS Close	STARTIME, RUNTIME, LASTREC

For each primary RLDS, you must issue a separate NOTIFY.PRILOG command for open, zero or more EOVs, and close.

STARTIME(*time_stamp***)**
Required keyword you use to specify the starting time of a primary RLDS.

The time stamp must be in standard form.

See Table 16 for a description of the use of the STARTIME keyword with other keywords in the NOTIFY.PRILOG command.

CHKPTCT(0 | *value*)
Optional keyword you use to specify the number of checkpoints completed on the RLDS volumes.

The valid values for CHKPTCT are:

0 No checkpoints in the RLDS volume

1 A single checkpoint in the RLDS volume

2 More than one checkpoint in the RLDS volume

IMS uses the value of CHKPTCT to determine which logs are necessary to recover a Fast Path area with concurrent image copy.

CHKPTID(*chkptid*)
Optional keyword you use to specify the oldest checkpoint ID for an active partition specification table (PST) on an RLDS volume. The checkpoint ID must be in the standard form for a time stamp.

FILESEQ(1 | *value*)
Optional keyword you use to specify the file sequence number of the primary RLDS that is identified. You can specify this keyword only if you have specified the VOLSER keyword.

FIRSTREC(*number*)
Optional keyword you use to specify the log record sequence number of the first log record of the RLDS. For the first RLDS of the PRILOG, it corresponds to the first log record that was written during initialization of the IMS subsystem.

FIRSTREC is required if DSN is specified and is invalid if RUNTIME is specified.

The log record sequence number can be one of the following:

- A hexadecimal number

  This number is 1 to 16 characters, enclosed in single quotation marks and preceded by the letter, X. For example: `FIRSTREC(X'10B9C')`.
- A decimal number

  This number is a decimal number from 0 to (2**64)-1, without delimiters. For example: `FIRSTREC(68508)`.

In either case, leading zeros can be omitted.

INTERIM
Optional keyword you use to specify that an interim log data set record is to be created. Before you issue the NOTIFY.PRILOG INTERIM command, you must create a corresponding primary recovery log record.

LASTREC(*number*)
Optional keyword you use to specify the log record sequence number of the last log record of the RLDS.

LASTREC is required if RUNTIME is specified and VOLSER is not specified (that is, on a `CLOSE` call). LASTREC is invalid if DSN is specified.

The log record sequence number can be one of the following:

- A hexadecimal number

  This number is 1 to 16 characters, enclosed in single quotation marks and preceded by the letter, X. For example: `LASTREC(X'10B9C')`.
- A decimal number

  This number is a decimal number from 0 to (2**64)-1, without delimiters. For example: `LASTREC(68508)`.

In either case, leading zeros can be omitted.

RLDS
Optional keyword you use to specify that an RLDS record is to be created or updated.

SSID(*name*)
Optional keyword you use to specify the name of the IMS subsystem that created the log data set.

The SSID is an eight-character alphanumeric string that represents a valid IMS subsystem identification name. If you do not specify SSID, DBRC uses the default subsystem identifier in the RECON header record. You use the INIT.RECON or CHANGE.RECON command to set the default subsystem identifier in the RECON header record. If you have not specified a default in the RECON header record, you must specify SSID.

UNIT(3400 | *unittype*)
Optional keyword you use to specify the unit type of the RLDSs. You only specify the UNIT keyword if you specify the DSN keyword. The unit type can be up to 8 alphanumeric characters long.

VOLSER(*volser*)
Optional keyword you use to specify the volume serial number of the log volume being recorded for the identified primary RLDS. For an EOV notification, this volume serial number is that of the volume being started.

You must use the VOLSER keyword during RLDS open and EOV processing.

Examples

Here are some examples of using the NOTIFY.PRILOG (for RLDS) command.

Example of adding primary RLDS information to the RECON data set

In this example, information about a primary RLDS is to be added to the RECON data set. The VOLSER and DSN keywords indicate that the information to be added relates to the opening of the primary RLDS. The STARTIME keyword specifies the time stamp of the opening of the primary RLDS. The first RUNTIME keyword specifies the time stamp of the EOV of the first volume of the primary RLDS. The second RUNTIME keyword specifies the time stamp of the closing volume of the primary RLDS.

```
//NFYPRILG JOB
.
.
.
//SYSIN    DD   *
     NOTIFY.PRILOG   RLDS STARTIME(07067020101023456)  -
                     VOLSER(VOL001)  DSN(PRILOG1) FIRSTREC(001)
     NOTIFY.PRILOG   RLDS STARTIME(07067020101023456)  -
                     VOLSER(VOL002)  RUNTIME(070670202020)
     NOTIFY.PRILOG   RLDS STARTIME(07067020101023456)  -
                     LASTREC(9999) RUNTIME(070670303030)
/*
```

Example of adding interim-primary RLDS information to the RECON data set

In this example, information about the interim-primary RLDS is to be added to the RECON data set. The STARTIME keyword specifies the time stamp of the opening of the interim primary RLDS.

```
//NFYPRILG JOB
.
.
.
//SYSIN    DD   *
     NOTIFY.PRILOG  RLDS STARTIME(07254123456123456) -
                    DSN(DSNIRLDS) -
                    VOLSER(VOL008) -
                    FIRSTREC(077) -
                    INTERIM
/*
```

Related reference:

"DBRC time stamps" on page 52

NOTIFY.PRILOG command (for SLDS)

Use a NOTIFY.PRILOG command to add information about a primary SLDS to the RECON data set and to manually create interim-primary log data set records in the RECON data set. This is information that could not be added to the RECON data set from the IMS system log processing exit routines.

If you are processing DBDSs with IMS, you should not need to use this command under normal operating conditions. You must specify a NOTIFY.ALLOC command for each DBDS for which change records might exist on the primary SLDS being added.

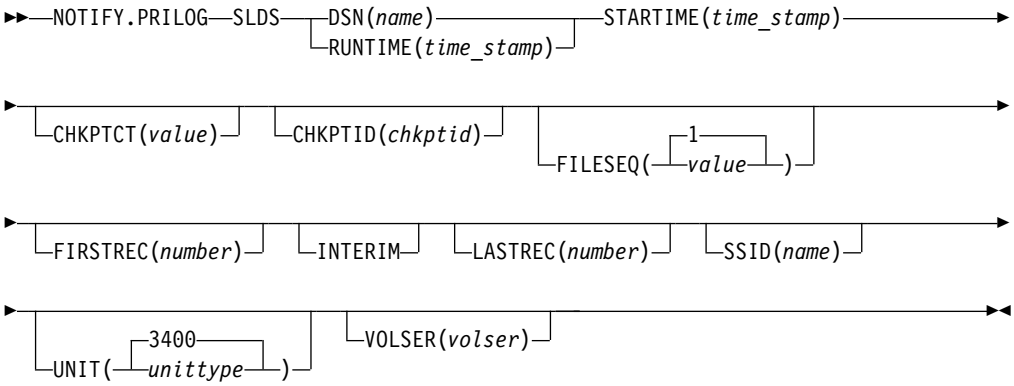
This command adds or completes a data set entry in the PRISLD record. If you are modifying an existing completed data set entry, you should use the CHANGE.PRILOG (for SLDS) command.

When you issue a NOTIFY.PRILOG for an SLDS, a PRILOG record must exist for the corresponding RLDS. Use NOTIFY.PRILOG (for RLDS) to add information about an SLDS that a batch subsystem creates, because DBRC considers such a data set to be an RLDS.

Subsections:

- "Syntax"
- "Keywords"
- "Example of adding primary SLDS information to the RECON data set" on page 366

Syntax



Keywords

SLDS
Required keyword that specifies that a record will be created or updated for an SLDS.

Note: Failure to specify the SLDS keyword results in an attempt to create or update an RLDS record.

DSN(*name***) | RUNTIME(***time_stamp***)**
Mutually exclusive, required keywords.

DSN
Specifies the data set name of the primary SLDS for which a log record is being created in the RECON data set.

RUNTIME
Specifies the time stamp of a close or EOV operation for the specified primary SLDS. The time stamp must be in standard form.

These two keywords are used in conjunction with the STARTIME, FIRSTREC, LASTREC, and VOLSER keywords to identify what type of primary system log data set entry is to be added to the RECON data set. The following table indicates which keyword combinations are required for each type of primary system log data set entry.

Definitions:

SLDS Open
Starting to use an SLDS and inserting a starting time.

SLDS EOV
SLDS end of volume (EOV) notification from the operating system. This is analogous to an OLDS Switch.

SLDS Close
Stopping the use of an RLDS (inserting a stop time)

Table 17. Keywords of NOTIFY.PRILOG (SLDS) command for open, EOV, and close

Type of log entry	Required keywords
SLDS Open	STARTIME, DSN, VOLSER, FIRSTREC
SLDS EOV	STARTIME, VOLSER, RUNTIME
SLDS Close	STARTIME, RUNTIME, LASTREC

For each primary SLDS, you must issue a separate NOTIFY.PRILOG command for open, zero or more EOVs, and close.

STARTIME(*time_stamp***)**
Required keyword that you use to specify the starting time of a primary SLDS. Use the log start time from the subsystem record or the PRILOG record. The time stamp must be in standard form. For a description of the use of this keyword with other keywords in the NOTIFY.PRILOG command, see Table 17.

CHKPTCT(*value***)**
Optional keyword that you use to change the number of checkpoints completed on the SLDS volumes. You specify a value for each SLDS volume that is designated.

The valid values for CHKPTCT are:

0 No checkpoints in the SLDS volume

1 A single checkpoint in the SLDS volume

2 More than one checkpoint in the SLDS volume

IMS uses the value of CHKPTCT to determine which logs are necessary to recover a Fast Path area with concurrent image copy.

CHKPTID(*chkptid***)**
Optional keyword that you use to specify the oldest checkpoint ID for an active PST on an SLDS volume. The checkpoint ID must be in the standard form for a time stamp.

FILESEQ(1 | *value*)
Optional keyword that you use to specify the file sequence number of the primary SLDS that is identified. You specify this keyword only if you have also specified the VOLSER keyword.

FIRSTREC(*number*)
Optional keyword that you use to specify the log record sequence number of the first log record of the SLDS. For the first SLDS of the PRISLD, FIRSTREC corresponds to the first log record that was written during initialization of the IMS subsystem.

FIRSTREC is required if DSN is specified and is invalid if RUNTIME is specified.

The log record sequence number can be one of the following:

- A hexadecimal number

  This number is 1 to 16 characters, enclosed in single quotation marks and preceded by the letter, X. For example: `FIRSTREC(X'10B9C')`.
- A decimal number

  This number is a decimal number from 0 to (2**64)-1, without delimiters. For example: `FIRSTREC(68508)`.

In either case, leading zeros can be omitted.

INTERIM
Optional keyword that you use to specify that an interim log data set record is to be created.

LASTREC(*number*)
Optional keyword that you use to specify the log record sequence number of the last log record of the SLDS.

LASTREC is required if RUNTIME is specified and VOLSER is not specified (that is, on a `CLOSE` call). LASTREC is invalid if DSN is specified.

The log record sequence number can be one of the following:

- A hexadecimal number

  This number is 1 to 16 characters, enclosed in single quotation marks and preceded by the letter, X. For example: `LASTREC(X'10B9C')`.
- A decimal number

  This number is a decimal number from 0 to (2**64)-1, without delimiters. For example: `LASTREC(68508)`.

In either case, leading zeros can be omitted.

SSID(*name*)
Optional keyword that you use to specify the name of the IMS subsystem that created the log data set.

The SSID is an eight-character, alphanumeric string that represents a valid IMS subsystem identification name. If you do not specify SSID, DBRC uses the default subsystem identifier in the RECON header record. You use the INIT.RECON or CHANGE.RECON command to set the default subsystem identifier in the RECON header record. If you have not specified a default in the RECON header record, you must specify SSID.

UNIT(3400 | *unittype*)
Optional keyword that you use to specify the unit type of the SLDSss. You only specify the UNIT keyword if you specify the DSN keyword. The unit type can be up to 8 alphanumeric characters.

VOLSER(*volser*)
Optional keyword that you use to specify the volume serial number of the log volume being recorded for the identified primary SLDS. For an EOV notification, this volume serial number is that of the volume being started.

Note: You must use the VOLSER keyword during SLDS open and EOV.

Example of adding primary SLDS information to the RECON data set

In this example, information about a primary SLDS is to be added to the RECON data set. The VOLSER and DSN keywords indicate that the information to be added relates to the opening of the primary SLDS. The STARTIME keyword specifies the time stamp of the opening of the primary SLDS. The first RUNTIME keyword specifies the time stamp of the EOV of the first volume of the primary SLDS. The second RUNTIME keyword specifies the time stamp of the closing volume of the primary SLDS.

```
//NFYPRILG JOB
⋮
//SYSIN    DD   *
     NOTIFY.PRILOG  SLDS STARTIME(07067020101023456)  -
                    VOLSER(VOL004)  DSN(PRILOG4) FIRSTREC(7000)
     NOTIFY.PRILOG  SLDS STARTIME(07067020101023456)  -
                    VOLSER(VOL005)  RUNTIME(070670202020)
     NOTIFY.PRILOG  SLDS STARTIME(07067020101023456)  -
                    RUNTIME(070670303030) LASTREC(8889)
/*
```

Related reference:

"DBRC time stamps" on page 52

NOTIFY.RECOV command

Use a NOTIFY.RECOV command to add information about recovery of a specified DBDS or DEDB area to the RECON data set.

You must use this command whenever you perform the recovery of a DBDS or DEDB area in any way other than using the Database Recovery utility (for example, by restoring the DASD volume on which the DBDS or area resides). In addition, you can notify DBRC when you recover a DBDS or DEDB area using the Database Recovery utility.

You can also use this command if you need to recover a DEDB area using the user image copy data set created by the DEDB Alter utility. Before you run the Database Recovery utility, issue the NOTIFY.RECOV command to notify the DBRC that you restored the DEDB area.

When specifying the RCVTIME keyword to inform DBRC of a time-stamp recovery, the RECON data set must contain a record of the image copy data set that you used to restore the DBDS or DEDB area. The image copy record can be either a standard or a clean nonstandard image copy. If it is a clean nonstandard image copy, then its time stamp cannot fall within the range of an existing time stamp recovery (the time between the RECOV TO and RUN times). The time stamp of the image copy record must be equal to that specified in the RCVTIME keyword of the NOTIFY.RECOV command. Do not use the NOTIFY.RECOV command with fuzzy nonstandard image copies.

For a receive, specify both the RCVUSID and RUNUSID parameters. Specify the same *usid* value for both parameters: the update set identifier (*usid*) of the image copy that was used for recovery. For a time-stamp recovery, specify the RCVUSID parameter with the *usid* value that the database was recovered to, and specify the RUNUSID parameter with the *usid* that was current when the recovery occurred. Future recoveries will ignore all records associated with *usid* values between those specified for the RCVUSID and RUNUSID parameters.

In a data sharing environment, after you notify DBRC of a nonstandard recovery or an IMS recovery, DBRC turns off the recovery-needed flag and decreases the counter in the appropriate DBDS and DB records in the RECON data set.

If you need to recover a DEDB area using the user image copy data set that was created by the DEDB Alter utility, you must issue the NOTIFY.RECOV command before you run the Database Recovery utility.

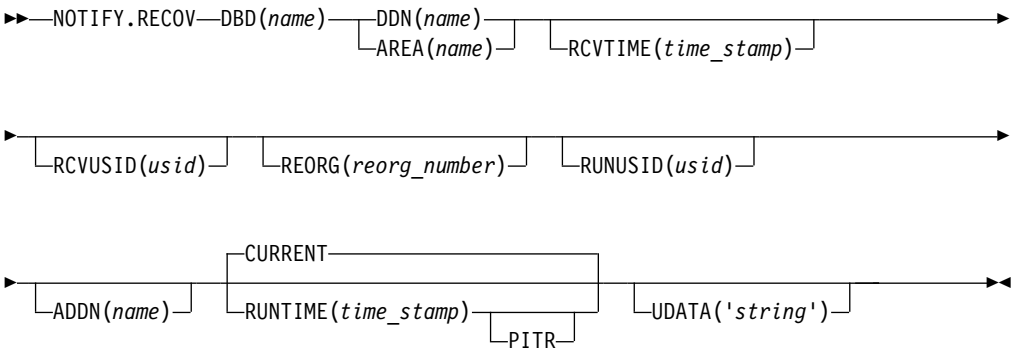
- Issue the NOTIFY.RECOV command to notify DBRC that you restored the DEDB area. The RCVTIME(*time_stamp*) is the timestamp of the user image copy that was created by the DEDB Alter utility and recorded in the RECON: `NOTIFY.RECOV DBD(name) AREA(name) RCVTIME(time_stamp)`
- Issue the GENJCL.RECOV command to generate the recovery JCL with the USEAREA keyword to indicate that no image copy data set is to be included in the generated JCL: `GENJCL.RECOV DBD(name) AREA(name) USEAREA`

Restriction: This command is not allowed for ILDS or Index DBDSs of HALDB partitions.

Subsections:

- "Syntax"
- "Keywords"
- "Example of adding DBDS recovery information to the RECON data set" on page 369

Syntax



Keywords

DBD(*name***)**
Required keyword you use to specify the database name of the DBDS or DEDB area.

Restriction for HALDB databases: When using the NOTIFY.RECOV command for HALDB databases, *name* must specify the name of a HALDB partition.

DDN(*name***) | AREA(***name***)**
Mutually exclusive, required keywords you use to specify the ddname of the DBDS or DEDB area for which DBRC is to add the database recovery record to the RECON data set.

ADDN(*name***)**
Optional keyword you use to specify the ADS DD name of the ADS for which a Fast Path DEDB area recovery record is being added to the RECON data set.

You can specify this keyword only when you specify the AREA(*name*) keyword.

CURRENT | RUNTIME(*time_stamp***) PITR**
Mutually exclusive, optional keywords you use to specify the time stamp at which the DBDS or DEDB area was recovered.

CURRENT
Specifies that the current time stamp is to be used as the time stamp of the recovery. You can add the recovery information to the RECON data set in a later step of the same job that performs the recovery if you specify CURRENT.

RUNTIME
Specifies the actual time stamp of a recovery of the DBDS or DEDB area. The time stamp must be in standard form.

The optional PITR keyword specifies a point-in-time recovery. You must use the RCVTIME keyword if you use PITR.

RCVTIME(*time_stamp***)**
Optional keyword you use to specify the point in time to which the DBDS or DEDB area was restored. It can be any time when the DBDS or area was not being updated, that is, a time that is not covered by an active ALLOC record in the RECON data set.

If you do not specify the RCVTIME keyword, you are notifying DBRC of a full recovery.

You must use RCVTIME if you use the RUNTIME keyword with the PITR keyword.

RCVUSID(*usid***)**
Optional keyword you use to specify the effective update set identifier (USID) to which the DBD or DEDB area was recovered.

The USID value you use is the one in the listing of the IMAGE record.

REORG(*reorg_number***)**
Optional keyword that specifies the database reorganization number that was effective after the database was recovered to the value specified for this parameter. This number must be 1 - 65535.

Restriction: This keyword is valid only if reorganization number verification is active in the RECON data set and the database is a HALDB partition. If the NOTIFY.RECOV command is issued as part of a database recovery, the reorganization number in the HALDB partition database record is also changed to the value given for this parameter.

RUNUSID(*usid*)
Optional keyword you use to specify the current update set identifier (*usid*) at the time the database or DEDB area was recovered. If this is a receive, RUNUSID is the *usid* value of the image copy that was used for recovery.

UDATA('*string*')
Optional keyword you use to specify up to 80 bytes of information about the identified recovery. You can use the variable field of this keyword to describe the recovery. The *string* value must be enclosed in single quotation marks if blanks or any other special characters are included.

Example of adding DBDS recovery information to the RECON data set

In this example, information about recovery of a specified DBDS is to be added to the RECON data set. The RUNTIME keyword specifies the time stamp of the recovery of the DBDS. The PITR keyword specifies a point-in-time recovery. The RCVTIME keyword specifies the time stamp to which the specified DBDS was recovered.

```
 //NFYRECOV JOB
⋮
 //SYSIN    DD   *
      NOTIFY.RECOV   DBD(DB1)  DDN(DDN1)  -
                     RUNTIME(071351015366)  -
                     RCVTIME(071350905297)  -
                     PITR
 /*
```

After the command is executed, a listing of the RECON shows the RECOV record, as illustrated as follows.

```
RECOV
 RUN     = 2007.135 10:15:36.6 -08:00     *  RUN USID      = 0000000005
 RECOV TO= 2007.135 09:05:29.7 -08:00        RECOV TO USID = 0000000004
 POINT-IN-TIME
```

Example of changed reorganization numbers in the RECON data set

These examples show records with new reorganization numbers that were added as a result of issuing NOTIFY.RECOV (external) commands.

```
RECOV
 RUN     = 2009.033 10:15:36.612345 -08:00   *  RUN USID      = 0000000005
                       CMD   PRIOR REORG# = 00009       NEW REORG# = 00010
RECOV
 RUN     = 2009.033 10:15:36.612345 -08:00   *  RUN USID      = 0000000005
 RECOV TO= 2009.033 09:05:29.712345 -08:00      RECOV TO USID = 0000000004
                       CMD   PRIOR REORG# = 00009       NEW REORG# = 00010
RECOV
 RUN     = 2009.034 10:15:36.612345 -08:00   *  RUN USID      = 0000000005
 RECOV TO= 2009.034 09:05:29.712345 -08:00      RECOV TO USID = 0000000004
 POINT-IN-TIME         CMD   PRIOR REORG# = 00009       NEW REORG# = 00010
```

Related reference:

"DBRC time stamps" on page 52

NOTIFY.REORG command

Use a NOTIFY.REORG command to add a record to the RECON data set about the reorganization of the database to which an identified DBDS belongs.

The information in the reorganization record is used by DBRC to determine which image copy data sets, change accumulation data sets, and log data sets are valid as input to a subsequent recovery of the identified DBDS.

Normally, the use of the NOTIFY.REORG command is not required because reorganization records are created automatically by IMS and the IMS utilities.

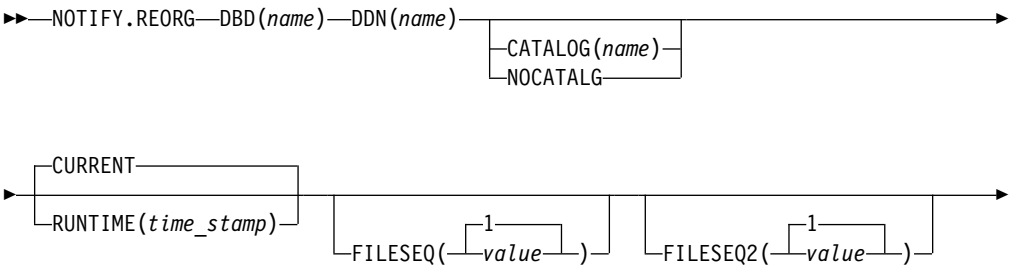
Restrictions:

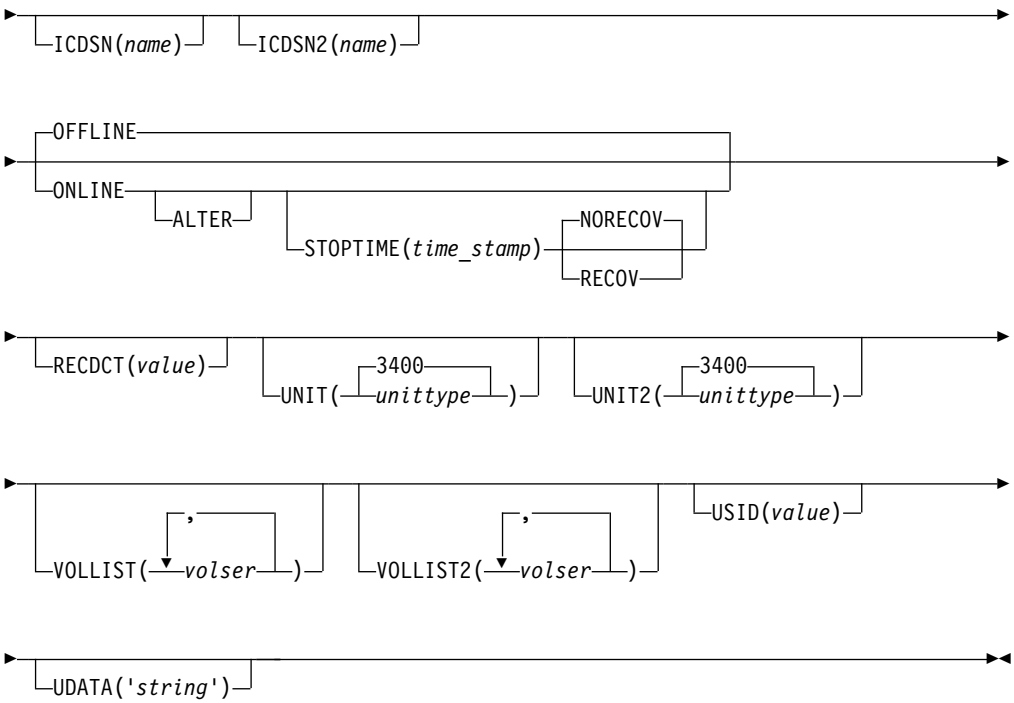
- This command should not be used following the reorganization of a Fast Path DEDB. Such databases can be recovered after a reorganization.
- This command turns on the flag needed by the image copy process in the DBDS record. You must either run an image copy or issue the CHANGE.DBDS ICOFF command to turn off the flag.
- This command can also be used to record in the RECON data set the equivalent of an image copy data set that was created for the HISAM Reorganization Reload utility. Use this command only if you are using these logs as an image copy data set.
- All optional keywords except CURRENT or RUNTIME apply only to the image copy data set that was created as part of the processing by the HISAM Reorganization Reload utility.
- You must specify a NOTIFY.REORG command for each DBDS in the database that was reorganized.
- For non-HALDB databases, either the name of an IMS catalog that is enabled to manage ACBs must be specified in the RECON or with the NOTIFY.REORG command or, if the IMS catalog does not manage ACBs, a DD statement for the IMS.DBDLIB data set must be provided in the job stream of the NOTIFY.REORG command.
- The NOTIFY.REORG command is not allowed for ILDS or Index DBDSs of HALDB partitions.

Subsections:

- "Syntax"
- "Keywords" on page 371
- "Example of adding DBDS reorganization information to the RECON data set" on page 373

Syntax





Keywords

DBD(*name*)
Required keyword you use to identify the database name of the DBDS that was reorganized.

Restriction for HALDB databases: When using the NOTIFY.REORG command for HALDB databases, *name* must specify the name of a HALDB partition.

DDN(*name*)
Required keyword you use to identify the data set ddname of the DBDS that was reorganized.

CATALOG(*name*) | NOCATALG
Mutually exclusive, optional parameters that you use to indicate whether a catalog or a DBDLIB is to be used during the NOTIFY.REORG command. If these parameters are omitted, then the specification in the RECON header determines whether IMS uses an IMS catalog or a DBDLIB for the NOTIFY.REORG.

For databases that are defined using Database Definition Language (DDL), DBRC must access a catalog.

CATALOG(*name*)
Specifies that the NOTIFY.REORG command uses the specified IMS catalog instead of the DBD library.

Specify *name* as an eight-character alphanumeric string that matches the name of a HALDB IMS catalog database that is registered in the RECON data set. Any IMS catalog name that is less than eight-characters is padded with blanks.

The IMS catalog name is validated during the NOTIFY.REORG command. If the catalog does not exist or is not defined as a HALDB database, the command fails.

NOCATALG
Specifies that the NOTIFY.REORG command uses the DBD library instead of the IMS catalog.

CURRENT | RUNTIME(*time_stamp*)
Mutually exclusive, optional keywords you use to specify the time stamp of the reorganization of the identified DBDS.

CURRENT
Specifies that the current time stamp is to be placed in the reorganization record. You can specify a NOTIFY.REORG command as a later step in the same job that performs the reorganization if you specify CURRENT.

RUNTIME
Specifies that the actual time stamp of the reorganization is to be placed in the reorganization record. The time stamp must be in standard form.

FILESEQ(1 | *value*)
Optional keyword you use to specify the file sequence number of the identified DBDS that was included in the logs that were used as input to a run of the HISAM Reorganization Reload utility. The description of the ICDSN keyword contains information about the log data set with which this keyword is associated. You use this keyword only when the VOLLIST keyword has also been specified.

FILESEQ2(1 | *value*)
Optional keyword you use to specify the file sequence number of the identified DBDS when it was included in the logs that were used as input to the HISAM Reorganization Reload utility. The description of the ICDSN2 keyword contains information about the log data set with which this keyword is associated. You use this keyword only when the VOLLIST2 keyword has also been specified.

ICDSN(*name*)
Optional keyword you use to specify the data set name of the image copy data set that was created as part of a HISAM reorganization of a database. (If you reorganized your database using the HISAM Reorganization Reload utility, the logs that were used as input to that utility can be used as image copy data sets.)

You can specify an ICDSN keyword only if the corresponding DBDS was identified to the RECON data set with the NOREUSE attribute in an `INIT.DBDS` command.

ICDSN2(*name*)
Optional keyword you use to specify the data set name of the duplicate image copy data set that was created as part of a HISAM reorganization of a database. (If you reorganized your database using the HISAM Reorganization Reload utility, the logs that were used as input to that utility can be used as image copy data sets.)

You can specify an ICDSN2 keyword only if you have specified an ICDSN keyword.

OFFLINE | ONLINE
Mutually exclusive, optional keywords that specify whether the reorganization is a HALDB OLR or an offline reorganization.

OFFLINE
Specifies that the DBDS was reorganized offline.

ONLINE
Specifies that the database is a HALDB and that it was reorganized online.

ALTER
Specifies that the reorganization altered the database structure.

STOPTIME(*time_stamp*)
Optional keyword that specifies the time when a HALDB OLR completed successfully.

NORECOV | RECOV
Mutually exclusive, optional keywords that specify whether the online reorganization can be used as input for recovery.

NORECOV
Specifies that the HALDB OLR cannot be used as input to recovery.

RECOV
Specifies that the HALDB OLR may be used as input to recovery.

RECDCT(*value*)
Optional keyword you use to specify the number of records that are contained in the identified DBDS. *value* must be a decimal number from 1 to 2 147 483 647.

UNIT(3400 | *unittype*)
Optional keyword you use to specify the unit type of the volumes on which the image copy data set resides. The unit type can be up to 8 alphanumeric characters.

UNIT2(3400 | *unittype*)
Optional keyword you use to specify the unit type of the volumes on which the duplicate image copy data set resides. The unit type can be up to 8 alphanumeric characters.

VOLLIST(*volser*)
Optional keyword you use to specify the volume serial numbers of the volumes on which the image copy data set identified by the ICDSN keyword resides. You can specify up to 255 volume serial numbers for *volser*; each *volser* can be up to 6 alphanumeric characters.

VOLLIST2(*volser*)
Optional keyword you use to specify the volume serial numbers of the volumes on which the duplicate image copy data set, identified by the ICDSN2 keyword, resides. You can specify up to 255 volume serial numbers for *volser*; each can be up to six alphanumeric characters.

USID(*value*)
Optional keyword you use to specify the update set identifier of the database or area when the reorganization occurred.

UDATA('*string*')
Optional keyword you use to specify up to 80 bytes of information about the reorganization. You can use the variable field of this keyword to describe the reorganization. The *string* value must be enclosed in single quotation marks if blanks or any other special characters are included.

Example of adding DBDS reorganization information to the RECON data set

In this example, information about a reorganization of a specified DBDS is to be added to the RECON data set. The DBDLIB data set is specified, because DBRC requests a search of it to verify that the reorganization occurred. The names of two

image copy data sets for the reorganized DBDS are given. They both follow the data set naming convention, and a list of volumes is provided for both image copy data sets.

```
 //NFYREORG JOB
 .
 .
 .
 //IMS      DD   DSN=IMS.DBDLIB,DISP=SHR
 //SYSIN    DD   *
      NOTIFY.REORG DBD(DB1)  DDN(DD1) -
                   ICDSN(IMS.DB1.DD1.IC.ICDSN) -
                   VOLLIST(VOL001,VOL002,VOL003) FILESEQ(4) -
                   ICDSN2(IMS.DB1.DD1.IC2.ICDSN2) -
                   VOLLIST2(VOL004,VOL005,VOL006,VOL007) -
                   FILESEQ2(4) RECDCT(12345)
 /*
```

Example using the DBDLIB instead of the default catalog name

```
 //NFYREORG JOB
 .
 .
 .
 //IMS      DD   DSN=IMS.DBDLIB,DISP=SHR
 //SYSIN    DD   *
      NOTIFY.REORG DBD(DB1)  DDN(DD1) -
                   ICDSN(IMS.DB1.DD1.IC.ICDSN) -
                   VOLLIST(VOL001,VOL002,VOL003) FILESEQ(4) -
                   ICDSN2(IMS.DB1.DD1.IC2.ICDSN2) -
                   VOLLIST2(VOL004,VOL005,VOL006,VOL007) -
                   FILESEQ2(4) RECDCT(12345) –
                 NOCATALG
 /*
```

Example using the catalog name GREGCD00 instead of the default catalog name

```
//NFYREORG JOB
.
.
.
//SYSIN    DD   *
      NOTIFY.REORG DBD(DB1)  DDN(DD1) -
                   ICDSN(IMS.DB1.DD1.IC.ICDSN) -
                   VOLLIST(VOL001,VOL002,VOL003) FILESEQ(4) -
                   ICDSN2(IMS.DB1.DD1.IC2.ICDSN2) -
                   VOLLIST2(VOL004,VOL005,VOL006,VOL007) -
                   FILESEQ2(4) RECDCT(12345) –
                 CATALOG(GREGCD000)

 /*
```

Related reference:

"DBRC time stamps" on page 52

NOTIFY.SECLOG command (for OLDS)

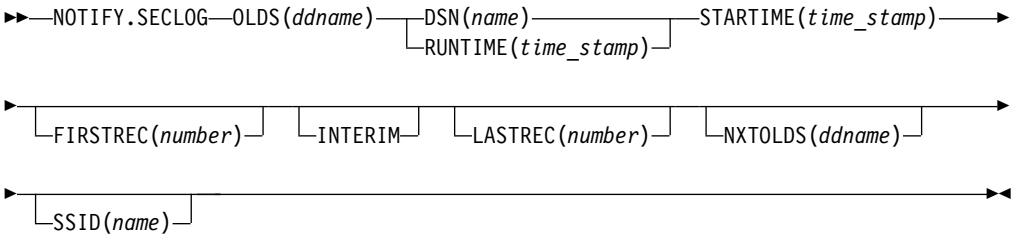
Use the NOTIFY.SECLOG command to add information about a secondary OLDS to the RECON data set and to manually create an ISECOLDS record in the RECON data set.

The RECON data set must already contain a PRIOLDS record with the same SSID and STARTIME. This is information that could not be added from the IMS log data exit routines. This command is not normally required.

Subsections:

- "Syntax"
- "Keywords"
- "Examples" on page 377

Syntax



Keywords

OLDS(*ddname***)**
Required keyword you use to specify that a record is to be created or updated in the RECON data set for the OLDS.

ddname is the ddname that the IMS online control region used when it used the OLDS.

DSN(*name***) | RUNTIME(***time_stamp***)**
Mutually exclusive, required keywords.

DSN
Specifies the data set name of the secondary OLDS for which an online log record is being created in the RECON data set.

RUNTIME
Specifies the time stamp of an open or close operation of the specified secondary OLDS. The time stamp must be in standard form.

DSN and RUNTIME are used in conjunction with the STARTIME, FIRSTREC, LASTREC, and NXTOLDS keywords to identify which type of secondary online log data set entry is to be added to the RECON data set. The following table indicates the keyword combinations that correspond to the type of secondary online log data set entry.

Table 18. Keywords of NOTIFY.SECLOG command (for OLDS) for open, switch, and close

Type of online log entry	Required keywords
OLDS Open	STARTIME, DSN, FIRSTREC
OLDS Switch	FIRSTREC, STARTIME, DSN, NXTOLDS
OLDS Close	LASTREC, STARTIME, RUNTIME

For each secondary OLDS, you must issue a separate NOTIFY.SECLOG command for open, switch, and close operations.

STARTIME(*time_stamp***)**
Required keyword you use to specify the starting time of a secondary OLDS. The time stamp must be in standard form.

For a description of the use of this keyword with other keywords in the NOTIFY.SECLOG command, see Table 18.

FIRSTREC(*number*)
Optional keyword you use to specify the log record sequence number of the first log record of the OLDS. For the first OLDS of the SECLOG, FIRSTREC corresponds to the first log record that was written during initialization of the IMS subsystem.

FIRSTREC is required for OLDS OPEN and SWITCH commands. It specifies the first log record sequence number on the OLDS being opened. It is invalid for a CLOSE command.

The log record sequence number can be one of the following:

- A hexadecimal number

  This number is 1 to 16 characters, enclosed in single quotation marks and preceded by the letter, X. For example: `FIRSTREC(X'10B9C')`.
- A decimal number

  This number is a decimal number from 0 to (2**64)-1, without delimiters. For example: `FIRSTREC(68508)`.

In either case, leading zeros can be omitted.

INTERIM
Optional keyword you use to specify that an interim log data set record is to be created. Before you create an interim log data set, you must create a secondary OLDS.

LASTREC(*number*)
Optional keyword you use to specify the log record sequence number of the last log record of the OLDS.

LASTREC is required for the OLDS CLOSE command. It is optional for the SWITCH command; if it is omitted, the FIRSTREC value minus 1, is recorded for the OLDS being closed. It is invalid for an OPEN command.

The log record sequence number can be one of the following:

- A hexadecimal number

  This number is 1 to 16 characters, enclosed in single quotation marks and preceded by the letter, X. For example: `LASTREC(X'10B9C')`.
- A decimal number

  This number is a decimal number from 0 to (2**64)-1, without delimiters. For example: `LASTREC(68508)`.

In either case, leading zeros can be omitted.

NXTOLDS(*ddname*)
Optional keyword you use when the RECON data set is to be updated to reflect an OLDS switch. The current OLDS is closed and an IMS online control region opens a new OLDS. *ddname* is the DD statement of the OLDS being opened. You specify the OLDS being closed with the OLDS(*ddname*) keyword. Use the DSN keyword to specify the data set name of the OLDS being opened. Use the STARTIME keyword to specify the close time of the OLDS being closed and the open time of the OLDS being opened.

SSID(*name*)
Optional keyword you use to specify the name of the IMS subsystem that created the log data set.

The SSID is an eight-character alphanumeric string that represents a valid IMS subsystem identification name. If you do not specify SSID, DBRC uses the default subsystem identifier in the RECON header record. You use the

INIT.RECON or CHANGE.RECON command to set the default subsystem identifier in the RECON header record. If you have not specified a default in the RECON header record, you must specify SSID.

Examples

Here are some examples of using the NOTIFY.SECLOG (for OLDS) command.

Example of creating the ISECOLDS record that corresponds to the OLDS

In this example, an ISECOLDS record corresponding to the OLDS is created.

```
//NFYSECLG JOB
⋮
//SYSIN    DD   *
    NOTIFY.SECLOG  SSID(IMSA) OLDS(DFSOLS03) -
                   DSN(IMS.INTERIM.LOG) -
                   STARTIME(073220522348) -
                   INTERIM
/*
```

Example of creating a SECOLDS record for 2 secondary OLDSs

In this example, you create a SECOLDS record for two secondary OLDSs that belong to IMS online subsystem IMSA. Both secondary OLDSs are closed. The first STARTIME keyword specifies the time stamp of the opening of the primary OLDS. The DSN keyword indicates that information added relates to the opening of the OLDS. NXTOLDS indicates an OLDS switch. The second STARTIME keyword and second DSN indicate the start time and DSN of the next OLDS. The third STARTIME keyword indicates the start time of the OLDS to be closed. The RUNTIME keyword is the time stamp of the closing volume.

```
NOTIFY.SECLOG SSID(IMSA) STARTIME(072171212120) OLDS(DFSOLS01)-
       DSN(IMS.OLSS01)
NOTIFY.SECLOG SSID(IMSA) STARTIME(072181212120) OLDS(DFSOLS01)-
       DSN(IMS.OLSS02) NXTOLDS(DFSOLS02)
NOTIFY.SECLOG SSID(IMSA) STARTIME(072181212120) OLDS(DFSOLS02)-
       RUNTIME(072191010101)
```

Related reference:

"DBRC time stamps" on page 52

NOTIFY.SECLOG command (for RLDS)

Use a NOTIFY.SECLOG command to add information about a secondary RLDS to the RECON data set and to manually create an ISECLOG record in the RECON data set.

The RECON data set must already contain a PRILOG with the same SSID and STARTIME. This is information that could not be added from the IMS log data exit routines. This command is not normally required.

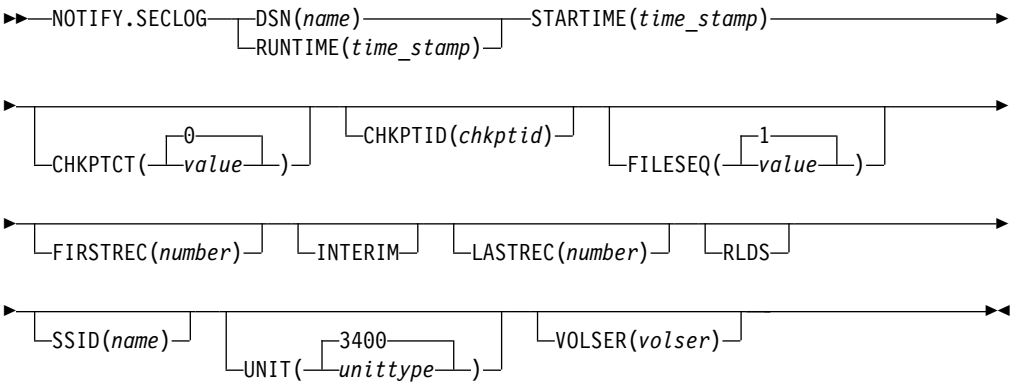
This command adds or completes a data set entry in the Primary Log record. If you are modifying an existing completed data set entry, you should use the CHANGE.SECLOG (RLDS) command.

Subsections:

- "Syntax" on page 378
- "Keywords" on page 378

• "Examples" on page 380

Syntax



Keywords

DSN(*name***) | RUNTIME(***time_stamp***)**
Mutually exclusive, required keywords.

DSN
Specifies the data set name of the secondary RLDS for which a recovery log record is being created in the RECON data set.

RUNTIME
Specifies the time stamp of an open, close, or EOV operation of the specified secondary RLDS. The time stamp must be in standard form.

DSN and RUNTIME are used in conjunction with the STARTIME and VOLSER keywords to identify which type of secondary recovery log data set entry is to be added to the RECON data set. The following table indicates the keyword combinations that correspond to the type of secondary recovery log data set entry.

Table 19. Keywords of NOTIFY.SECLOG command (for RLDS) for open, EOV, and close

Type of recovery log entry	Required keywords
RLDS Open	STARTIME, DSN, VOLSER
RLDS EOV	STARTIME, VOLSER, RUNTIME
RLDS Close	STARTIME, RUNTIME

For each secondary RLDS, you must issue a separate NOTIFY.SECLOG command for open, zero or more ends-of-volume, and close processing.

STARTIME(*time_stamp***)**
Required keyword you use to specify the starting time of a secondary RLDS. The time stamp must be in standard form.

If you issue a subsequent STARTIME keyword, that time is the start time of the volume. For a description of the use of the STARTIME keyword with other keywords in the NOTIFY.SECLOG command, see Table 19.

CHKPTCT(0 | *value*)
Optional keyword you use to change the number of checkpoints completed on the RLDS volumes.

The valid values for CHKPTCT are:

0 No checkpoints in the RLDS volume

1 A single checkpoint in the RLDS volume

2 More than one checkpoint in the RLDS volume

IMS uses the value of CHKPTCT to determine which logs are necessary to recover a Fast Path area with concurrent image copy.

CHKPTID(*chkptid*)
Optional keyword you use to specify the oldest checkpoint ID for an active PST on an RLDS volume. The checkpoint ID must be in standard form for a time stamp.

FILESEQ(1 | *value*)
Optional keyword you use to specify the file sequence number of the secondary RLDS that is identified. You specify this keyword only if you have also specified the VOLSER keyword.

FIRSTREC(*number*)
Optional keyword you use to specify the log record sequence number of the first log record of the RLDS. For the first RLDS of the SECLOG, it corresponds to the first log record that was written during initialization of the IMS subsystem.

FIRSTREC is required if DSN is specified and is invalid if RUNTIME is specified.

The log record sequence number can be one of the following:

- A hexadecimal number

  This number is 1 to 16 characters, enclosed in single quotation marks and preceded by the letter, X. For example: `FIRSTREC(X'10B9C')`.
- A decimal number

  This number is a decimal number from 0 to (2**64)-1, without delimiters. For example: `FIRSTREC(68508)`.

In either case, leading zeros can be omitted.

INTERIM
Optional keyword you use to specify that an interim log data set record is to be created.

LASTREC(*number*)
Optional keyword you use to specify the log record sequence number of the last log record of the RLDS.

LASTREC is required if RUNTIME is specified and VOLSER is not specified (that is, on a Close call). LASTREC is invalid if DSN is specified.

The log record sequence number can be one of the following:

- A hexadecimal number

  This number is 1 to 16 characters, enclosed in single quotation marks and preceded by the letter, X. For example: `LASTREC(X'10B9C')`.
- A decimal number

  This number is a decimal number from 0 to (2**64)-1, without delimiters. For example: `LASTREC(68508)`.

In either case, leading zeros can be omitted.

RLDS
Optional keyword you use to specify that a record is to be created or updated in the RECON data set for an IMS RLDS.

SSID(*name*)
Optional keyword you use to specify the name of the IMS subsystem that created the log data set.

The SSID is an eight-character alphanumeric string that represents a valid IMS subsystem identification name. If you do not specify SSID, DBRC uses the default subsystem identifier in the RECON header record. You use the INIT.RECON or CHANGE.RECON command to set the default subsystem identifier in the RECON header record. If you have not specified a default in the RECON header record, you must specify SSID.

UNIT(3400 | *unittype*)
Optional keyword you use to specify the unit type of the volumes on which the secondary RLDSs reside. You only specify the UNIT keyword if you specify the DSN keyword. The unit type can be up to eight alphanumeric characters.

VOLSER(*volser*)
Optional keyword you use to specify the volume serial number of the recovery log volume that is being recorded for the identified secondary RLDS. For an EOV notification, this volume serial number is that of the volume being started. Table 19 on page 378 identifies when you use the VOLSER keyword.

Examples

Here are some examples of using the NOTIFY.SECLOG (for RLDS) command.

Example of adding secondary RLDS information to the RECON data set

In this example, information about a secondary RLDS is to be added to the RECON data set. The STARTIME keyword identifies the secondary RLDS by its opening time stamp. The VOLSER and DSN keywords indicate that the information to be added relates to the opening of the primary RLDS. The first RUNTIME keyword specifies the time stamp of the EOV of the secondary RLDS. The second RUNTIME keyword specifies the time stamp of the closing of the secondary RLDS.

```
//NFYSECLG JOB
.
.
.
//SYSIN    DD   *
     NOTIFY.SECLOG  RLDS STARTIME(07067020101023456)  -
                    DSN(DSN003)  VOLSER (VOL001)
     NOTIFY.SECLOG  RLDS STARTIME(07067020101023456)  -
                    RUNTIME(070680204500)  VOLSER(VOL002)
     NOTIFY.SECLOG  RLDS STARTIME(07067020101023456)  -
                    RUNTIME(070682030000)
/*
```

Example of adding interim secondary RLDS information to the RECON data set

In this example, information about the interim secondary RLDS is to be added to the RECON data set. The STARTIME keyword specifies the time stamp of the opening of the interim-secondary RLDS, and the RUNTIME keyword specifies the time stamp of the closing of the interim-secondary RLDS.

```
//NFYSECLG JOB
⋮
//SYSIN    DD   *
    NOTIFY.SECLOG    RLDS RUNTIME(072561630000)-
                     STARTIME(07254123456123456) INTERIM
/*
```

Related reference:

"DBRC time stamps" on page 52

NOTIFY.SECLOG command (for SLDS)

Use the NOTIFY.SECLOG command to add information about a secondary SLDS to the RECON data set and to manually create an ISECSLDS record in the RECON data set.

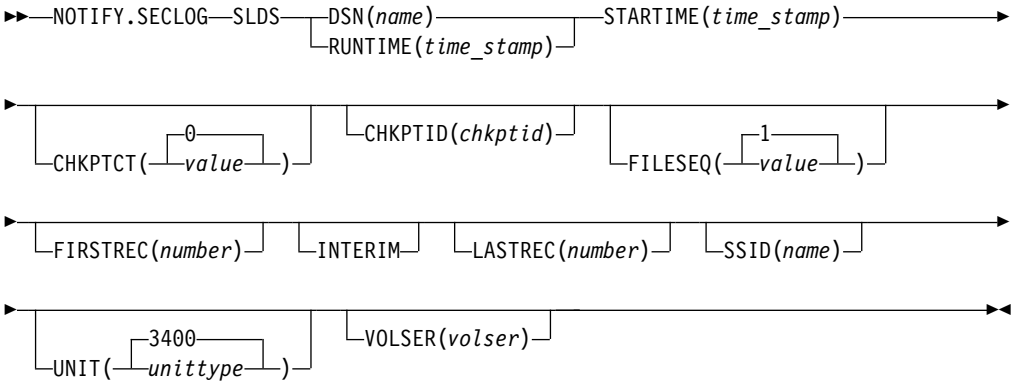
The RECON data set must already contain a primary log data set record with the same SSID and STARTIME. This is information that could not be added from the log data exists of the IMS system. This command is not normally required.

This command adds or completes a data set entry in the Primary or Secondary Log record. If you are modifying an existing completed data set entry, use the CHANGE.SECLOG command (for SLDS).

Subsections:

- "Syntax"
- "Keywords"
- "Example of adding secondary SLDS information to the RECON data set" on page 384

Syntax



Keywords

SLDS
Required keyword that specifies that an SLDS record will be created or updated.

Note: Failure to specify either the SLDS keyword results in an attempt to create or update an RLDS record.

DSN(*name***) | RUNTIME(***time_stamp***)**
Mutually exclusive, required keywords.

DSN
Specifies the data set name of the secondary SLDS for which a system log record is being created in the RECON data set.

RUNTIME
Specifies the time stamp of an open, close, or EOV operation of the specified secondary SLDS. The time stamp must be in standard form.

DSN and RUNTIME are used in conjunction with the STARTIME and VOLSER keywords to identify which type of secondary system log data set entry is to be added to the RECON data set. The following table indicates the keyword combinations that correspond to the type of secondary system log data set entry.

Table 20. Keywords of NOTIFY.SECLOG (for SLDS) command for open, EOV, and close

Type of system log entry	Required keywords
SLDS Open	STARTIME, DSN, VOLSER
SLDS EOV	STARTIME, VOLSER, RUNTIME
SLDS Close	STARTIME, RUNTIME

For each secondary SLDS, you must issue a separate NOTIFY.SECLOG command for open, zero or more ends-of-volume, and close processing.

STARTIME(*time_stamp***)**
Required keyword that you use to specify the starting time of a secondary SLDS. The time stamp must be in standard form. For a description of the use of the STARTIME keyword with other keywords in the NOTIFY.SECLOG command, see Table 20.

CHKPTCT(0 | *value*)
Optional keyword that you use to change the number of checkpoints completed on the SLDS volumes.

The valid values for CHKPTCT are:

0 No checkpoints in the SLDS volume

1 A single checkpoint in the SLDS volume

2 More than one checkpoint in the SLDS volume

IMS uses the value of CHKPTCT to determine which logs are necessary to recover a Fast Path area with concurrent image copy.

CHKPTID(*chkptid***)**
Optional keyword that you use to specify the oldest checkpoint ID for an active PST on an SLDS volume. The checkpoint ID must be in standard form for a time stamp.

FILESEQ(1 | *value*)
Optional keyword that you use to specify the file sequence number of the secondary SLDS that is identified. You specify this keyword only if you specify the VOLSER keyword.

FIRSTREC(*number***)**
Optional keyword that you use to specify the log record sequence number of the first log record of the SLDS. For the first SLDS of the SECSLD , FIRSTREC corresponds to the first log record that was written during initialization of the IMS subsystem.

FIRSTREC is required if DSN is specified and is invalid if RUNTIME is specified.

The log record sequence number can be one of the following:

- A hexadecimal number

  This number is 1 to 16 characters, enclosed in single quotation marks and preceded by the letter, X. For example: `FIRSTREC(X'10B9C')`.
- A decimal number

  This number is a decimal number from 0 to (2**64)-1, without delimiters. For example: `FIRSTREC(68508)`.

In either case, leading zeros can be omitted.

INTERIM
Optional keyword that you use to specify that an interim log data set record is to be created.

LASTREC(*number*)
Optional keyword that you use to specify the log record sequence number of the last log record of the SLDS.

LASTREC is required if RUNTIME is specified and VOLSER is not specified (that is, on a close call). LASTREC is invalid if DSN is specified.

The log record sequence number can be one of the following:

- A hexadecimal number

  This number is 1 to 16 characters, enclosed in single quotation marks and preceded by the letter, X. For example: `LASTREC(X'10B9C')`.
- A decimal number

  This number is a decimal number from 0 to (2**64)-1, without delimiters. For example: `LASTREC(68508)`.

In either case, leading zeros can be omitted.

SSID(*name*)
Optional keyword that you use to specify the name of the IMS subsystem that created the log data set.

The SSID is an eight-character alphanumeric string that represents a valid IMS subsystem identification name. If you do not specify SSID, DBRC uses the default subsystem identifier in the RECON header record. You use the INIT.RECON or CHANGE.RECON command to set the default subsystem identifier in the RECON header record. If you have not specified a default in the RECON header record, you must specify SSID.

UNIT(3400 | *unittype*)
Optional keyword that you use to specify the unit type of the volumes on which the secondary SLDSs reside. You only specify the UNIT keyword if you specify the DSN keyword. The unit type can be up to 8 alphanumeric characters.

VOLSER(*volser*)
Optional keyword that you use to specify the volume serial number of the system log volume that is being recorded for the identified secondary SLDS. For an EOV notification, this volume serial number is that of the volume being started. Table 20 on page 382 identifies when you use the VOLSER keyword.

Example of adding secondary SLDS information to the RECON data set

In this example, information about a secondary SLDS is to be added to the RECON data set. The STARTIME keyword identifies the secondary SLDS by its opening time stamp. The VOLSER and DSN keywords indicate that the information to be added relates to the opening of the secondary SLDS. The first RUNTIME keyword specifies the time stamp of the EOV of the secondary SLDS. The second RUNTIME keyword specifies the time stamp of the closing of the secondary SLDS.

```
//NFYSECLG JOB
⋮
//SYSIN    DD   *
     NOTIFY.SECLOG   SLDS STARTIME(07067020101023456)  -
                     SSID(IMSC) DSN(DSN006) VOLSER(VOL009)
     NOTIFY.SECLOG   SLDS STARTIME(07067020101023456)  -
                     RUNTIME(070680204500)  VOLSER(VOL003)
     NOTIFY.SECLOG   SLDS STARTIME(07067020101023456)  -
                     RUNTIME(070682030000)
/*
```

Related reference:

"DBRC time stamps" on page 52

"NOTIFY.PRILOG command (for RLDS)" on page 359

NOTIFY.SUBSYS command

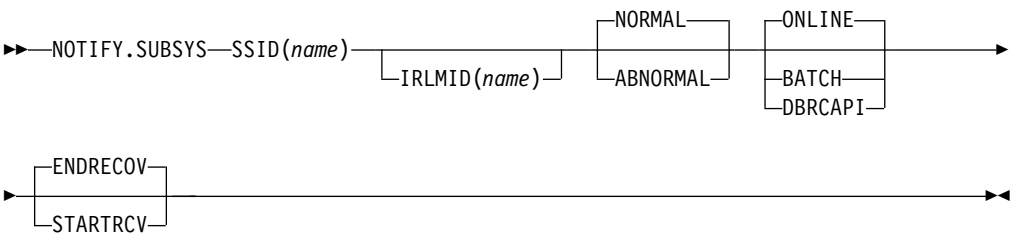
Use the NOTIFY.SUBSYS command to create a subsystem entry in the RECON data set.

A check is made to ensure that a subsystem entry for the specified subsystem does not exist in the RECON data set. This command is not normally required.

Subsections:

- "Syntax"
- "Keywords"
- "Example of adding a new subsystem record to the RECON data set" on page 385

Syntax



Keywords

SSID(*name*)

Required keyword you use to specify the name of the subsystem for which information is to be added. The SSID is an eight-character alphanumeric string that represents a valid z/OS and IMS subsystem identification name.

IRLMID(*name*)
Optional keyword you use to specify the name of the IRLM with which the subsystem is communicating. The IRLMID is a five-character alphanumeric string.

If IRLMID is not specified, the subsystem is not using an IRLM.

Note: The BACKIRLM keyword is not a valid keyword for the NOTIFY.SUBSYS command.

NORMAL | ABNORMAL
Mutually exclusive, optional keywords you use to specify the status of the subsystem.

NORMAL
Specifies that the previous run of the subsystem ended normally and that the subsystem is to continue normal processing.

ABNORMAL
Specifies that the previous run of the subsystem ended abnormally and recovery processing is required.

ONLINE | BATCH | DBRCAPI
Mutually exclusive, optional keywords you use to specify the type of subsystem from which notification is made.

ONLINE
Specifies that notification is made from an online subsystem.

BATCH
Specifies that notification is made from a batch subsystem.

DBRCAPI
Specifies that notification is made from a DBRC API application. IRLMID, NORMAL, ABNORMAL, ENDRECOV and STARTRCV are all not allowed when DBRCAPI is specified.

ENDRECOV | STARTRCV
Mutually exclusive, optional keywords you use to specify the sign-on state of the subsystem.

ENDRECOV
Specifies that the subsystem has signed on normally or that a sign-on recovery-complete call was successful.

STARTRCV
Specifies that the subsystem has signed on for recovery-start processing.

Example of adding a new subsystem record to the RECON data set

In this example, a new subsystem record identified by the SSID keyword is added to the RECON data set. In addition, the subsystem record is marked as online.

```
//NOTIFYSS  JOB
.
.
.
//SYSIN     DD   *
   NOTIFY.SUBSYS  SSID(IMS34)  ONLINE
/*
```

NOTIFY.UIC command

Use the NOTIFY.UIC command to add information to the RECON data set about a nonstandard image copy data set related to the DBDS or DEDB area that is identified in the command.

A nonstandard image copy data set is one that was not created by the supported image copy utility such as, one created through a tape dump of the DASD volume that contains the identified DBDS or DEDB area. Using the NOTIFY.UIC command is the only way you can record in the RECON data set the existence of nonstandard image copy data sets.

You cannot issue this command for a DBDS defined with the REUSE attribute.

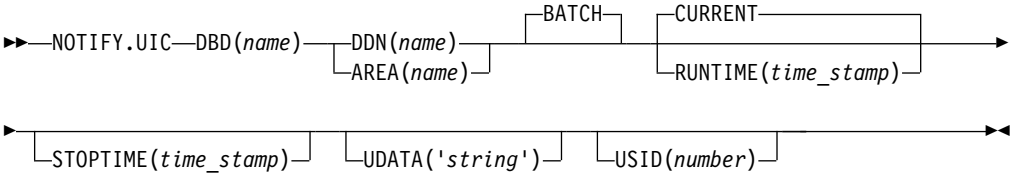
Restrictions:

- A nonstandard image copy data set cannot be used as input to the Database Recovery utility. For more information, see Database Recovery utility (DFSURDB0).
- This command is not allowed for ILDS or Index DBDSs of HALDB partitions.
- The NOTIFY.UIC command will fail if it is attempting to record an image copy of a HALDB that has an online reorganization in progress. This is because an image copy of a HALDB cannot be taken while the HALDB is being reorganized online and the reorganization has not completed (either it is in progress or was terminated).

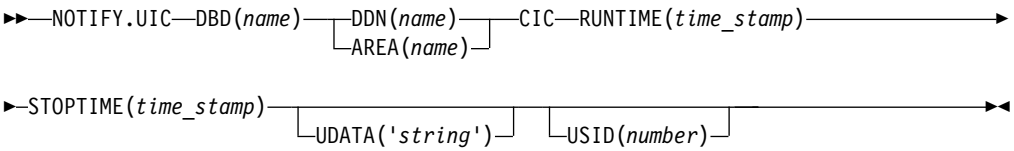
Subsections:

- "Syntax"
- "Keywords"
- "Examples of adding nonstandard ICDSN information to the RECON data set" on page 388

Syntax



Or



Keywords

BATCH | CIC

Mutually exclusive, optional parameters you use to indicate whether an image

copy created by means other than the IMS image copy utilities is a clean image copy or whether it is a concurrent image copy.

BATCH
Indicates that an image copy created by means other than the IMS image copy utilities is a clean image copy.

CIC
Indicates that an image copy created by means other than the IMS image copy utilities is a concurrent image copy. If you specify CIC, you must also specify the RUNTIME and STOPTIME keyword. The CIC keyword cannot be used with VSAM KSDS databases.

DBD(*name***)**
Required keyword you use to specify the database name of the DBDS or DEDB area for which the nonstandard image copy data set was created.

Restriction for HALDB databases: When using the NOTIFY.UIC command for HALDB databases, *name* must specify the name of a HALDB partition.

DDN(*name***) | AREA(***name***)**
Mutually exclusive, required keywords you use to identify, by its name, the DBDS or DEDB area for which the nonstandard image copy data set was created.

CURRENT | RUNTIME(*time_stamp***)**
Mutually exclusive, optional keywords you use to specify the time stamp of the creation of the nonstandard image copy data set.

CURRENT
Specifies that the current time stamp is to be used as the time stamp of the creation of the specified image copy data set. You can create the nonstandard image copy data set and record its creation in the RECON data set as separate steps of a single job if you specify CURRENT.

CURRENT is applicable only to NOTIFY.UIC BATCH commands.

RUNTIME
Specifies the actual time stamp of the creation of the identified nonstandard image copy data set. The time stamp must be in standard form.

STOPTIME(*time_stamp*)
Specifies the stop time of a concurrent image copy. If you specify CIC, you must also specify this keyword. The time stamp must be in standard form. If CIC is not specified, STOPTIME will be ignored

UDATA('*string***')**
Optional keyword you use to specify up to 80 bytes of information about the identified, nonstandard image copy data set. You can use the variable field of this keyword to describe how the nonstandard image copy data set was created. The *string* value must be enclosed in single quotation marks if blanks or any other special characters are included.

USID(*number***)**
Optional keyword you use to specify the value of the update set identifier of the database or area when the image copy data set was created.

Examples of adding nonstandard ICDSN information to the RECON data set

In this example, information about a nonstandard image copy data set is to be added to the RECON data set. The RUNTIME keyword specifies the time stamp of the creation of the nonstandard image copy data set. The UDATA keyword specifies the user data to be recorded in the record in the RECON data set that is updated by this command.

```
 //NFYUIC   JOB
.
.
.
 //SYSIN    DD   *
      NOTIFY.UIC DBD(DB1)  DDN(DD1)  -
                 RUNTIME(070670201010)  -
                 UDATA('DUMP OF VOLUME VOL001 AT 070670201010')
 /*
```

In this example, information about a nonstandard concurrent image copy data set is added to the RECON data set.

```
//NFYUIC JOB



//SYSIN DD *
NOTIFY.UIC CIC -
DBD(DB1) DDN(DD1) -
RUNTIME(070670201010) -
STOPTIME(070670203115) -
UDATA('CIC DUMP OF VOLUME VOL001 AT 20201010')
/*
```

Related reference:

"DBRC time stamps" on page 52

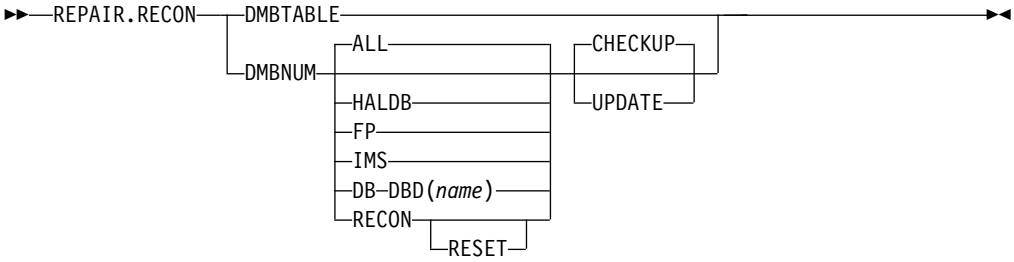
Chapter 17. REPAIR.RECON command

Use the REPAIR.RECON command to rebuild a corrupted or missing DMB Table record or to check for and correct inconsistencies in the data management block (DMB) numbers of the RECON data set.

Subsections:

- "Syntax"
- "Keywords"
- "Usage notes" on page 390
- "Example of checking for inconsistencies in the DMB number representation" on page 391

Syntax



Keywords

The following keywords are valid for the REPAIR.RECON command:

DMBTABLE | DMBNUM

DMBTABLE
Rebuilds the DMB Table record in the RECON data set. If the DMB Table record is missing, it is created. The DMB Table record indicates which data management block (DMB) numbers are assigned to existing databases. All database records are read during the rebuild process. Any inconsistencies that are found with the DMB numbers are reported, and they are corrected if the database or area is unauthorized.

DMBNUM
Checks for or corrects inconsistencies in the internal representation of the DMB number in various records in the RECON data set. You can specify all types of database records, certain types of database records, a specific database record, or the RECON header.

ALL | HALDB | FP | IMS | DB DBD(*name*) | RECON RESET

ALL
Process all types of database records. **ALL** is the default.

HALDB
Process all HALDB database records.

FP Process all Fast Path database and area records.

© Copyright IBM Corp. 1974, 2017

IMS
Process all IMS database records, except HALDB database records.

DB Process a specific database record.

DBD(*name***)**
When you specify **DB**, a required keyword that you use to indicate the HALDB, Fast Path DEDB, or IMS database to process.

RECON
Process only the RECON header.

RESET
When you specify **RECON**, an optional keyword that you specify to reset the DMB# and LAST USED DMB# fields in the RECON header.

When a RECON DMB Table record exists, both of these fields are reset to 32,767. The RECON DMB Table record is used to determine the next DMB number to be assigned when the next database is registered. When a RECON DMB Table record does not exist and no databases are registered, the values of the fields are reset to the same values that are used when a RECON data set is initialized.

CHECKUP | **UPDATE**
Check or correct the DMB numbers.

CHECKUP
Check the DMB numbers for inconsistencies. Any inconsistencies that are found are reported. **CHECKUP** is the default.

UPDATE
Correct any inconsistencies in the DMB numbers. Any inconsistencies that are found are reported, and they are fixed if the database or area is unauthorized.

Usage notes

If an error is found when the command is run, the command continues processing to report all of the errors. If the command completes with a return code of 12 or above, no records are changed.

If parallel RECON access is used, RECON activity for all DBRC instances is quiesced for the following commands:

- REPAIR.RECON DMBTABLE
- REPAIR.RECON DMBNUM ALL | HALDB| FP | IMS

A check for the existence of retained locks that are related to the Copy1 and Copy2 of the RECON data sets is completed before the REPAIR.RECON command is invoked. If any retained locks exist, the command fails and a message is issued.

In parallel RECON access mode, any retained locks that exist for the RECON data sets must be resolved. Ensure that the SMS server (SMSVSAM) is active and Transactional VSAM (TVS) is active on all processors and resubmit the command.

Recommendations:

- Issue the REPAIR.RECON command on a backup copy of the RECON data set to determine the impact that the command will have on the availability of the RECON data set. Then, issue the command when access to the actual RECON data set would be minimal.
- Issue the REPAIR.RECON DMBTABLE command only if a problem has been identified and, typically, at the direction of IBM Software Support.
- Issue the REPAIR.RECON DMBNUM CHECKUP command to assess if any DMB numbers in the RECON data set are inconsistent before issuing REPAIR.RECON DBNUM UPDATE command to correct the inconsistencies.
- Issue the REPAIR.RECON DMBNUM CHECKUP command after the RECON data sets have been upgraded to the latest version of IMS to verify that all of the DMB number representations for the various records in the RECON data set are valid. If inconsistencies are found, issue the REPAIR.RECON DMBNUM UPDATE to correct the inconsistencies.

  Schedule the REPAIR.RECON DMBNUM CHECKUP to run once a year.

Example of checking for inconsistencies in the DMB number representation

In this example, the data management block (DMB) number representation in the RECON header, database, and Fast Path area records are checked for inconsistencies for all databases.

```
//REPAIRRE JOB
//JOBLIB   DD DSN=xxxxx
//R1       EXEC PGM=DSPURX00
.
.
.
//SYSIN    DD   *
     REPAIR.RECON DMBNUM ALL CHECKUP
/*
```

Example of correcting inconsistencies in the DMB number representation

In this example, any inconsistencies in the data management block (DMB) number representation in the RECON header, database, and Fast Path area records are changed to be consistent.

```
//REPAIRRE JOB
//JOBLIB   DD DSN=xxxxx
//R1       EXEC PGM=DSPURX00
.
.
.
//SYSIN    DD   *
     REPAIR.RECON DMBNUM ALL UPDATE
/*
```

Chapter 18. Sample listings from the RECON data set

The sample listings from the RECON data set show the format and content of various records from an example RECON data set, as they are listed by LIST.RECON commands.

Sample listing of LIST.DB command output

Use the LIST.DB command to receive a list of databases that are registered in the RECON data set. You can produce a RECON listing by using the LIST.DB command from either an active or a remote site.

Sample listing of LIST.HISTORY command output

Use the LIST.HISTORY command to produce a history of activity for DBDSs or DEDB areas. You can produce a RECON listing by using the LIST.HISTORY command from either an active or a remote site.

Subsections:

- "Example of LIST.HISTORY command output"
- "Examples of the timeline summary output" on page 402

Example of LIST.HISTORY command output

The following is an example of LIST.HISTORY command output that includes a timeline.

```
IMS VERSION 15 RELEASE 1 DATA BASE RECOVERY CONTROL          PAGE 0001
  LIST.HISTORY DBD(DBOHIDK5)
17.132 10:05:17.427858              LISTING OF RECON                  PAGE 0002
--------------------------------------------------------------------------------
 DB
  DBD=DBOHIDK5                          DMB#=3      CHANGE#=5     TYPE=HALDB
  SHARE LEVEL=3               GSGNAME=**NULL**
  DBRCVGRP=**NULL**
  PSNAME=**NULL**  DBORG=PHIDAM   DSORG=OSAM    CURRENT PARTITION ID=00004
  FLAGS:                             COUNTERS:
    RECOVERABLE           =YES         PARTITIONS              =4
    ONLINE REORG CAPABLE  =YES         DATA SET GROUP MEMBERS  =1
                                       ALTER COUNT             =0
                                       ALTER COMPLETE COUNT    =0
--------------------------------------------------------------------------------
17.132 10:05:17.427858              LISTING OF RECON                  PAGE 0003
--------------------------------------------------------------------------------
 DB
  DBD=POHIDKA   MASTER DB=DBOHIDK5  IRLMID=*NULL  CHANGE#=2     TYPE=PART
  USID=0000000005  AUTHORIZED USID=0000000005  HARD USID=0000000005
  RECEIVE USID=0000000005  RECEIVE NEEDED USID=0000000000
  DSN PREFIX=IMSTESTS.DBOHIDK5                          PARTITION ID=00001
  PREVIOUS PARTITION=**NULL**    NEXT PARTITION=POHIDKB
  OLRIMSID=**NULL**    ACTIVE DBDS=A-J
  REORG#=00000
  ONLINE REORG STATISTICS:
      OLR BYTES MOVED = 0
      OLR SEGMENTS MOVED = 0
      OLR ROOT SEGMENTS MOVED = 0

  FREE SPACE:
```

© Copyright IBM Corp. 1974, 2017

```
   FREE BLOCK FREQ FACTOR=0    FREE SPACE PERCENTAGE=0

 PARTITION HIGH KEY/STRING (CHAR):              (LENGTH=5  )
   K0200
 PARTITION HIGH KEY/STRING (HEX):
   D2F0F2F0F0

 OSAM BLOCK SIZE:
   A = 4096

 FLAGS:                              COUNTERS:
   BACKOUT NEEDED        =OFF          RECOVERY NEEDED COUNT   =0
   READ ONLY             =OFF          IMAGE COPY NEEDED COUNT =0
   PROHIBIT AUTHORIZATION=OFF          AUTHORIZED SUBSYSTEMS   =0
                                       HELD AUTHORIZATION STATE=0
                                       EEQE COUNT              =0
   TRACKING SUSPENDED    =NO           RECEIVE REQUIRED COUNT  =0
   OFR REQUIRED          =NO           OLR ACTIVE HARD COUNT   =0
   PARTITION INIT NEEDED =NO           OLR INACTIVE HARD COUNT =0
   OLREORG CURSOR ACTIVE =NO
   PARTITION DISABLED    =NO
   ONLINE REORG CAPABLE  =YES
   REORG INTENT          =NO
   QUIESCE IN PROGRESS   =NO
   QUIESCE HELD          =NO
   ALTER IN PROGRESS     =NO
   PARTITION ALTERED     =NO
--------------------------------------------------------------------------------
17.132 10:05:17.427858             LISTING OF RECON                  PAGE 0004
--------------------------------------------------------------------------------
 DBDS
  DSN=IMSTESTS.DBOHIDK5.A00001                                    TYPE=PART
  DBD=POHIDKA   DDN=POHIDKAA DSID=001 DBORG=HIDAM  DSORG=OSAM
  CAGRP=CAGRP1    GENMAX=2     IC AVAIL=0     IC USED=2     DSSN=00000004
  NOREUSE         RECOVPD=0       OTHER DDN=**NULL**
  DEFLTJCL=**NULL**  ICJCL=PICJCL    OICJCL=POICJCL   RECOVJCL=PRECOJCL
  RECVJCL=PRECVJCL
  FLAGS:                              COUNTERS:
    IC NEEDED      =OFF
    RECOV NEEDED   =OFF
    RECEIVE NEEDED =OFF                 EEQE COUNT              =0
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
 IMAGE
  RUN     = 17.132 09:55:38.790739         *  RECORD COUNT =60
  STOP    = 00.000 00:00:00.000000            BATCH      USID=0000000002

 IC1
  DSN=IMSVS.POHIDKA.POHIDKAA.IC.IC095537           FILE SEQ=0001
  UNIT=SYSDA                          VOLS DEF=0001 VOLS USED=0001
                                      VOLSER=222222



 ALLOC
  ALLOC   =17.132 09:55:52.648653         *  ALLOC LRID =0000000000000000
  DSSN=0000000002 USID=0000000003 START = 17.132 09:55:50.813040


 ALLOC
  ALLOC   =17.132 09:59:28.695919         *  ALLOC LRID =0000000000000000
  DSSN=0000000003 USID=0000000004 START = 17.132 09:58:31.994836
  DEALLOC =17.132 09:59:49.372832          DEALLOC LRID =0000000000000000


 REORG
  RUN     = 17.132 10:01:14.301509         *    USID = 0000000004
```

```
 REORG#  = 00000

IMAGE
 RUN     = 17.132 10:01:18.279960         *  RECORD COUNT =15
 STOP    = 00.000 00:00:00.000000            BATCH      USID=0000000004

IC1
 DSN=IMSVS.POHIDKA.POHIDKAA.IC.IC100117           FILE SEQ=0001
 UNIT=SYSDA                          VOLS DEF=0001 VOLS USED=0001
                                     VOLSER=222222



 CA
17.132 10:05:17.427858              LISTING OF RECON                  PAGE 0005
  DSN=IMSVS.CAGRP1.CA.CA170201                    FILE SEQ=1
  CAGRP=CAGRP1       STOP    = 17.132 10:02:46.844810         *
                     UNIT=SYSDA      VOLS DEF=1    VOLS USED=1
                                     VOLSER=222222
  RUN     = 17.132 10:02:54.777684
   DBD=POHIDKA  DDN=POHIDKAA PURGETIME = 17.132 10:01:18.279960
     CHANGES ACCUMULATED=NO  COMPLETE CA=YES INDOUBT EEQES=NO
     LSN    = 000000000000      DSSN   = 0000000000
     LRID   = 00000000000000000  USID   = 0000000000

 ALLOC
  ALLOC   =17.132 10:03:13.617493         *  ALLOC LRID =0000000000000000
  DSSN=0000000004 USID=0000000005 START = 17.132 09:58:31.994836
  DEALLOC =17.132 10:03:31.814758          DEALLOC LRID =0000000000000000


 RECOV
  RUN     = 17.132 10:05:01.216151         *  RUN USID      = 0000000005

17.132 10:05:17.427858              LISTING OF RECON                  PAGE 0006
------------------------------------------------------------------------------
 DBDS
  DSN=IMSTESTS.DBOHIDK5.L00001                                TYPE=PART
  DBD=POHIDKA   DDN=POHIDKAL DSID=003 DBORG=INDEX  DSORG=VSAM
  FLAGS:                              COUNTERS:
    RECOV NEEDED   =OFF                 EEQE COUNT              =0
------------------------------------------------------------------------------
DSP0181I  NO CAGRP          RECORD FOUND
DSP0181I  NO CA             RECORD FOUND
DSP0181I  NO ALLOC          RECORD FOUND
DSP0181I  NO IMAGE          RECORD FOUND
DSP0181I  NO REORG          RECORD FOUND
DSP0181I  NO RECOV          RECORD FOUND
17.132 10:05:17.427858              LISTING OF RECON                  PAGE 0007
------------------------------------------------------------------------------
 DBDS
  DSN=IMSTESTS.DBOHIDK5.X00001                                TYPE=PART
  DBD=POHIDKA   DDN=POHIDKAX DSID=005 DBORG=INDEX  DSORG=VSAM
                                    OTHER DDN=**NULL**
  FLAGS:                              COUNTERS:
    RECOV NEEDED   =OFF                 EEQE COUNT              =0
------------------------------------------------------------------------------
DSP0181I  NO CAGRP          RECORD FOUND
DSP0181I  NO CA             RECORD FOUND
DSP0181I  NO ALLOC          RECORD FOUND
DSP0181I  NO IMAGE          RECORD FOUND
DSP0181I  NO REORG          RECORD FOUND
DSP0181I  NO RECOV          RECORD FOUND
17.132 10:05:17.427858              LISTING OF RECON                  PAGE 0008
------------------------------------------------------------------------------
 DB
```

```
 DBD=POHIDKB   MASTER DB=DBOHIDK5  IRLMID=*NULL  CHANGE#=3     TYPE=PART
 USID=0000000004  AUTHORIZED USID=0000000004  HARD USID=0000000004
 RECEIVE USID=0000000004  RECEIVE NEEDED USID=0000000000
 DSN PREFIX=IMSTESTS.DBOHIDK5                          PARTITION ID=00002
 PREVIOUS PARTITION=POHIDKA     NEXT PARTITION=POHIDKC
 OLRIMSID=**NULL**    ACTIVE DBDS=A-J
 REORG#=00000
 ONLINE REORG STATISTICS:
     OLR BYTES MOVED = 0
     OLR SEGMENTS MOVED = 0
     OLR ROOT SEGMENTS MOVED = 0

 FREE SPACE:
   FREE BLOCK FREQ FACTOR=0    FREE SPACE PERCENTAGE=0

 PARTITION HIGH KEY/STRING (CHAR):              (LENGTH=5  )
   K0400
 PARTITION HIGH KEY/STRING (HEX):
   D2F0F4F0F0

 OSAM BLOCK SIZE:
   A = 4096

 FLAGS:                              COUNTERS:
   BACKOUT NEEDED        =OFF          RECOVERY NEEDED COUNT   =0
   READ ONLY             =OFF          IMAGE COPY NEEDED COUNT =0
   PROHIBIT AUTHORIZATION=OFF          AUTHORIZED SUBSYSTEMS   =0
                                       HELD AUTHORIZATION STATE=0
                                       EEQE COUNT              =0
   TRACKING SUSPENDED    =NO           RECEIVE REQUIRED COUNT  =0
   OFR REQUIRED          =NO           OLR ACTIVE HARD COUNT   =0
   PARTITION INIT NEEDED =NO           OLR INACTIVE HARD COUNT =0
   OLREORG CURSOR ACTIVE =NO
   PARTITION DISABLED    =NO
   ONLINE REORG CAPABLE  =YES
   REORG INTENT          =NO
   QUIESCE IN PROGRESS   =NO
   QUIESCE HELD          =NO
   ALTER IN PROGRESS     =NO
   PARTITION ALTERED     =NO
--------------------------------------------------------------------------------
17.132 10:05:17.427858             LISTING OF RECON                  PAGE 0009
--------------------------------------------------------------------------------
 DBDS
  DSN=IMSTESTS.DBOHIDK5.A00002                                    TYPE=PART
  DBD=POHIDKB   DDN=POHIDKBA DSID=001 DBORG=HIDAM  DSORG=OSAM
  CAGRP=CAGRP1    GENMAX=2     IC AVAIL=0     IC USED=1     DSSN=00000003
  NOREUSE         RECOVPD=0       OTHER DDN=**NULL**
  DEFLTJCL=**NULL**  ICJCL=PICJCL    OICJCL=POICJCL   RECOVJCL=PRECOJCL
  RECVJCL=PRECVJCL
  FLAGS:                              COUNTERS:
    IC NEEDED      =OFF
    RECOV NEEDED   =OFF
    RECEIVE NEEDED =OFF                 EEQE COUNT              =0
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
 IMAGE
  RUN     = 17.132 09:55:39.976143         *  RECORD COUNT =2
  STOP    = 00.000 00:00:00.000000            BATCH      USID=0000000001

 IC1
  DSN=IMSVS.POHIDKB.POHIDKBA.IC.IC095537           FILE SEQ=0001
  UNIT=SYSDA                          VOLS DEF=0001 VOLS USED=0001
                                      VOLSER=222222
```

```
 ALLOC
  ALLOC   =17.132 09:55:53.114372         *  ALLOC LRID =0000000000000000
  DSSN=0000000001 USID=0000000002 START = 17.132 09:55:50.813040


 ALLOC
  ALLOC   =17.132 09:59:29.051080         *  ALLOC LRID =0000000000000000
  DSSN=0000000002 USID=0000000003 START = 17.132 09:58:31.994836
  DEALLOC =17.132 10:03:02.941604          DEALLOC LRID =0000000000000000


 CA
  DSN=IMSVS.CAGRP1.CA.CA170201                     FILE SEQ=1
  CAGRP=CAGRP1       STOP    = 17.132 10:02:46.844810         *
                     UNIT=SYSDA      VOLS DEF=1    VOLS USED=1
                                     VOLSER=222222
  RUN     = 17.132 10:02:54.777684
   DBD=POHIDKB  DDN=POHIDKBA PURGETIME = 17.132 09:55:39.976143
     CHANGES ACCUMULATED=YES COMPLETE CA=YES INDOUBT EEQES=NO
     LSN    = 000000000000      DSSN   = 0000000002
     LRID   = 0000000000001D46  USID   = 0000000003

 ALLOC
  ALLOC   =17.132 10:03:14.171247         *  ALLOC LRID =0000000000000000
  DSSN=0000000003 USID=0000000004 START = 17.132 09:58:31.994836
  DEALLOC =17.132 10:03:31.814758          DEALLOC LRID =0000000000000000


17.132 10:05:17.427858              LISTING OF RECON                  PAGE 0010
 RECOV
  RUN     = 17.132 10:05:03.815010         *  RUN USID      = 0000000004
DSP0181I  NO REORG          RECORD FOUND
17.132 10:05:17.427858              LISTING OF RECON                  PAGE 0011
-------------------------------------------------------------------------------
 DBDS
  DSN=IMSTESTS.DBOHIDK5.L00002                                  TYPE=PART
  DBD=POHIDKB   DDN=POHIDKBL DSID=003 DBORG=INDEX  DSORG=VSAM
  FLAGS:                              COUNTERS:
    RECOV NEEDED   =OFF                 EEQE COUNT              =0
-------------------------------------------------------------------------------
DSP0181I  NO CAGRP          RECORD FOUND
DSP0181I  NO CA             RECORD FOUND
DSP0181I  NO ALLOC          RECORD FOUND
DSP0181I  NO IMAGE          RECORD FOUND
DSP0181I  NO REORG          RECORD FOUND
DSP0181I  NO RECOV          RECORD FOUND
17.132 10:05:17.427858              LISTING OF RECON                  PAGE 0012
-------------------------------------------------------------------------------
 DBDS
  DSN=IMSTESTS.DBOHIDK5.X00002                                  TYPE=PART
  DBD=POHIDKB   DDN=POHIDKBX DSID=005 DBORG=INDEX  DSORG=VSAM
                                   OTHER DDN=**NULL**
  FLAGS:                              COUNTERS:
    RECOV NEEDED   =OFF                 EEQE COUNT              =0
-------------------------------------------------------------------------------
DSP0181I  NO CAGRP          RECORD FOUND
DSP0181I  NO CA             RECORD FOUND
DSP0181I  NO ALLOC          RECORD FOUND
DSP0181I  NO IMAGE          RECORD FOUND
DSP0181I  NO REORG          RECORD FOUND
DSP0181I  NO RECOV          RECORD FOUND
17.132 10:05:17.427858              LISTING OF RECON                  PAGE 0013
-------------------------------------------------------------------------------
 DB
  DBD=POHIDKC   MASTER DB=DBOHIDK5  IRLMID=*NULL  CHANGE#=4     TYPE=PART
  USID=0000000004  AUTHORIZED USID=0000000004  HARD USID=0000000004
```

```
 RECEIVE USID=0000000004  RECEIVE NEEDED USID=0000000000
 DSN PREFIX=IMSTESTS.DBOHIDK5                          PARTITION ID=00003
 PREVIOUS PARTITION=POHIDKB     NEXT PARTITION=POHIDKD
 OLRIMSID=**NULL**    ACTIVE DBDS=A-J
 REORG#=00000
 ONLINE REORG STATISTICS:
     OLR BYTES MOVED = 0
     OLR SEGMENTS MOVED = 0
     OLR ROOT SEGMENTS MOVED = 0

 FREE SPACE:
   FREE BLOCK FREQ FACTOR=0    FREE SPACE PERCENTAGE=0

 PARTITION HIGH KEY/STRING (CHAR):              (LENGTH=5  )
   K0600
 PARTITION HIGH KEY/STRING (HEX):
   D2F0F6F0F0

 OSAM BLOCK SIZE:
   A = 4096

 FLAGS:                              COUNTERS:
   BACKOUT NEEDED        =OFF          RECOVERY NEEDED COUNT   =0
   READ ONLY             =OFF          IMAGE COPY NEEDED COUNT =0
   PROHIBIT AUTHORIZATION=OFF          AUTHORIZED SUBSYSTEMS   =0
                                       HELD AUTHORIZATION STATE=0
                                       EEQE COUNT              =0
   TRACKING SUSPENDED    =NO           RECEIVE REQUIRED COUNT  =0
   OFR REQUIRED          =NO           OLR ACTIVE HARD COUNT   =0
   PARTITION INIT NEEDED =NO           OLR INACTIVE HARD COUNT =0
   OLREORG CURSOR ACTIVE =NO
   PARTITION DISABLED    =NO
   ONLINE REORG CAPABLE  =YES
   REORG INTENT          =NO
   QUIESCE IN PROGRESS   =NO
   QUIESCE HELD          =NO
   ALTER IN PROGRESS     =NO
   PARTITION ALTERED     =NO
--------------------------------------------------------------------------------
17.132 10:05:17.427858             LISTING OF RECON                  PAGE 0014
--------------------------------------------------------------------------------
 DBDS
  DSN=IMSTESTS.DBOHIDK5.A00003                                  TYPE=PART
  DBD=POHIDKC   DDN=POHIDKCA DSID=001 DBORG=HIDAM  DSORG=OSAM
  CAGRP=CAGRP1    GENMAX=2     IC AVAIL=0     IC USED=1     DSSN=00000003
  NOREUSE         RECOVPD=0       OTHER DDN=**NULL**
  DEFLTJCL=**NULL**  ICJCL=PICJCL    OICJCL=POICJCL   RECOVJCL=PRECOJCL
  RECVJCL=PRECVJCL
  FLAGS:                              COUNTERS:
    IC NEEDED      =OFF
    RECOV NEEDED   =OFF
    RECEIVE NEEDED =OFF                 EEQE COUNT              =0
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
 IMAGE
  RUN     = 17.132 09:55:41.157199         *  RECORD COUNT =2
  STOP    = 00.000 00:00:00.000000            BATCH      USID=0000000001

 IC1
  DSN=IMSVS.POHIDKC.POHIDKCA.IC.IC095537           FILE SEQ=0001
  UNIT=SYSDA                          VOLS DEF=0001 VOLS USED=0001
                                      VOLSER=222222



 ALLOC
  ALLOC   =17.132 09:55:53.381303         *  ALLOC LRID =0000000000000000
```

```
 DSSN=0000000001 USID=0000000002 START = 17.132 09:55:50.813040


ALLOC
 ALLOC   =17.132 09:59:29.342652         *  ALLOC LRID =0000000000000000
 DSSN=0000000002 USID=0000000003 START = 17.132 09:58:31.994836
 DEALLOC =17.132 10:03:03.040050          DEALLOC LRID =0000000000000000


CA
 DSN=IMSVS.CAGRP1.CA.CA170201                     FILE SEQ=1
 CAGRP=CAGRP1       STOP    = 17.132 10:02:46.844810         *
                    UNIT=SYSDA      VOLS DEF=1    VOLS USED=1
                                    VOLSER=222222
 RUN     = 17.132 10:02:54.777684
  DBD=POHIDKC  DDN=POHIDKCA PURGETIME = 17.132 09:55:41.157199
    CHANGES ACCUMULATED=YES COMPLETE CA=YES INDOUBT EEQES=NO
    LSN    = 000000000000      DSSN   = 0000000002
    LRID   = 0000000000001EB9  USID   = 0000000003

ALLOC
 ALLOC   =17.132 10:03:14.514883         *  ALLOC LRID =0000000000000000
 DSSN=0000000003 USID=0000000004 START = 17.132 09:58:31.994836
 DEALLOC =17.132 10:03:31.814758          DEALLOC LRID =0000000000000000
17.132 10:05:17.427858              LISTING OF RECON                  PAGE 0015
 RECOV
  RUN     = 17.132 10:05:06.945800         *  RUN USID      = 0000000004

DSP0181I  NO REORG          RECORD FOUND
17.132 10:05:17.427858              LISTING OF RECON                  PAGE 0016
-------------------------------------------------------------------------------
 DBDS
  DSN=IMSTESTS.DBOHIDK5.L00003                                  TYPE=PART
  DBD=POHIDKC   DDN=POHIDKCL DSID=003 DBORG=INDEX  DSORG=VSAM
  FLAGS:                             COUNTERS:
    RECOV NEEDED   =OFF                EEQE COUNT              =0
-------------------------------------------------------------------------------
DSP0181I  NO CAGRP          RECORD FOUND
DSP0181I  NO CA             RECORD FOUND
DSP0181I  NO ALLOC          RECORD FOUND
DSP0181I  NO IMAGE          RECORD FOUND
DSP0181I  NO REORG          RECORD FOUND
DSP0181I  NO RECOV          RECORD FOUND
17.132 10:05:17.427858              LISTING OF RECON                  PAGE 0017
-------------------------------------------------------------------------------
 DBDS
  DSN=IMSTESTS.DBOHIDK5.X00003                                  TYPE=PART
  DBD=POHIDKC   DDN=POHIDKCX DSID=005 DBORG=INDEX  DSORG=VSAM
                                  OTHER DDN=**NULL**
  FLAGS:                             COUNTERS:
    RECOV NEEDED   =OFF                EEQE COUNT              =0
-------------------------------------------------------------------------------
DSP0181I  NO CAGRP          RECORD FOUND
DSP0181I  NO CA             RECORD FOUND
DSP0181I  NO ALLOC          RECORD FOUND
DSP0181I  NO IMAGE          RECORD FOUND
DSP0181I  NO REORG          RECORD FOUND
DSP0181I  NO RECOV          RECORD FOUND
17.132 10:05:17.427858              LISTING OF RECON                  PAGE 0018
-------------------------------------------------------------------------------
 DB
  DBD=POHIDKD   MASTER DB=DBOHIDK5  IRLMID=*NULL  CHANGE#=5     TYPE=PART
  USID=0000000005  AUTHORIZED USID=0000000005  HARD USID=0000000005
  RECEIVE USID=0000000005  RECEIVE NEEDED USID=0000000000
  DSN PREFIX=IMSTESTS.DBOHIDK5                          PARTITION ID=00004
  PREVIOUS PARTITION=POHIDKC     NEXT PARTITION=**NULL**
```

```
 OLRIMSID=**NULL**    ACTIVE DBDS=A-J
 REORG#=00000
 ONLINE REORG STATISTICS:
     OLR BYTES MOVED = 0
     OLR SEGMENTS MOVED = 0
     OLR ROOT SEGMENTS MOVED = 0

 FREE SPACE:
   FREE BLOCK FREQ FACTOR=0    FREE SPACE PERCENTAGE=0

 PARTITION HIGH KEY/STRING (CHAR):              (LENGTH=5  )
   ŸŸŸŸŸ
 PARTITION HIGH KEY/STRING (HEX):
   FFFFFFFFFF

 OSAM BLOCK SIZE:
   A = 4096

 FLAGS:                              COUNTERS:
   BACKOUT NEEDED        =OFF          RECOVERY NEEDED COUNT   =0
   READ ONLY             =OFF          IMAGE COPY NEEDED COUNT =0
   PROHIBIT AUTHORIZATION=OFF          AUTHORIZED SUBSYSTEMS   =0
                                       HELD AUTHORIZATION STATE=0
                                       EEQE COUNT              =0
   TRACKING SUSPENDED    =NO           RECEIVE REQUIRED COUNT  =0
   OFR REQUIRED          =NO           OLR ACTIVE HARD COUNT   =0
   PARTITION INIT NEEDED =NO           OLR INACTIVE HARD COUNT =0
   OLREORG CURSOR ACTIVE =NO
   PARTITION DISABLED    =NO
   ONLINE REORG CAPABLE  =YES
   REORG INTENT          =NO
   QUIESCE IN PROGRESS   =NO
   QUIESCE HELD          =NO
   ALTER IN PROGRESS     =NO
   PARTITION ALTERED     =NO
--------------------------------------------------------------------------------
17.132 10:05:17.427858              LISTING OF RECON                  PAGE 0019
--------------------------------------------------------------------------------
 DBDS
  DSN=IMSTESTS.DBOHIDK5.A00004                                    TYPE=PART
  DBD=POHIDKD   DDN=POHIDKDA DSID=001 DBORG=HIDAM  DSORG=OSAM
  CAGRP=CAGRP1    GENMAX=2     IC AVAIL=0     IC USED=1     DSSN=00000004
  NOREUSE         RECOVPD=0       OTHER DDN=**NULL**
  DEFLTJCL=**NULL**  ICJCL=PICJCL    OICJCL=POICJCL   RECOVJCL=PRECOJCL
  RECVJCL=PRECVJCL
  FLAGS:                              COUNTERS:
    IC NEEDED      =OFF
    RECOV NEEDED   =OFF
    RECEIVE NEEDED =OFF                 EEQE COUNT              =0
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
 IMAGE
  RUN     = 17.132 09:55:42.200255         *  RECORD COUNT =60
  STOP    = 00.000 00:00:00.000000            BATCH      USID=0000000002

 IC1
  DSN=IMSVS.POHIDKD.POHIDKDA.IC.IC095537           FILE SEQ=0001
  UNIT=SYSDA                          VOLS DEF=0001 VOLS USED=0001
                                      VOLSER=222222



 ALLOC
  ALLOC   =17.132 09:55:51.931499         *  ALLOC LRID =0000000000000000
  DSSN=0000000002 USID=0000000003 START = 17.132 09:55:50.813040
```

```
 ALLOC
  ALLOC   =17.132 09:59:29.618104         *  ALLOC LRID =0000000000000000
  DSSN=0000000003 USID=0000000004 START = 17.132 09:58:31.994836
  DEALLOC =17.132 10:03:03.169020          DEALLOC LRID =0000000000000000


 CA
  DSN=IMSVS.CAGRP1.CA.CA170201                     FILE SEQ=1
  CAGRP=CAGRP1       STOP    = 17.132 10:02:46.844810         *
                     UNIT=SYSDA      VOLS DEF=1    VOLS USED=1
                                     VOLSER=222222
  RUN     = 17.132 10:02:54.777684
   DBD=POHIDKD  DDN=POHIDKDA PURGETIME = 17.132 09:55:42.200255
     CHANGES ACCUMULATED=YES COMPLETE CA=YES INDOUBT EEQES=NO
     LSN    = 000000000000      DSSN   = 0000000003
     LRID   = 0000000000001EC3  USID   = 0000000004

 ALLOC
  ALLOC   =17.132 10:03:14.849937         *  ALLOC LRID =0000000000000000
  DSSN=0000000004 USID=0000000005 START = 17.132 09:58:31.994836
  DEALLOC =17.132 10:03:31.814758          DEALLOC LRID =0000000000000000

17.132 10:05:17.427858              LISTING OF RECON                  PAGE 0020
 RECOV
  RUN     = 17.132 10:05:09.581948         *  RUN USID      = 0000000005

DSP0181I  NO REORG          RECORD FOUND
17.132 10:05:17.427858              LISTING OF RECON                  PAGE 0021
-------------------------------------------------------------------------------
 DBDS
  DSN=IMSTESTS.DBOHIDK5.L00004                                TYPE=PART
  DBD=POHIDKD   DDN=POHIDKDL DSID=003 DBORG=INDEX  DSORG=VSAM
  FLAGS:                             COUNTERS:
    RECOV NEEDED   =OFF                EEQE COUNT              =0
-------------------------------------------------------------------------------
DSP0181I  NO CAGRP          RECORD FOUND
DSP0181I  NO CA             RECORD FOUND
DSP0181I  NO ALLOC          RECORD FOUND
DSP0181I  NO IMAGE          RECORD FOUND
DSP0181I  NO REORG          RECORD FOUND
DSP0181I  NO RECOV          RECORD FOUND
17.132 10:05:17.427858              LISTING OF RECON                  PAGE 0022
-------------------------------------------------------------------------------
 DBDS
  DSN=IMSTESTS.DBOHIDK5.X00004                                TYPE=PART
  DBD=POHIDKD   DDN=POHIDKDX DSID=005 DBORG=INDEX  DSORG=VSAM
                                   OTHER DDN=**NULL**
  FLAGS:                             COUNTERS:
    RECOV NEEDED   =OFF                EEQE COUNT              =0
-------------------------------------------------------------------------------
DSP0181I  NO CAGRP          RECORD FOUND
DSP0181I  NO CA             RECORD FOUND
DSP0181I  NO ALLOC          RECORD FOUND
DSP0181I  NO IMAGE          RECORD FOUND
DSP0181I  NO REORG          RECORD FOUND
DSP0181I  NO RECOV          RECORD FOUND
17.132 10:05:17.427858              LISTING OF RECON                  PAGE 0023
-------------------------------------------------------------------------------
 PRILOG                                    RECORD SIZE=     304
  START = 17.132 09:55:50.813040         *    SSID=PWKBLP2  VERSION=15.1
  STOP  = 17.132 09:56:06.037616              #DSN=1
  GSGNAME=**NULL**                                 NORMAL TERM
  FIRST RECORD ID= 0000000000000001       PRILOG TOKEN= 0

  DSN=IMSTESTS.BATCH.LOG                                  UNIT=SYSDA
  START = 17.132 09:55:50.813040           FIRST DS LSN= 0000000000000001
```

```
  STOP  = 17.132 09:56:06.037616           LAST  DS LSN= 00000000000030E5
  FILE SEQ=0001    #VOLUMES=0001

    VOLSER=DSHR03 STOPTIME = 17.132 09:56:06.037616
      CKPTCT=0    CHKPT ID = 00.000 00:00:00.000000
      LOCK SEQUENCE#=  000000000000
--------------------------------------------------------------------------------
17.132 10:05:17.427858             LISTING OF RECON                  PAGE 0024
--------------------------------------------------------------------------------
 PRILOG                                          RECORD SIZE=     784
  START = 17.132 09:58:31.994836         *    SSID=IMS1     VERSION=15.1
  STOP  = 00.000 00:00:00.000000              #DSN=4
  GSGNAME=**NULL**
  FIRST RECORD ID= 0000000000000001       PRILOG TOKEN= 0
  EARLIEST CHECKPOINT = 17.132 09:58:32.554073

  DSN=IMSVS.RLDSP.IMS1.D17132.T0958319.V00                 UNIT=SYSDA
  START = 17.132 09:58:31.994836           FIRST DS LSN= 0000000000000001
  STOP  = 17.132 09:59:29.840457           LAST  DS LSN= 0000000000000D22
  FILE SEQ=0001    #VOLUMES=0001

    VOLSER=000000 STOPTIME = 17.132 09:59:29.840457
      CKPTCT=1    CHKPT ID = 17.132 09:58:32.554073
      LOCK SEQUENCE#=  000000000000

  DSN=IMSVS.RLDSP.IMS1.D17132.T0959298.V00                 UNIT=SYSDA
  START = 17.132 09:59:29.840457           FIRST DS LSN= 0000000000000D23
  STOP  = 17.132 09:59:49.448219           LAST  DS LSN= 0000000000001ED6
  FILE SEQ=0001    #VOLUMES=0001

    VOLSER=000000 STOPTIME = 17.132 09:59:49.448219
      CKPTCT=0    CHKPT ID = 17.132 09:58:32.554073
      LOCK SEQUENCE#=  000000000000

  DSN=IMSVS.RLDSP.IMS1.D17132.T0959494.V00                 UNIT=SYSDA
  START = 17.132 09:59:49.448219           FIRST DS LSN= 0000000000001ED7
  STOP  = 17.132 10:02:46.844810           LAST  DS LSN= 00000000000020E4
  FILE SEQ=0001    #VOLUMES=0001

    VOLSER=000000 STOPTIME = 17.132 10:02:46.844810
      CKPTCT=1    CHKPT ID = 17.132 09:59:49.448675
      LOCK SEQUENCE#=  000000000000

  DSN=IMSVS.RLDSP.IMS1.D17132.T1002468.V00                 UNIT=SYSDA
  START = 17.132 10:02:46.844810           FIRST DS LSN= 00000000000020E5
  STOP  = 17.132 10:03:31.907722           LAST  DS LSN= 000000000000291E
  FILE SEQ=0001    #VOLUMES=0001

    VOLSER=000000 STOPTIME = 17.132 10:03:31.907722
      CKPTCT=0    CHKPT ID = 17.132 09:59:49.448675
      LOCK SEQUENCE#=  000000000000
--------------------------------------------------------------------------------
```

Examples of the timeline summary output

The following are examples of the timeline summary output, which is included in LIST.HISTORY command output.

Example 1: Concurrent events and database quiesce deallocation event

This example shows the timeline summary output for an IMS database. The output includes allocation events and a deallocation event created by a database quiesce command:

```
17.132 10:05:17.427858              LISTING OF RECON                   PAGE 0025
+-------------------------------------------------------------------------------+
|Timeline for DBDS: POHIDKA  POHIDKAA                                           |
|                            USID=00000005   AUTHORIZED=00000005                |
|                          RECEIVE=00000000         HARD=00000005               |
|                     ACTIVE DBDS=A-J                                           |
+-Time-----------------+Events----+---+--+-----------------------------------+
|                      |IC        |   |  |                                   |
|                      |REORG     |   |US|Subsystem                          |
|                      |RECOV     |CA |ID|Logs and Allocs                    |
+----------------------+----------+---+--+-----------------------------------+
17.132 09:55:38.790739  B              2
17.132 09:55:50.813040                    PWKBLP2
17.132 09:55:52.648653                 3  A
17.132 09:56:06.037616                 3  C
17.132 09:58:31.994836                    IMS1
17.132 09:59:28.695919                 4  A
17.132 09:59:29.840457                 |  s
17.132 09:59:49.372832                 4  D
17.132 09:59:49.448219                    s
17.132 10:01:14.301509  G                 |
17.132 10:01:18.279960  B           <     4  |
17.132 10:02:46.844810              S        s
17.132 10:02:54.777684             Rnc       |
17.132 10:03:13.617493                  5  A
17.132 10:03:31.814758                  5  D
17.132 10:03:31.907722                     s
17.132 10:05:01.216151  R                  |
+----------------------+----------+---+--+-----------------------------------+
In the timeline, only the last digit of USID is shown.
           IC: B = Batch, U = UIC, O = OIC, C = CIC, s =  Stop
         REORG: G = REORG, GL = ONLINE REORG, GT = ALTER REORG
                s = Stoptime if online reorg
                y/n = Online reorg may be used as input
                      to recovery
                The last digit of the reorg# is shown if the reorg# is not zero
         RECOV: R = Run time, * = RECOVTO time (if any)
                The last digit of the reorg# is shown if the reorg# is not zero
            CA: R = Run time, S = Stop time, < = Purge time
                ? = Inconsistency between times
                y/n = CHANGES ACCUMULATED, i/c = (IN)/COMPLETE
          Logs: SSID = Open time, C = Log Close,
                v = Vol close, s = DS close
        Allocs: D = Dealloc time, A = Alloc time, DQ = Dealloc time with QUIESCE
+-------------------------------------------------------------------------------+
```

Explanation: Time stamp entry for 17.132 09:55:52.648653 records the allocation event. Time stamp entry for 17.132 09:59:49.372832 reflects a deallocation event.

Example 2: Reorganization number recorded in REORG event

This example shows the timeline summary output for a HALDB partition database. The output includes the reorganization number recorded for a HALDB online reorganization:

```
17.132 10:05:17.427858              LISTING OF RECON                   PAGE 0031
+-------------------------------------------------------------------------------+
|Timeline for DBDS: POHIDKC  POHIDKCA                                           |
|                            USID=00000004   AUTHORIZED=00000004                |
|                          RECEIVE=00000000         HARD=00000004               |
|                     ACTIVE DBDS=A-J                                           |
+-Time-----------------+Events----+---+--+-----------------------------------+
|                      |IC        |   |  |                                   |
|                      |REORG     |   |US|Subsystem                          |
|                      |RECOV     |CA |ID|Logs and Allocs                    |
+----------------------+----------+---+--+-----------------------------------+
```

```
17.132 09:55:41.157199  B          <   1
17.132 09:55:50.813040                 PWKBLP2
17.132 09:55:53.381303             |   2  A
17.132 09:56:06.037616             |   2  C
17.132 09:58:31.994836                 IMS1
17.132 09:59:29.342652             |   3  A
17.132 09:59:29.840457             |   |  s
17.132 09:59:49.448219             |   |  s
17.132 10:02:46.844810             S   |  s
17.132 10:02:54.777684            Ryc  |  |
17.132 10:03:03.040050                 3  D
17.132 10:03:14.514883                 4  A
17.132 10:03:31.814758                 4  D
17.132 10:03:31.907722                 s
17.132 10:05:06.945800  R              |
+---------------------+----------+---+--+-------------------------------------+
In the timeline, only the last digit of USID is shown.
            IC: B = Batch, U = UIC, O = OIC, C = CIC, s =  Stop
         RECOV: R = Run time, * = RECOVTO time (if any)
                The last digit of the reorg# is shown if the reorg# is not zero
            CA: R = Run time, S = Stop time, < = Purge time
                ? = Inconsistency between times
                y/n = CHANGES ACCUMULATED, i/c = (IN)/COMPLETE
          Logs: SSID = Open time, C = Log Close,
                v = Vol close, s = DS close
        Allocs: D = Dealloc time, A = Alloc time, DQ = Dealloc time with QUIESCE
+-------------------------------------------------------------------------------+
```

Explanation: The A-J DBDSs are the active data sets. POHIDKCA is the partner DBDS for this partition DBDS. The current reorganization number is set to 4.

Example 3: Reorganization number set by time-stamp recovery in RECOV event

This example shows the timeline summary outputs for a HALDB partition database. In the output, the reorganization number is set by time stamp recovery in the RECOV event:

```
17.132 10:05:17.427858              LISTING OF RECON                  PAGE 0034
+-------------------------------------------------------------------------------+
|Timeline for DBDS: POHIDKD  POHIDKDA                                           |
|                            USID=00000005   AUTHORIZED=00000005                |
|                         RECEIVE=00000000         HARD=00000005                |
|                   ACTIVE DBDS=A-J                                             |
+-Time-----------------+Events----+---+--+-------------------------------------+
|                      |IC        |   |  |                                     |
|                      |REORG     |   |US|Subsystem                            |
|                      |RECOV     |CA |ID|Logs and Allocs                      |
+----------------------+----------+---+--+-------------------------------------+
17.132 09:55:42.200255  B          <   2
17.132 09:55:50.813040                 PWKBLP2
17.132 09:55:51.931499             |   3  A
17.132 09:56:06.037616             |   3  C
17.132 09:58:31.994836                 IMS1
17.132 09:59:29.618104             |   4  A
17.132 09:59:29.840457             |   |  s
17.132 09:59:49.448219             |   |  s
17.132 10:02:46.844810             S   |  s
17.132 10:02:54.777684            Ryc  |  |
17.132 10:03:03.169020                 4  D
17.132 10:03:14.849937                 5  A
17.132 10:03:31.814758                 5  D
17.132 10:03:31.907722                 s
17.132 10:05:09.581948  R              |
+---------------------+----------+---+--+-------------------------------------+
In the timeline, only the last digit of USID is shown.
            IC: B = Batch, U = UIC, O = OIC, C = CIC, s =  Stop
```

```
         RECOV: R = Run time, * = RECOVTO time (if any)
                The last digit of the reorg# is shown if the reorg# is not zero
            CA: R = Run time, S = Stop time, < = Purge time
                ? = Inconsistency between times
                y/n = CHANGES ACCUMULATED, i/c = (IN)/COMPLETE
          Logs: SSID = Open time, C = Log Close,
                v = Vol close, s = DS close
        Allocs: D = Dealloc time, A = Alloc time, DQ = Dealloc time with QUIESCE
```

Explanation: The A-J DBDSs are the active data sets. POHIDKD is the partner DBDS for this partition DBDS. The current reorganization number is set to 5. This data set was the input data set used for the HALDB online reorganization at timeline entry 17.132 10:02:54.777684. The data set shows that the reorganization number was 3 before the HALDB online reorganization ran. The HALDB online reorganization incremented the reorganization number to 4 for the database. The HALDB online reorganization record in the output data set lists the reorganization number as 4. See the listing of the partner DBDS (POHIDKAM). The timeline entry for the time stamp recovery at 17.132 10:05:09.581948 recovered the data set before the last HALDB online reorganization. The reorganization number was set to 5.

The timeline summary for the partner DBDS shows the following:

```
17.132 10:05:18.098606              LISTING OF RECON                  PAGE 0009
+-------------------------------------------------------------------------------+
|Timeline for DBDS: POHIDKA  POHIDKAA                                           |
|                            USID=00000005   AUTHORIZED=00000005                |
|                         RECEIVE=00000000         HARD=00000005                |
|                    ACTIVE DBDS=A-J                                            |
+-Time-----------------+Events----+---+--+-----------------------------------+
|                      |IC        |   |  |                                   |
|                      |REORG     |   |US|Subsystem                          |
|                      |RECOV     |CA |ID|Logs and Allocs                    |
+----------------------+----------+---+--+-----------------------------------+
17.132 09:55:38.790739  B              2
17.132 09:55:50.813040                    PWKBLP2
17.132 09:55:52.648653                 3  A
17.132 09:56:06.037616                 3  C
17.132 09:58:31.994836                    IMS1
17.132 09:59:28.695919                 4  A
17.132 09:59:29.840457                 |  s
17.132 09:59:49.372832                 4  D
17.132 09:59:49.448219                    s
17.132 10:01:14.301509  G                 |
17.132 10:01:18.279960  B           <     4  |
17.132 10:02:46.844810              S        s
17.132 10:02:54.777684             Rnc       |
17.132 10:03:13.617493                 5  A
17.132 10:03:31.814758                 5  D
17.132 10:03:31.907722                    s
17.132 10:05:01.216151  R                 |
+----------------------+----------+---+--+-----------------------------------+
In the timeline, only the last digit of USID is shown.
             IC: B = Batch, U = UIC, O = OIC, C = CIC, s =  Stop
          REORG: G = REORG, GL = ONLINE REORG, GT = ALTER REORG
                 s = Stoptime if online reorg
                 y/n = Online reorg may be used as input
                       to recovery
                 The last digit of the reorg# is shown if the reorg# is not zero
          RECOV: R = Run time, * = RECOVTO time (if any)
                 The last digit of the reorg# is shown if the reorg# is not zero
             CA: R = Run time, S = Stop time, < = Purge time
                 ? = Inconsistency between times
                 y/n = CHANGES ACCUMULATED, i/c = (IN)/COMPLETE
```

```
          Logs: SSID = Open time, C = Log Close,
                v = Vol close, s = DS close
       Allocs: D = Dealloc time, A = Alloc time, DQ = Dealloc time with QUIESCE
```

Explanation: The A-J DBDSs are the active data sets. POHIDKAA is the partner DBDS for this partition DBDS. The current reorganization number is set to 5. This data set was the output data set used for the HALDB online reorganization at timeline entry 14.161 13:07:22:012345. It shows that the reorganization number was incremented to 4. The timeline summary for the partner DBDS shows that a time-stamp recovery occurred before the HALDB online reorganization and set the reorganization number to 5.

Sample listing of a RECON data set

Sample RECON record listings are shown to demonstrate a RECON data set.

Subsections:

- "RECON status record"
- "Log records"
- "SSYS record" on page 407
- "BACKOUT record" on page 407
- "CAGRP and CA records" on page 408
- "DBGRP, DBDSGRP, and RECOVGRP records" on page 409
- "DB (IMS) and related records" on page 414
- "DB (HALDB and PART) and related records" on page 416
- "DB (FP) and related records" on page 419

RECON status record

This sample shows the RECON status records.

```
17.131 10:01:48.307694              LISTING OF RECON                  PAGE 0003
-------------------------------------------------------------------------------
 DB
  DBD=DHVNTK03                          DMB#=1      CHANGE#=5     TYPE=HALDB
  SHARE LEVEL=0               GSGNAME=**NULL**
  DBRCVGRP=**NULL**
  PSNAME=**NULL**  DBORG=PHIDAM   DSORG=VSAM    CURRENT PARTITION ID=00004
  FLAGS:                            COUNTERS:
    RECOVERABLE           =YES        PARTITIONS              =4
    ONLINE REORG CAPABLE  =YES        DATA SET GROUP MEMBERS  =1
                                      ALTER COUNT             =0
                                      ALTER COMPLETE COUNT    =0
```

Log records

This sample shows the log records.

```
17.132 14:42:52.868734              LISTING OF RECON                  PAGE 0003
-------------------------------------------------------------------------------
 PRILOG                                    RECORD SIZE=     304
  START = 17.132 14:32:44.400778         *    SSID=LOADPDJK VERSION=15.1
  STOP  = 17.132 14:32:52.747727              #DSN=1
  GSGNAME=**NULL**                                  NORMAL TERM
  FIRST RECORD ID= 0000000000000001       PRILOG TOKEN= 0

  DSN=IMSTESTS.PDJKLOG1                                    UNIT=SYSDA
  START = 17.132 14:32:44.400778           FIRST DS LSN= 0000000000000001
  STOP  = 17.132 14:32:52.747727           LAST  DS LSN= 000000000000011C
```

```
  FILE SEQ=0001    #VOLUMES=0001

    VOLSER=DSHR03 STOPTIME = 17.132 14:32:52.747727
      CKPTCT=0    CHKPT ID = 00.000 00:00:00.000000
      LOCK SEQUENCE#=  000000000000

 LOGALL
  START   = 17.132 14:32:44.400778         *
  EARLIEST ALLOC TIME = 00.000 00:00:00.000000
  DBDS ALLOC=0
```

SSYS record

This sample shows the SSYS record.

```
17.135 10:58:20.921964              LISTING OF RECON                  PAGE 0009
-------------------------------------------------------------------------------
 SSYS
  SSID=IMS1      LOG START=17.135 10:51:56.572733
  SSTYPE=ONLINE  ABNORMAL TERM=OFF  RECOVERY STARTED=NO   BACKUP=NO
  TRACKED=NO     TRACKER TERM=OFF   SHARING COVERED DBS=NO
  IRLMID=IRLM2     IRLM STATUS=NORMAL        GSGNAME=**NULL**
  COEXISTENCE LEVEL=15.1

  AUTHORIZED DATA BASES/AREAS=3      VERSION=15.1  XRF CAPABLE=NO
                                                    ENCODED
    -DBD-        -AREA-   -LEVEL-  -ACCESS INTENT-  -STATE-
    DIVNTZ02     **NULL**    3        UPDATE           3
    DXVNTZ02     **NULL**    3        UPDATE           3
    DHVNTZ02     **NULL**    3        UPDATE           3
DSP0177I  NO BACKOUT   RECORDS DEFINED IN RECON
DSP0128I  NO CHANGE ACCUMULATION GROUPS DEFINED IN RECON
DSP0093I  NO DBDSGRP RECORD FOUND IN RECON
```

BACKOUT record

This sample shows the BACKOUT record.

```
17.132 12:11:49.689722              LISTING OF RECON                  PAGE 0002
-------------------------------------------------------------------------------
 BACKOUT
  SSID=SYS1      #UORS=2

    RECOVERY TOKEN=E2E8E2F140404040000000000400000000
    TIME=17.132 12:06:41.965920           PSB=PLAPJK02
                       BMP
      ASSOCIATED DATA BASES=3

                   BACKED   DYN BKOUT
      -DBD-        -OUT -   -FAILURE-
      DXHIDJ03      NO         YES
      DHVNTK03      NO         YES
      DXVNTK03      NO         YES

    RECOVERY TOKEN=E2E8E2F140404040000000000700000000
    TIME=17.132 12:09:04.631786           PSB=PSBEJK05
                INFLT   BMP
      ASSOCIATED DATA BASES=3

                   BACKED   DYN BKOUT
      -DBD-        -OUT -   -FAILURE-
      DBVHDJ05      NO         NO
      DBOHIDK5      NO         NO
      DXVHIDK5      NO         NO
```

```
DSP0180I  NUMBER OF RECORDS LISTED IS        1
DSP0203I  COMMAND COMPLETED WITH CONDITION CODE 00
DSP0220I  COMMAND COMPLETION TIME 17.132 12:11:50.039380
```

CAGRP and CA records

This sample shows the CAGRP and CA records.

```
17.132 10:37:44.401520              LISTING OF RECON                  PAGE 0010
-------------------------------------------------------------------------------
 CAGRP
  GRPNAME=CAGRP1   GRPMAX=10   CA AVAIL=2    CA USED=0
  REUSE   CAJCL=CAJCL          DEFLTJCL=**NULL**    RECOVPD=0
                               #MEMBERS=1     -DBD-     -DDN-
                                              HISAMISA  HISAMISA
-------------------------------------------------------------------------------

                            AVAILABLE DATA SET
                               CAGRP=CAGRP1

 CA
* DSN=IMSLI.CAGRP1.CA.CA001                        FILE SEQ=1
  CAGRP=CAGRP1       CREATE  = 17.132 10:37:28.977308         *
                     UNIT=3400       VOLS DEF=1    VOLS USED=0
                                     VOLSER=USER02

                            AVAILABLE DATA SET
                               CAGRP=CAGRP1

 CA
* DSN=IMSLI.CAGRP1.CA.CA002                        FILE SEQ=1
  CAGRP=CAGRP1       CREATE  = 17.132 10:37:28.989169         *
                     UNIT=3400       VOLS DEF=1    VOLS USED=0
                                     VOLSER=USER02
17.132 10:37:44.401520              LISTING OF RECON                  PAGE 0011
-------------------------------------------------------------------------------
 CAGRP
  GRPNAME=CAGRP2   GRPMAX=2    CA AVAIL=2    CA USED=0
  REUSE   CAJCL=CAJCL          DEFLTJCL=**NULL**    RECOVPD=0
                               #MEMBERS=1     -DBD-     -DDN-
                                              HDAMVSAM  DD1
-------------------------------------------------------------------------------

                            AVAILABLE DATA SET
                               CAGRP=CAGRP2

 CA
* DSN=IMSLI.CAGRP2.CA.CA001                        FILE SEQ=1
  CAGRP=CAGRP2       CREATE  = 17.132 10:37:29.001831         *
                     UNIT=3400       VOLS DEF=1    VOLS USED=0
                                     VOLSER=USER02

                            AVAILABLE DATA SET
                               CAGRP=CAGRP2

 CA
* DSN=IMSLI.CAGRP2.CA.CA002                        FILE SEQ=4
  CAGRP=CAGRP2       CREATE  = 17.132 10:37:29.012359         *
                     UNIT=3400       VOLS DEF=1    VOLS USED=0
                                     VOLSER=CAVO11
17.132 10:37:44.401520              LISTING OF RECON                  PAGE 0012
-------------------------------------------------------------------------------
 CAGRP
  GRPNAME=CAGRP3   GRPMAX=2    CA AVAIL=0    CA USED=0
  NOREUSE CAJCL=CAJCL          DEFLTJCL=**NULL**    RECOVPD=0
                               #MEMBERS=1     -DBD-     -DDN-
                                              HISAMVSA  HISAMVSA
```

```
--------------------------------------------------------------------------------
17.132 10:37:44.401520              LISTING OF RECON                  PAGE 0013
--------------------------------------------------------------------------------
 CAGRP
  GRPNAME=CAGRP4   GRPMAX=2    CA AVAIL=0    CA USED=0
  REUSE   CAJCL=CAJCL          DEFLTJCL=**NULL**    RECOVPD=0
                               #MEMBERS=1     -DBD-     -DDN-
                                              SHISAM    SHISAM
--------------------------------------------------------------------------------
17.132 10:37:44.401520              LISTING OF RECON                  PAGE 0014
--------------------------------------------------------------------------------
 CAGRP
  GRPNAME=CAGRP5   GRPMAX=2    CA AVAIL=0    CA USED=0
  REUSE   CAJCL=CAJCL          DEFLTJCL=**NULL**    RECOVPD=0
                               #MEMBERS=1     -DBD-     -DDN-
                                              SHISAMVS  SHISAMVS
--------------------------------------------------------------------------------
17.132 10:37:44.401520              LISTING OF RECON                  PAGE 0015
--------------------------------------------------------------------------------
 CAGRP
  GRPNAME=CAGRP6   GRPMAX=4    CA AVAIL=0    CA USED=0
  REUSE   CAJCL=CAJCL          DEFLTJCL=**NULL**    RECOVPD=0
                               #MEMBERS=1     -DBD-     -DDN-
                                              INDEXVSA  INDEXVSA
```

DBGRP, DBDSGRP, and RECOVGRP records

This sample shows the DBGRP, DBDSGRP, and RECOVGRP records.

```
17.132 14:42:52.868734              LISTING OF RECON                  PAGE 0016
--------------------------------------------------------------------------------
 DBGRP
  GRPNAME=MYGRP               #MEMBERS=300   -DBD/AREA-
                                              PDHDKAA
                                              PDHDKAB
                                              PDHDKAC
                                              PDHDKAD
                                              PDHDKAE
                                              PDHDKAF
                                              PDHDKAG
                                              PDHDKAH
                                              PDHDKAI
                                              PDHDKAJ
                                              PDHDKAK
                                              PDHDKAL
                                              PDHDKAM
                                              PDHDKAN
                                              PDHDKAO
                                              PDHDKAP
                                              PDHDKAQ
                                              PDHDKAR
                                              PDHDKAS
                                              PDHDKAT
                                              PDHDKAU
                                              PDHDKAV
                                              PDHDKAW
                                              PDHDKAX
                                              PDHDKAY
                                              PDHDKAZ
                                              PDHDKA0
                                              PDHDKA1
                                              PDHDKA2
                                              PDHDKA3
                                              PDHDKA4
                                              PDHDKA5
                                              PDHDKA6
                                              PDHDKA7
```

```
                                        PDHDKA8
                                        PDHDKA9
                                        PDHDKBA
                                        PDHDKBB
                                        PDHDKBC
                                        PDHDKBD
                                        PDHDKBE
                                        PDHDKBF
                                        PDHDKBG
                                        PDHDKBH
17.132 14:42:52.868734          LISTING OF RECON                PAGE 0017
                                        PDHDKBI
                                        PDHDKBJ
                                        PDHDKBK
                                        PDHDKBL
                                        PDHDKBM
                                        PDHDKBN
                                        PDHDKBO
                                        PDHDKBP
                                        PDHDKBQ
                                        PDHDKBR
                                        PDHDKBS
                                        PDHDKBT
                                        PDHDKBU
                                        PDHDKBV
                                        PDHDKBW
                                        PDHDKBX
                                        PDHDKBY
                                        PDHDKBZ
                                        PDHDKB0
                                        PDHDKB1
                                        PDHDKB2
                                        PDHDKB3
                                        PDHDKB4
                                        PDHDKB5
                                        PDHDKB6
                                        PDHDKB7
                                        PDHDKB8
                                        PDHDKB9
                                        PDHDKCA
                                        PDHDKCB
                                        PDHDKCC
                                        PDHDKCD
                                        PDHDKCE
                                        PDHDKCF
                                        PDHDKCG
                                        PDHDKCH
                                        PDHDKCI
                                        PDHDKCJ
                                        PDHDKCK
                                        PDHDKCL
                                        PDHDKCM
                                        PDHDKCN
                                        PDHDKCO
                                        PDHDKCP
                                        PDHDKCQ
17.132 14:42:52.868734          LISTING OF RECON                PAGE 0018
                                        PDHDKCR
                                        PDHDKCS
                                        PDHDKCT
                                        PDHDKCU
                                        PDHDKCV
                                        PDHDKCW
                                        PDHDKCX
                                        PDHDKCY
                                        PDHDKCZ
                                        PDHDKC0
```

```
                                                PDHDKC1
                                                PDHDKC2
                                                PDHDKC3
                                                PDHDKC4
                                                PDHDKC5
                                                PDHDKC6
                                                PDHDKC7
                                                PDHDKC8
                                                PDHDKC9
                                                PDHDKDA
                                                PDHDKDB
                                                PDHDKDC
                                                PDHDKDD
                                                PDHDKDE
                                                PDHDKDF
                                                PDHDKDG
                                                PDHDKDH
                                                PDHDKDI
                                                PDHDKDJ
                                                PDHDKDK
                                                PDHDKDL
                                                PDHDKDM
                                                PDHDKDN
                                                PDHDKDO
                                                PDHDKDP
                                                PDHDKDQ
                                                PDHDKDR
                                                PDHDKDS
                                                PDHDKDT
                                                PDHDKDU
                                                PDHDKDV
                                                PDHDKDW
                                                PDHDKDX
                                                PDHDKDY
                                                PDHDKDZ
17.132 14:42:52.868734          LISTING OF RECON                    PAGE 0019
                                                PDHDKD0
                                                PDHDKD1
                                                PDHDKD2
                                                PDHDKD3
                                                PDHDKD4
                                                PDHDKD5
                                                PDHDKD6
                                                PDHDKD7
                                                PDHDKD8
                                                PDHDKD9
                                                PDHDKEA
                                                PDHDKEB
                                                PDHDKEC
                                                PDHDKED
                                                PDHDKEE
                                                PDHDKEF
                                                PDHDOAA
                                                PDHDOAB
                                                PDHDOAC
                                                PDHDOAD
                                                PDHDOAE
                                                PDHDOAF
                                                PDHDOAG
                                                PDHDOAH
                                                PDHDOAI
                                                PDHDOAJ
                                                PDHDOAK
                                                PDHDOAL
                                                PDHDOAM
                                                PDHDOAN
                                                PDHDOAO
```

```
                                        PDHDOAP
                                        PDHDOAQ
                                        PDHDOAR
                                        PDHDOAS
                                        PDHDOAT
                                        PDHDOAU
                                        PDHDOAV
                                        PDHDOAW
                                        PDHDOAX
                                        PDHDOAY
                                        PDHDOAZ
                                        PDHDOA0
                                        PDHDOA1
                                        PDHDOA2
17.132 14:42:52.868734          LISTING OF RECON                PAGE 0020
                                        PDHDOA3
                                        PDHDOA4
                                        PDHDOA5
                                        PDHDOA6
                                        PDHDOA7
                                        PDHDOA8
                                        PDHDOA9
                                        PDHDOBA
                                        PDHDOBB
                                        PDHDOBC
                                        PDHDOBD
                                        PDHDOBE
                                        PDHDOBF
                                        PDHDOBG
                                        PDHDOBH
                                        PDHDOBI
                                        PDHDOBJ
                                        PDHDOBK
                                        PDHDOBL
                                        PDHDOBM
                                        PDHDOBN
                                        PDHDOBO
                                        PDHDOBP
                                        PDHDOBQ
                                        PDHDOBR
                                        PDHDOBS
                                        PDHDOBT
                                        PDHDOBU
                                        PDHDOBV
                                        PDHDOBW
                                        PDHDOBX
                                        PDHDOBY
                                        PDHDOBZ
                                        PDHDOB0
                                        PDHDOB1
                                        PDHDOB2
                                        PDHDOB3
                                        PDHDOB4
                                        PDHDOB5
                                        PDHDOB6
                                        PDHDOB7
                                        PDHDOB8
                                        PDHDOB9
                                        PDHDOCA
                                        PDHDOCB
17.132 14:42:52.868734          LISTING OF RECON                PAGE 0021
                                        PDHDOCC
                                        PDHDOCD
                                        PDHDOCE
                                        PDHDOCF
                                        PDHDOCG
                                        PDHDOCH
```

```
                                        PDHDOCI
                                        PDHDOCJ
                                        PDHDOCK
                                        PDHDOCL
                                        PDHDOCM
                                        PDHDOCN
                                        PDHDOCO
                                        PDHDOCP
                                        PDHDOCQ
                                        PDHDOCR
                                        PDHDOCS
                                        PDHDOCT
                                        PDHDOCU
                                        PDHDOCV
                                        PDHDOCW
                                        PDHDOCX
                                        PDHDOCY
                                        PDHDOCZ
                                        PDHDOC0
                                        PDHDOC1
                                        PDHDOC2
                                        PDHDOC3
                                        PDHDOC4
                                        PDHDOC5
                                        PDHDOC6
                                        PDHDOC7
                                        PDHDOC8
                                        PDHDOC9
                                        PDHDODA
                                        PDHDODB
                                        PDHDODC
                                        PDHDODD
                                        PDHDODE
                                        PDHDODF
                                        PDHDODG
                                        PDHDODH
                                        PDHDODI
                                        PDHDODJ
                                        PDHDODK
17.132 14:42:52.868734          LISTING OF RECON                PAGE 0022
                                        PDHDODL
                                        PDHDODM
                                        PDHDODN
                                        PDHDODO
                                        PDHDODP
                                        PDHDODQ
                                        PDHDODR
                                        PDHDODS
                                        PDHDODT
                                        PDHDODU
                                        PDHDODV
                                        PDHDODW
                                        PDHDODX
                                        PDHDODY
                                        PDHDODZ
                                        PDHDOD0
                                        PDHDOD1
                                        PDHDOD2
                                        PDHDOD3
                                        PDHDOD4
                                        PDHDOD5
                                        PDHDOD6
                                        PDHDOD7
                                        PDHDOD8
                                        PDHDOD9
                                        PDHDOEA
                                        PDHDOEB
```

```
                                          PDHDOEC
                                          PDHDOED
                                          PDHDOEE
                                          PDHDOEF
17.132 14:30:45.565622              LISTING OF RECON                  PAGE 0017
-------------------------------------------------------------------------------
 DBDSGRP
  GRPNAME=DBDSGRP1            #MEMBERS=1      -DBD-    -DDN/AREA-
                                               DIVNTZ02  DBHVSAM1

 DBDSGRP
  GRPNAME=DBDSGRP5            #MEMBERS=2      -DBD-    -DDN/AREA-
                                               DBOVLFPC  VLOSAM01
                                               DEDBJN23  DB23AR0
17.135 09:36:29.931706              LISTING OF RECON                  PAGE 0012
-------------------------------------------------------------------------------
 RECOVGRP
  GRPNAME=RECOVGP1            #MEMBERS=6      -DBD-     -AREA-
                                               DBVHDJ05
                                               DBOHIDK5
                                               DXVHIDK5
                                               D2XHIDK5
                                               D2XHDJ05
                                               DEDBDD01  DD01AR0

 RECOVGRP
  GRPNAME=RECOVGP2            #MEMBERS=3      -DBD-     -AREA-
                                               DHVNTZ02
                                               DXVNTZ02
                                               DIVNTZ02

 RECOVGRP
  GRPNAME=RECOVGP3            #MEMBERS=8      -DBD-     -AREA-
                                               DEDBJN23  DB23AR0
                                               DEDBJN23  DB23AR1
                                               DEDBJN23  DB23AR2
                                               DEDBJN23  DB23AR3
                                               DEDBJN23  DB23AR4
                                               DEDBJN23  DB23AR5
                                               DEDBJN23  DB23AR6
                                               DEDBJN23  DB23AR7
```

DB (IMS) and related records

This sample shows the DB (IMS) and related records.

```
17.132 10:37:44.401520              LISTING OF RECON                  PAGE 0016
-------------------------------------------------------------------------------
 DB
  DBD=HDAMVSAM                                      DMB#=6       TYPE=IMS
  SHARE LEVEL=0               GSGNAME=**NULL**      USID=0000000001
  AUTHORIZED USID=0000000000  RECEIVE USID=0000000000 HARD USID=0000000000
  RECEIVE NEEDED USID=0000000000
  DBRCVGRP=**NULL**
  FLAGS:                               COUNTERS:
    BACKOUT NEEDED        =OFF           RECOVERY NEEDED COUNT   =0
    READ ONLY             =OFF           IMAGE COPY NEEDED COUNT =1
    PROHIBIT AUTHORIZATION=OFF           AUTHORIZED SUBSYSTEMS   =0
    RECOVERABLE           =YES           HELD AUTHORIZATION STATE=0
                                         EEQE COUNT              =0
    TRACKING SUSPENDED    =NO            RECEIVE REQUIRED COUNT  =0
    OFR REQUIRED          =NO
    REORG INTENT          =NO
    QUIESCE IN PROGRESS   =NO
    QUIESCE HELD          =NO
-------------------------------------------------------------------------------
```

```
17.132 10:37:44.401520              LISTING OF RECON                  PAGE 0017
-------------------------------------------------------------------------------
 DBDS
  DSN=HDAMVSAM                                                      TYPE=IMS
  DBD=HDAMVSAM  DDN=DD1      DSID=001 DBORG=HDAM   DSORG=VSAM
  CAGRP=CAGRP2    GENMAX=2     IC AVAIL=0     IC USED=0     DSSN=00000000
  REUSE           RECOVPD=0
  DEFLTJCL=**NULL**  ICJCL=ICJCL     OICJCL=OICJCL    RECOVJCL=RECOVJCL
  RECVJCL=ICRCVJCL
  FLAGS:                             COUNTERS:
    IC NEEDED      =ON
    RECOV NEEDED   =OFF
    RECEIVE NEEDED =OFF                EEQE COUNT              =0
-------------------------------------------------------------------------------

 ALLOC
  ALLOC   =80.030 12:12:13.100000         *  ALLOC LRID =0000000000000000
  DSSN=0000000000 USID=0000000000 START = 80.030 12:12:12.000000
  DEALLOC =80.031 13:13:13.000000          DEALLOC LRID =0000000000000000
          CMD

 REORG
  RUN     = 80.030 12:12:12.100000         *    USID = 0000000000
  REORG#  = 00000
17.132 10:37:44.401520              LISTING OF RECON                  PAGE 0018
-------------------------------------------------------------------------------
 DB
  DBD=HIDAMVSA                                      DMB#=1       TYPE=IMS
  SHARE LEVEL=0               GSGNAME=**NULL**     USID=0000000001
  AUTHORIZED USID=0000000000  RECEIVE USID=0000000000 HARD USID=0000000000
  RECEIVE NEEDED USID=0000000000
  DBRCVGRP=**NULL**
  FLAGS:                             COUNTERS:
    BACKOUT NEEDED        =OFF         RECOVERY NEEDED COUNT   =0
    READ ONLY             =OFF         IMAGE COPY NEEDED COUNT =0
    PROHIBIT AUTHORIZATION=OFF         AUTHORIZED SUBSYSTEMS   =0
    RECOVERABLE           =YES         HELD AUTHORIZATION STATE=0
                                       EEQE COUNT              =0
    TRACKING SUSPENDED    =NO          RECEIVE REQUIRED COUNT  =0
    OFR REQUIRED          =NO
    REORG INTENT          =NO
    QUIESCE IN PROGRESS   =NO
    QUIESCE HELD          =NO
-------------------------------------------------------------------------------
17.132 10:37:44.401520              LISTING OF RECON                  PAGE 0019
-------------------------------------------------------------------------------
 DBDS
  DSN=HIDAMVSA                                                      TYPE=IMS
  DBD=HIDAMVSA  DDN=HIDAMVS1 DSID=001 DBORG=HIDAM  DSORG=VSAM
  CAGRP=**NULL**  GENMAX=2     IC AVAIL=1     IC USED=0     DSSN=00000000
  REUSE           RECOVPD=0
  DEFLTJCL=**NULL**  ICJCL=ICJCL     OICJCL=OICJCL    RECOVJCL=RECOVJCL
  RECVJCL=ICRCVJCL
  FLAGS:                             COUNTERS:
    IC NEEDED      =OFF
    IC RECOMMENDED =ON
    RECOV NEEDED   =OFF
    RECEIVE NEEDED =OFF                EEQE COUNT              =0
-------------------------------------------------------------------------------

                          AVAILABLE DATA SET
                       DBD=HIDAMVSA DDN=HIDAMVS1

 IMAGE
* CREATE  = 17.132 10:37:28.821180         *

 IC1
```

```
  DSN=DBRCIMS.HIDAMVSA.HIDAMVS1.IC.ICDSN1          FILE SEQ=0001
  UNIT=3400                          VOLS DEF=0001 VOLS USED=0000
                                     VOLSER=VOL003
17.132 10:37:44.401520              LISTING OF RECON                  PAGE 0020
-------------------------------------------------------------------------------
 DBDS
  DSN=HIDAMVSA                                                    TYPE=IMS
  DBD=HIDAMVSA  DDN=HIDAMVS2 DSID=002 DBORG=HIDAM  DSORG=VSAM
  CAGRP=**NULL**  GENMAX=2     IC AVAIL=1     IC USED=0     DSSN=00000000
  REUSE           RECOVPD=0
  DEFLTJCL=**NULL**  ICJCL=ICJCL     OICJCL=OICJCL    RECOVJCL=RECOVJCL
  RECVJCL=ICRCVJCL
  FLAGS:                             COUNTERS:
    IC NEEDED      =OFF
    IC RECOMMENDED =ON
    RECOV NEEDED   =OFF
    RECEIVE NEEDED =OFF                EEQE COUNT              =0
-------------------------------------------------------------------------------

                           AVAILABLE DATA SET
                        DBD=HIDAMVSA DDN=HIDAMVS2

 IMAGE
* CREATE  = 17.132 10:37:28.835174         *

 IC1
  DSN=DBRCIMS.HIDAMVSA.HIDAMVS2.IC.ICDSN2          FILE SEQ=0001
  UNIT=3400                          VOLS DEF=0001 VOLS USED=0000
                                     VOLSER=VOL004
```

DB (HALDB and PART) and related records

This sample shows the DB (HALDB and PART) and related records.

```
17.131 10:01:48.307694              LISTING OF RECON                  PAGE 0003
-------------------------------------------------------------------------------
 DB
  DBD=DHVNTK03                         DMB#=1      CHANGE#=5     TYPE=HALDB
  SHARE LEVEL=0               GSGNAME=**NULL**
  DBRCVGRP=**NULL**
  PSNAME=**NULL**  DBORG=PHIDAM   DSORG=VSAM    CURRENT PARTITION ID=00004
  FLAGS:                             COUNTERS:
    RECOVERABLE           =YES         PARTITIONS              =4
    ONLINE REORG CAPABLE  =YES         DATA SET GROUP MEMBERS  =1
                                       ALTER COUNT             =0
                                       ALTER COMPLETE COUNT    =0
-------------------------------------------------------------------------------
17.131 10:01:48.307694              LISTING OF RECON                  PAGE 0004
-------------------------------------------------------------------------------
 DB
  DBD=PHVNTKA   MASTER DB=DHVNTK03                CHANGE#=2     TYPE=PART
  USID=0000000001  AUTHORIZED USID=0000000000  HARD USID=0000000000
  RECEIVE USID=0000000000  RECEIVE NEEDED USID=0000000000
  DSN PREFIX=IMSTESTS.DHVNTK03                          PARTITION ID=00001
  PREVIOUS PARTITION=**NULL**    NEXT PARTITION=PHVNTKB
  OLRIMSID=**NULL**    ACTIVE DBDS=A-J
  REORG#=00000
  ONLINE REORG STATISTICS:
      OLR BYTES MOVED = 0
      OLR SEGMENTS MOVED = 0
      OLR ROOT SEGMENTS MOVED = 0

  FREE SPACE:
    FREE BLOCK FREQ FACTOR=0    FREE SPACE PERCENTAGE=0

  PARTITION HIGH KEY/STRING (CHAR):              (LENGTH=5  )
    K0200
```

```
  PARTITION HIGH KEY/STRING (HEX):
    D2F0F2F0F0

  FLAGS:                              COUNTERS:
    BACKOUT NEEDED        =OFF          RECOVERY NEEDED COUNT   =0
    READ ONLY             =OFF          IMAGE COPY NEEDED COUNT =0
    PROHIBIT AUTHORIZATION=OFF          AUTHORIZED SUBSYSTEMS   =0
                                        HELD AUTHORIZATION STATE=0
                                        EEQE COUNT              =0
    TRACKING SUSPENDED    =NO           RECEIVE REQUIRED COUNT  =0
    OFR REQUIRED          =NO           OLR ACTIVE HARD COUNT   =0
    PARTITION INIT NEEDED =YES          OLR INACTIVE HARD COUNT =0
    OLREORG CURSOR ACTIVE =NO
    PARTITION DISABLED    =NO
    ONLINE REORG CAPABLE  =YES
    REORG INTENT          =NO
    QUIESCE IN PROGRESS   =NO
    QUIESCE HELD          =NO
    ALTER IN PROGRESS     =NO
    PARTITION ALTERED     =NO
--------------------------------------------------------------------------------
17.131 10:01:48.307694              LISTING OF RECON                  PAGE 0005
--------------------------------------------------------------------------------
 DB
  DBD=PHVNTKB   MASTER DB=DHVNTK03                CHANGE#=3     TYPE=PART
  USID=0000000001  AUTHORIZED USID=0000000000  HARD USID=0000000000
  RECEIVE USID=0000000000  RECEIVE NEEDED USID=0000000000
  DSN PREFIX=IMSTESTS.DHVNTK03                          PARTITION ID=00002
  PREVIOUS PARTITION=PHVNTKA     NEXT PARTITION=PHVNTKC
  OLRIMSID=**NULL**    ACTIVE DBDS=A-J
  REORG#=00000
  ONLINE REORG STATISTICS:
      OLR BYTES MOVED = 0
      OLR SEGMENTS MOVED = 0
      OLR ROOT SEGMENTS MOVED = 0

  FREE SPACE:
    FREE BLOCK FREQ FACTOR=0    FREE SPACE PERCENTAGE=0

  PARTITION HIGH KEY/STRING (CHAR):               (LENGTH=5  )
    K0400
  PARTITION HIGH KEY/STRING (HEX):
    D2F0F4F0F0

  FLAGS:                              COUNTERS:
    BACKOUT NEEDED        =OFF          RECOVERY NEEDED COUNT   =0
    READ ONLY             =OFF          IMAGE COPY NEEDED COUNT =0
    PROHIBIT AUTHORIZATION=OFF          AUTHORIZED SUBSYSTEMS   =0
                                        HELD AUTHORIZATION STATE=0
                                        EEQE COUNT              =0
    TRACKING SUSPENDED    =NO           RECEIVE REQUIRED COUNT  =0
    OFR REQUIRED          =NO           OLR ACTIVE HARD COUNT   =0
    PARTITION INIT NEEDED =YES          OLR INACTIVE HARD COUNT =0
    OLREORG CURSOR ACTIVE =NO
    PARTITION DISABLED    =NO
    ONLINE REORG CAPABLE  =YES
    REORG INTENT          =NO
    QUIESCE IN PROGRESS   =NO
    QUIESCE HELD          =NO
    ALTER IN PROGRESS     =NO
    PARTITION ALTERED     =NO
--------------------------------------------------------------------------------
17.131 10:01:48.307694              LISTING OF RECON                  PAGE 0006
--------------------------------------------------------------------------------
 DB
  DBD=PHVNTKC   MASTER DB=DHVNTK03                CHANGE#=4     TYPE=PART
  USID=0000000001  AUTHORIZED USID=0000000000  HARD USID=0000000000
```

```
  RECEIVE USID=0000000000  RECEIVE NEEDED USID=0000000000
  DSN PREFIX=IMSTESTS.DHVNTK03                          PARTITION ID=00003
  PREVIOUS PARTITION=PHVNTKB     NEXT PARTITION=PHVNTKD
  OLRIMSID=**NULL**    ACTIVE DBDS=A-J
  REORG#=00000
  ONLINE REORG STATISTICS:
      OLR BYTES MOVED = 0
      OLR SEGMENTS MOVED = 0
      OLR ROOT SEGMENTS MOVED = 0

  FREE SPACE:
    FREE BLOCK FREQ FACTOR=0    FREE SPACE PERCENTAGE=0

  PARTITION HIGH KEY/STRING (CHAR):               (LENGTH=5  )
    K0600
  PARTITION HIGH KEY/STRING (HEX):
    D2F0F6F0F0

  FLAGS:                              COUNTERS:
    BACKOUT NEEDED        =OFF          RECOVERY NEEDED COUNT   =0
    READ ONLY             =OFF          IMAGE COPY NEEDED COUNT =0
    PROHIBIT AUTHORIZATION=OFF          AUTHORIZED SUBSYSTEMS   =0
                                        HELD AUTHORIZATION STATE=0
                                        EEQE COUNT              =0
    TRACKING SUSPENDED    =NO           RECEIVE REQUIRED COUNT  =0
    OFR REQUIRED          =NO           OLR ACTIVE HARD COUNT   =0
    PARTITION INIT NEEDED =YES          OLR INACTIVE HARD COUNT =0
    OLREORG CURSOR ACTIVE =NO
    PARTITION DISABLED    =NO
    ONLINE REORG CAPABLE  =YES
    REORG INTENT          =NO
    QUIESCE IN PROGRESS   =NO
    QUIESCE HELD          =NO
    ALTER IN PROGRESS     =NO
    PARTITION ALTERED     =NO
--------------------------------------------------------------------------------
17.131 10:01:48.307694              LISTING OF RECON                  PAGE 0007
--------------------------------------------------------------------------------
 DB
  DBD=PHVNTKD   MASTER DB=DHVNTK03                CHANGE#=5     TYPE=PART
  USID=0000000001  AUTHORIZED USID=0000000000  HARD USID=0000000000
  RECEIVE USID=0000000000  RECEIVE NEEDED USID=0000000000
  DSN PREFIX=IMSTESTS.DHVNTK03                          PARTITION ID=00004
  PREVIOUS PARTITION=PHVNTKC     NEXT PARTITION=**NULL**
  OLRIMSID=**NULL**    ACTIVE DBDS=A-J
  REORG#=00000
  ONLINE REORG STATISTICS:
      OLR BYTES MOVED = 0
      OLR SEGMENTS MOVED = 0
      OLR ROOT SEGMENTS MOVED = 0

  FREE SPACE:
    FREE BLOCK FREQ FACTOR=0    FREE SPACE PERCENTAGE=0

  PARTITION HIGH KEY/STRING (CHAR):               (LENGTH=5  )
    ŸŸŸŸŸ
  PARTITION HIGH KEY/STRING (HEX):
    FFFFFFFFFF

  FLAGS:                              COUNTERS:
    BACKOUT NEEDED        =OFF          RECOVERY NEEDED COUNT   =0
    READ ONLY             =OFF          IMAGE COPY NEEDED COUNT =0
    PROHIBIT AUTHORIZATION=OFF          AUTHORIZED SUBSYSTEMS   =0
                                        HELD AUTHORIZATION STATE=0
                                        EEQE COUNT              =0
    TRACKING SUSPENDED    =NO           RECEIVE REQUIRED COUNT  =0
    OFR REQUIRED          =NO           OLR ACTIVE HARD COUNT   =0
```

```
   PARTITION INIT NEEDED =YES         OLR INACTIVE HARD COUNT =0
   OLREORG CURSOR ACTIVE =NO
   PARTITION DISABLED    =NO
   ONLINE REORG CAPABLE  =YES
   REORG INTENT          =NO
   QUIESCE IN PROGRESS   =NO
   QUIESCE HELD          =NO
   ALTER IN PROGRESS     =NO
   PARTITION ALTERED     =NO
--------------------------------------------------------------------------------
DSP0180I  NUMBER OF RECORDS LISTED IS        5
DSP0203I  COMMAND COMPLETED WITH CONDITION CODE 00
DSP0220I  COMMAND COMPLETION TIME 17.131 10:01:48.615478
         IMS VERSION 15 RELEASE 1 DATA BASE RECOVERY CONTROL          PAGE 0008
DSP0211I  COMMAND PROCESSING COMPLETE
DSP0211I  HIGHEST CONDITION CODE = 00
```

DB (FP) and related records

This sample shows the DB (FP) and related records.

```
 17.132 10:46:50.860042              LISTING OF RECON                  PAGE 0024
--------------------------------------------------------------------------------
 DB
  DBD=DEDBJ002                                        DMB#=2       TYPE=FP
  SHARE LEVEL=3               RANDOMIZER=**NULL**
  FLAGS:                              COUNTERS:
                                        RECOVERY NEEDED COUNT   =0
                                        IMAGE COPY NEEDED COUNT =0
    PROHIBIT AUTHORIZATION=OFF          AUTHORIZED AREAS        =1
    RECOVERABLE           =YES          EEQE COUNT              =0
    FULLSEG DEFAULT       =NO           ALTER COUNT             =0
--------------------------------------------------------------------------------
17.132 10:46:50.860042             LISTING OF RECON                  PAGE 0025
--------------------------------------------------------------------------------
 DBDS
  DBD=DEDBJ002  AREA=DBJ2AR0                                        TYPE=FP
  SHARE LEVEL=3             DSID=00001 DBORG=DEDB   DSORG=VSAM
  GSGNAME=**NULL**             USID=0000000003
  AUTHORIZED USID=0000000003  RECEIVE USID=0000000003 HARD USID=0000000003
  RECEIVE NEEDED USID=0000000000
  CAGRP=**NULL**  GENMAX=2     IC AVAIL=0     IC USED=1     DSSN=00000002
  NOREUSE         RECOVPD=0      NOVSO  NOPREOPEN  NOPRELOAD  NOFULLSG
  CFSTR1=**NULL**         CFSTR2=**NULL**          NOLKASID  NOMAS
  DEFLTJCL=**NULL**  ICJCL=ICJCL    RECVJCL=ICRCVJCL  RECOVJCL=RECOVJCL
  DBRCVGRP=**NULL**
  FLAGS:                             COUNTERS:
    PROHIBIT AUTHORIZATION=OFF         AUTHORIZED SUBSYSTEMS   =3
                                       HELD AUTHORIZATION STATE=3
    IC NEEDED             =OFF         ADS AVAIL #             =1
    RECOV NEEDED          =OFF         REGISTERED ADS #        =1
                                       EEQE COUNT              =0
    RECEIVE NEEDED        =OFF         SHADOW ADS AVAIL #      =0
    OFR REQUIRED          =NO          REGISTERED SHADOW ADS # =1
    TRACKING SUSPENDED    =NO
    HSSP CIC IN PROGRESS  =NO
    QUIESCE IN PROGRESS   =NO
    QUIESCE HELD          =NO
    ALTER IN PROGRESS     =NO

  ADS LIST:
                                                                 CREATE
    -ADS DDN--ADS DSN-                                        -STAT- -RUNNING-
    DBJ2AR0S IMSTESTS.DEDBJ002.DBJ2AR0.SHADOW                 AVAIL     NO

  SHADOW ADS LIST:
                                                                 CREATE
```

```
   -ADS DDN--ADS DSN-                                        -STAT- -RUNNING-
   DBJ2AR0  IMSTESTS.DEDBJ002.DBJ2AR0                        UNAVAIL   NO

  ASSOCIATED SUBSYSTEM INFORMATION:
                              ENCODED
    -SSID-   -ACCESS INTENT-  -STATE-  -SS ROLE-
   IMS1         UPDATE           3      ACTIVE
   IMS3         UPDATE           3      ACTIVE
   IMS2         UPDATE           3      ACTIVE
--------------------------------------------------------------------------------

 ALLOC
  ALLOC   =17.132 10:46:44.535695         *  ALLOC LRID =0000000000000000
  DSSN=0000000002 USID=0000000003 START = 17.132 10:40:46.604713


 ALLOC
  ALLOC   =17.132 10:46:44.668111         *  ALLOC LRID =0000000000000000
  DSSN=0000000002 USID=0000000003 START = 17.132 10:42:40.019438
```

Fields in a RECON listing, by record type

These topics describe the fields that can be present in a listing of the RECON data set by record type.

Subsections:

- "RECON record fields"
- "Log record fields" on page 425
- "LOGALL record fields" on page 425
- "Online log record fields" on page 425
- "SSYS record fields" on page 425
- "BACKOUT record fields" on page 425
- "CAGRP record fields" on page 425
- "CA record fields" on page 425
- "Data group record fields" on page 425
- "DB (IMS) record fields" on page 426
- "DB (HALDB) record fields" on page 426
- "DB (PART) record fields" on page 426
- "DB (Fast Path) record fields" on page 426
- "DBDS (non-Fast Path) record fields" on page 426
- "DBDS (Fast Path) record fields" on page 426
- "ALLOC record fields" on page 426
- "IMAGE record fields" on page 426
- "REORG record fields" on page 426
- "RECOV record fields" on page 426

RECON record fields

A RECON record's fields and their corresponding line numbers are described in the table.

Table 21. Fields present in the RECON record

Record type	Line number	Field	Contents
RECON	1		
	2	RECOVERY CONTROL DATA SET, IMS VxRx	Identifies the version and release of the RECON data set.
	3	DMB# = *nnn*	Represents the value last assigned to a new database record. If the value equals 32,767, this value does not change and DMB numbers are being reused. Field LAST USED DMB# represents the last DMB number assigned.
		LAST USED DMB#= *nnn*	*nnn* represents the DMB number last assigned, if less than 32,767. If the DMB number has reached 32,767, *nnn* represents the last reused DMB. Only printed if greater than zero.
		INIT TOKEN= *token*	The token assigned to this RECON when it was initialized.
	4	FORCER | NOFORCER	Indicates whether databases must be registered in the RECON data set. FORCER indicates that all databases must be registered. NOFORCER indicates that all databases do not need to be registered.
		LOG DSN CHECK= *xxxxxx*	The type of log data set name checking; *xxxxxx* is CHECK17, CHECK44, or NOCHECK.
		STARTNEW= YES | NO	When I/O errors exist on one of the RECON data sets, YES indicates that new jobs are to start. NO indicates that no new jobs are to start.
	5	TAPE UNIT= *unittype*	The default unit type for log data sets, NOREUSE image copy data sets, and NOREUSE change accumulation data sets that reside on tape devices.
		DASD UNIT= *unittype*	The default unit type for log data sets, NOREUSE image copy data sets, and NOREUSE change accumulation data sets that reside on DASD devices.
		TRACEON | TRACEOFF	TRACEON indicates DBRC is to produce external GTF trace entries.
		SSID= *xxxxxxxx*	The IMS subsystem default name.
	6	LISTDLOG= YES | NO	Indicates whether the names of log data sets deleted from RECON are to be listed.
		CA | IC | LOG DATA SETS CATALOGED= YES | NO	Indicates whether CA, IC, and log data sets are to be treated as if they are cataloged. To bypass volume serial checking of utility DD statements, specify YES and the data set is cataloged. If you specify NO (or the data set is not cataloged), volume serial checking takes place.

Table 21. Fields present in the RECON record (continued)

Record type	Line number	Field	Contents
	7	MINIMUM VERSION = *x.x*	Indicates the minimum version of IMS that is allowed to sign on using these RECON data sets.
		CROSS DBRC SERVICE LEVEL ID = *nnnnn*	Indicates that the maintenance that corresponds to the given value has been applied to all systems that access the RECON data set.
	8	REORG NUMBER VERIFICATION = YES | NO	Indicates whether HALDB partition reorganization number verification is enabled.
	9	LOG RETENTION PERIOD= *yy.ddd hh:mm:ss.t*	Indicates the minimum amount of time that DBRC is to keep log records in the RECON data set.
	10	COMMAND AUTH=SAF | EXIT | BOTH | NONE	Indication of the level of DBRC command authorization being performed.
		HLQ=*hlq name*	High level qualifier of the resource name used to determine command authorization. **NULL** is printed when command authorization is not enabled.
	11	RCNQUAL=*data_set_name*	A substring of the RECON COPY1 data set name, which will be used in future access attempts on the RECON to determine whether the data set is a copy and whether command authorization should be enforced.
	12	TRACE OPTIONS= *options*	This line is printed only if an IBM representative has instructed you to gather documentation for problem analysis. It indicates that an IBM representative has provided instructions to enable DBRC trace records to be written to the DBRC trace table for problem analysis. The IBM representative will have provided the options.
	13	CATALOG=*catalog_name*	Name of the IMS catalog from which to retrieve the database definitions. **NULL** is printed when an IMS catalog is not used and the database definitions are retrieved from the DBDLIB data set.
	14	ACCESS=SERIAL | PARALLEL	The access mode for the RECON data set, either SERIAL access using RESERVE serialization or PARALLEL access using record-level locking.
		LIST=STATIC | CONCURR	The default option for LIST commands with parallel RECON access. It can be overridden by the STATIC | CONCURR keyword on the LIST.xxxxxxxx command. It has no meaning for serial RECON access (ACCESS=SERIAL).

Table 21. Fields present in the RECON record (continued)

Record type	Line number	Field	Contents
	15	SIZALERT= *xxxxxxxx xxxxxxxx xx*	The (optional) thresholds that trigger messages to warn you that a record has grown unnaturally large. Field values can be *dsnum*, *volnum* and *percent*. The *dsnum* and *volnum* values apply only to the PRILOG-family records. Given the 16 MB maximum allowable size of a PRILOG record, for practical purposes, *dsnum* and *volnum* are obsolete.
	16	LOGALERT= *xxxxxxxx xxxxxxxx*	The (optional) threshold that triggers the DSP0287W message (warning you when you just have time to shut down an online IMS subsystem before it ends abnormally because the PRILOG record exceeds RECORDSIZE). Field values can be dsnum and volnum. Given the 16 MB maximum allowable size of a PRILOG record, for practical purposes, LOGALERT is obsolete.
The following line is printed only if a failure occurred during a multiple update to the RECON:			
	17	UPDATE TYPE= *nnnn*	The type of multiple update that was in progress.
		DBID= *xxxxxxxx xxxxxxxx*	Data set name and data set ddname of the DBDS involved in the multiple update. Can be blank.
		CAGRP= *xxxxxxxx*	Name of the CA group involved in the multiple update. Contains blanks if no group was involved.
		NEW DDN= *xxxxxxxx*	New ddname, if any, that was associated with the DBDS that was involved in the multiple update. Contains blanks if no new name or DBDS was involved.
	18	OLD RECORD KEY	This field is printed only if the multiple update involves a key change.
	19	KEY TYPE= **xxxxx*	A description of the record type. The asterisk (*) is printed only if the record is available for future use (it is either new or scheduled for reuse). Whenever an unrecognizable record key type is found, BADTYP.KEY is printed, together with the key in hexadecimal characters.
		DBD= *xxxxxxxx*, DDN= *xxxxxxxx*	The database name and data set ddname of a DBDS. If the record is PRILOG, SECLOG, IPRI, or ISEC, the fields are printed as `**NULL**`. If the record is a CAGRP or DB record, the DBD field is `**NULL**`, and the DDN field contains the name of the CA group or database.
		TIME= *time_stamp*	The time stamp of the key of the record.

Table 21. Fields present in the RECON record (continued)

Record type	Line number	Field	Contents
	20-21	NEW RECORD KEY	This field is printed only if a new record key is being added. The additional fields that are printed for this part of the record are similar to the fields that are printed for OLD RECORD KEY.
	22-23	BASE RECORD KEY	This field is printed only if a new record key is being changed. The additional fields that are printed for this part of the record are similar to the fields that are printed for OLD RECORD KEY.
	24-25	SSID= *ssidname* IRLMID= *irlmidname* SHARE LVL #= *n* #DB= *n* B/O#= *n*	Information about the multiple update.
	26	FLAGS:	A heading for the following lines.
	27	RECOV= ON | OFF NORECOV= ON | OFF ICON= ON | OFF ICOFF= ON | OFF	Information about the multiple update.
	28	NORMAL= ON | OFF ABNORMAL= ON | OFF STARTRCV= ON | OFF ENDRECOV= ON | OFF	Information about the multiple update.
	29	READON= ON | OFF READOFF= ON | OFF AUTH= ON | OFF NOAUTH= ON | OFF	Information about the multiple update.
	30	B/O DONE= ON | OFF PASS 1= ON | OFF BACKOUT= ON | OFF SHARELVL= ON | OFF	Information about the multiple update.
	31	TIME STAMP INFORMATION:	Heading for section containing RECON time stamp information.
	32	TIMEZIN= *offset*	User-specified input time stamp offset default. If no offset is specified, this field displays `%SYS` meaning that the offset of the z/OS clock is to be used.
		-LABEL- -OFFSET-	These column headings are printed if a Time Zone Label Table has been defined.
	33-nnn	*label offset*	The list of defined time zone offset labels with their corresponding offset values.
	34	OUTPUT FORMAT:	Time-stamp output format settings.
		DEFAULT= *offset display form yearsize*	Default time-stamp output format settings.
	35	CURRENT= *offset display form yearsize*	The current time-stamp output format settings.
	36	IMSPLEX=*imsplex_name*, GROUP ID=*group_ID*	The IMSplex name and the DBRC group identifier associated with the RECON data set.
	37	-DDNAME- -STATUS- -DATA SET NAME-	Column headings for the following one to three lines.

Table 21. Fields present in the RECON record (continued)

Record type	Line number	Field	Contents
	38	RECON1 | RECON2 | RECON3 COPY1 *dsname*	The ddname, status, and data set name of the Copy1 RECON.
	39-40	RECON1 | RECON2 | RECON3 | COPY2 | DISCARDED | SPARE | UNAVAILABLE *dsname* or blank	These lines identify the ddname, status, and data set name of the RECON backup (Copy2), the RECON spare, and any RECON data sets that are not usable (DISCARDED). A RECON data set that could not be accessed is shown as UNAVAILABLE (unknown dsname and status).
	41	NUMBER OF REGISTERED DATABASES: *x*	The number of databases that are registered to DBRC.

Log record fields

A log record's fields, including types, and their corresponding line numbers are described in the table.

LOGALL record fields

A log allocation record's fields and their corresponding line numbers are described in the table.

Online log record fields

An online log record's fields and their corresponding line numbers are described in the table.

SSYS record fields

A subsystem record's fields and their corresponding line numbers are described in the table.

BACKOUT record fields

A backout record's fields and their corresponding line numbers are described in the table.

CAGRP record fields

A change accumulation group record's fields and their corresponding line numbers are described in the table.

CA record fields

A change accumulation record's fields and their corresponding line numbers are described in the table.

Data group record fields

A database data group record's fields and their corresponding line numbers are described in the table.

DB (IMS) record fields

An IMS database record's fields and their corresponding line numbers are described in the table.

DB (HALDB) record fields

The record fields and their corresponding line numbers for a HALDB master are described in the table.

DB (PART) record fields

The record fields and their corresponding line numbers for a HALDB partition database are described in the table.

DB (Fast Path) record fields

A Fast Path database record's fields and their corresponding line numbers are described in the table.

DBDS (non-Fast Path) record fields

A non-Fast Path database data set record's fields and their corresponding line numbers are described in the table.

DBDS (Fast Path) record fields

A Fast Path database data set record's fields and their corresponding line numbers are described in the table.

ALLOC record fields

An allocation record's fields and their corresponding line numbers are described in the table.

IMAGE record fields

An image record's fields and their corresponding line numbers are described in the table.

REORG record fields

A reorganization record's fields and their corresponding line numbers are described in the table.

RECOV record fields

A recovery record's fields and their corresponding line numbers are described in the table.

Part 5. Fast Database Recovery commands

Use the Fast Database Recovery (FDBR) commands to work with an IMS Fast Database Recovery region.

- Chapter 19, “MODIFY fdbrproc commands,” on page 429
- Chapter 20, “START fdbrproc command,” on page 433

© Copyright IBM Corp. 1974, 2017

Chapter 19. MODIFY fdbrproc commands

Use the MODIFY (F) *fdbrproc* commands for an IMS Fast Database Recovery region to display its status, to stop it, and to recover from failures.

F fdbrproc,DUMP

Use the F *fdbrproc*,DUMP command to stop IMS Fast DB Recovery functions and request a dump for the region. The region ends with a return code X'0020'.

Syntax

F fdbrproc,RECOVER

Use the F *fdbrproc*,RECOVER command to initiate an IMS Fast DB Recovery of tracked IMS databases. Use this command, for example, after a timeout during z/OS cross-system coupling facility (XCF) or log surveillance.

Syntax

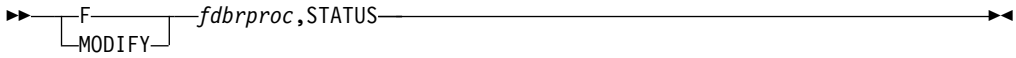
Usage notes

This command is not accepted if an IMS Fast DB Recovery region is currently starting or if it is recovering databases.

F fdbrproc,STATUS

Use the F *fdbrproc*,STATUS command to display status for an IMS Fast DB Recovery region.

Syntax



Output

The output displayed by this command includes the following information:

PHASE
This field indicates the phase of the Fast DB Recovery region. The following are possible values for the phase:
- INIT
- TRACKING

© Copyright IBM Corp. 1974, 2017

- RECOVERY

LOG-TIME
Time of the log record currently being processed by the Fast DB Recovery region. The length of time the Fast DB Recovery region lags behind IMS in reading the log is the difference between the current time, as shown by the time stamp, and the log time. This field displays `N/A` during FDBR initialization.

ACT-ID
The IMSID of the IMS subsystem that Fast DB Recovery is tracking. This field displays `N/A` during FDBR initialization.

GROUPNAME
z/OS cross-system coupling facility (XCF) group name used for XCF monitoring. This field displays `N/A` during FDBR initialization.

TIMEOUT
XCF timeout value. This field displays `N/A` during FDBR initialization.

AREA01
The sharing level for DEDB areas. The following are possible values for AREA01:
- RECOV
- NORECOV

You specify the sharing level for AREA01 in the DFSFDRxx member of IMS.PROCLIB.

SVSOOPEN
Indicates that all areas requiring redo processing in an FDBR system will be serially processed. This option is ignored for /ERE and XRF TKO processing. It is used to reduce the number of structures being allocated by FDBR for redo processing.

FPBUFF
Indicates that the control blocks for Fast Path DEDB processing are to be allocated in the FDBR private region rather than ECSA.

The LOCAL parameter is specified in the DFSFDRxx PROCLIB member. If this option is selected, the control blocks for DEDB processing are allocated in the FDBR control region private storage.

If this option is not specified, the control blocks for DEDB processing are allocated in ECSA storage. This is the default.

IRLMFAIL
Indicates the action that IMS will take if IRLM should fail.

ABEND
IMS ends abnormally with abend code `U3305`.

NOACT
IMS takes no action and continues normal processing.

You can specify this option in the DFSFDRxx member of the IMS PROCLIB data set. IRLMFAIL=NOACT is the default.

Example

```
F FDR1,STATUS
DFS000I PHASE: TRACKING  LOG-TIME: 17:48:41  FDR1
DFS000I     ACT-ID: SYSS3    GROUPNAME: FDRSYS3  FDR1
DFS000I TIMEOUT: 060 SEC AREA01: NORECOV     FDR1
DFS000I SVSOOPEN: SERIAL FPBUFF: LOCAL       FDR1
```

F fdbrproc,STOP

Use the F *fdbrproc,*STOP command to stop an IMS Fast DB Recovery region. The region ends with a return code X'0020'.

Syntax

F fdbrproc,TERM

Use the F *fdbrproc,*TERM command to stop IMS Fast DB Recovery tracking activity.

Syntax

Usage notes

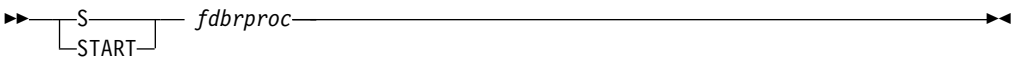
This command is not accepted if an IMS Fast DB Recovery region is currently starting or if it is recovering databases.

Chapter 20. START fdbrproc command

Although you normally start an IMS Fast DB Recovery region using a job submitted after the databases to be tracked are started, you can use the z/OS START command to start the region and tracking.

The fdbrproc parameter identifies the procedure name of the region to be started.

Syntax



© Copyright IBM Corp. 1974, 2017

Part 6. IMS Connect commands

These topics describe IMS Connect type-2 commands, IMS Connect WTOR commands, and IMS Connect z/OS (MVS) commands.

- Chapter 21, "Equivalent IMS Connect WTOR, z/OS, and type-2 commands," on page 437
- Chapter 22, "IMS Connect type-2 commands," on page 443
- Chapter 23, "IMS Connect WTOR commands," on page 445
- Chapter 24, "IMS Connect z/OS commands," on page 533

© Copyright IBM Corp. 1974, 2017

Chapter 21. Equivalent IMS Connect WTOR, z/OS, and type-2 commands

Certain IMS Connect WTOR commands, IMS Connect z/OS commands, and IMS Connect type-2 commands perform similar tasks.

The following table shows the different instances of the IMS Connect WTOR, z/OS, and type-2 commands that perform similar tasks.

Table 22. WTOR and IMS Connect z/OS equivalents for the QUERY IMSCON TYPE(ALIAS) command

QUERY IMSCON TYPE(ALIAS) command	Equivalent IMS Connect WTOR command	Equivalent IMS Connect z/OS command
QUERY IMSCON TYPE(ALIAS) NAME(*) SHOW(ALL | *show_parm*)	VIEWIA ALL	QUERY ALIAS NAME(*)
QUERY IMSCON TYPE(ALIAS) NAME(*alias_name*) SHOW(ALL | *show_parm*)	VIEWIA *alias_name*	QUERY ALIAS NAME(*aliasName*)
QUERY IMSCON TYPE(ALIAS) NAME(*alias_name*) ODBM(*odbm_name*)	VIEWIA *alias_name odbm_name*	QUERY ALIAS NAME(*aliasName*) ODBM(*odbmName*)

Table 23. WTOR and IMS Connect z/OS equivalents for the QUERY IMSCON TYPE(CONFIG) command

QUERY IMSCON TYPE(CONFIG) command	Equivalent IMS Connect WTOR command	Equivalent IMS Connect z/OS command
QUERY IMSCON TYPE(CONFIG) SHOW(ALL | *show_parm*)	VIEWHWS	QUERY MEMBER TYPE(IMSCON) SHOW(ALL)

Table 24. WTOR and IMS Connect z/OS equivalents for the QUERY IMSCON TYPE(DATASTORE) command

QUERY IMSCON TYPE(DATASTORE) command	Equivalent IMS Connect WTOR command	Equivalent IMS Connect z/OS command
QUERY IMSCON TYPE(DATASTORE) NAME(*) SHOW(ALL | *show_parm*)	VIEWDS ALL	QUERY DATASTORE NAME(*) SHOW(ALL)
QUERY IMSCON TYPE(DATASTORE) NAME(*datastore_name*) SHOW(ALL | *show_parm*)	VIEWDS *datastore_name*	QUERY DATASTORE NAME(*datastore_name*)

Table 25. WTOR and IMS Connect z/OS equivalents for the QUERY IMSCON TYPE(IMSPLEX) command

QUERY IMSCON TYPE(IMSPLEX) command	Equivalent IMS Connect WTOR command	Equivalent IMS Connect z/OS command
QUERY IMSCON TYPE(IMSPLEX) NAME(*) SHOW(ALL | *show_parm*)	VIEWIP ALL	QUERY IMSPLEX NAME(*) SHOW(ALL)
QUERY IMSCON TYPE(IMSPLEX) NAME(*IMSplex_name*) SHOW(ALL | *show_parm*)	VIEWIP *IMSplex_name*	QUERY IMSPLEX NAME(*imsplexName*) SHOW(ALL)

© Copyright IBM Corp. 1974, 2017

Table 26. WTOR and IMS Connect z/OS equivalents for the QUERY IMSCON TYPE(MSC) command

QUERY IMSCON TYPE(MSC) command	Equivalent IMS Connect WTOR command	Equivalent IMS Connect z/OS command
QUERY IMSCON TYPE(MSC) NAME(*) SHOW(ALL | *show_parm*)	VIEWMSC ALL	QUERY MSC NAME(*)
QUERY IMSCON TYPE(MSC) NAME(*msc_id*) SHOW(ALL | *show_parm*)	VIEWMSC *msc_id*	QUERY MSC NAME(*msc_id*)

Table 27. WTOR and IMS Connect z/OS equivalents for the QUERY IMSCON TYPE(PORT) command

QUERY IMSCON TYPE(PORT) command	Equivalent IMS Connect WTOR command	Equivalent IMS Connect z/OS command
QUERY IMSCON TYPE(PORT) NAME(*) SHOW(ALL | *show_parm*)	VIEWPORT ALL	QUERY PORT NAME(*) SHOW(ALL)
QUERY IMSCON TYPE(PORT) NAME(*portid*) SHOW(ALL | *show_parm*)	VIEWPORT *port_id*	QUERY PORT NAME(*port_id*) SHOW(ALL)
QUERY IMSCON TYPE(PORT) NAME(LOCAL) SHOW(ALL | *show_parm*)	VIEWPORT LOCAL	QUERY PORT NAME(LOCAL) SHOW(ALL)

Table 28. WTOR and IMS Connect z/OS equivalents for the QUERY IMSCON TYPE(RMTIMSCON) command

QUERY IMSCON TYPE(RMTIMSCON) command	Equivalent IMS Connect WTOR command	Equivalent IMS Connect z/OS command
QUERY IMSCON TYPE(RMTIMSCON) NAME(*) SHOW(ALL | *show_parm*)	VIEWRMT ALL	QUERY RMTIMSCON NAME(*)
QUERY IMSCON TYPE(RMTIMSCON) NAME(*rmtimscon_name*) SHOW(ALL | *show_parm*)	VIEWRMT *rmtimscon_name*	QUERY RMTIMSCON NAME(*rmtimscon_name*)

Table 29. WTOR and IMS Connect z/OS equivalents for the QUERY IMSCON TYPE(UOR) command

QUERY IMSCON TYPE(UOR) command	Equivalent IMS Connect WTOR command	Equivalent IMS Connect z/OS command
QUERY IMSCON TYPE(UOR) NAME(*) SHOW(ALL | *show_parm*)	VIEWUOR ALL	QUERY UOR NAME(*) SHOW(ALL)
QUERY IMSCON TYPE(UOR) NAME(*uor_id*) SHOW(ALL | *show_parm*)	VIEWUOR *uor_id*	QUERY UOR NAME(*uor_id*) SHOW(ALL)
QUERY IMSCON TYPE(UOR) STATE(*state*)	None	None

Table 30. WTOR and IMS Connect z/OS equivalents for the UPDATE IMSCON TYPE(ALIAS) command

UPDATE IMSCON TYPE(ALIAS) command	Equivalent IMS Connect WTOR command	Equivalent IMS Connect z/OS command
UDPATE IMSCON TYPE(ALIAS) NAME(*alias_name*) ODBM(*odbm_name*) START(COMM)	STARTIA *alias_name odbm_name*	UPDATE ALIAS NAME(*aliasName*) ODBM(*odbmName*) START(ROUTE)
UDPATE IMSCON TYPE(ALIAS) NAME(*alias_name*) ODBM(*odbm_name*) STOP(COMM)	STOPIA *alias_name odbm_name*	UPDATE ALIAS NAME(*aliasName*) ODBM(*odbmName*) STOP(ROUTE)

Table 31. WTOR and IMS Connect z/OS equivalents for the UPDATE IMSCON TYPE(CLIENT) command

UPDATE IMSCON TYPE(CLIENT) command	Equivalent IMS Connect WTOR command	Equivalent IMS Connect z/OS command
UPDATE IMSCON TYPE(CLIENT) NAME(*client_name*) PORT(*portid*) STOP(COMM)	STOPCLNT *portid clientid*	DELETE PORT NAME(*portName*) CLIENT(*clientName*)

Table 32. WTOR and IMS Connect z/OS equivalents for the UPDATE IMSCON TYPE(CONFIG) command

UPDATE IMSCON TYPE(CONFIG) command	Equivalent IMS Connect WTOR command	Equivalent IMS Connect z/OS command
UPDATE IMSCON TYPE(CONFIG) SHUTDOWN(COMM)	CLOSEHWS	SHUTDOWN MEMBER
UPDATE IMSCON TYPE(CONFIG) SHUTDOWN(COMM) OPTION(FORCE)	CLOSEHWS FORCE	SHUTDOWN MEMBER OPTION(FORCE)
UPDATE IMSCON TYPE(CONFIG) SHUTDOWN(COMM) OPTION(QUIESCE)	CLOSEHWS QUIESCE	SHUTDOWN MEMBER OPTION(QUIESCE)
UPDATE IMSCON TYPE(CONFIG) SET(OAUTO(ON))	SETOAUTO YES	UPDATE MEMBER TYPE(IMSCON) SET(OAUTO(ON))
UPDATE IMSCON TYPE(CONFIG) SET(OAUTO(OFF)	SETOAUTO NO	UPDATE MEMBER TYPE(IMSCON) SET(OAUTO(OFF))
UPDATE IMSCON TYPE(CONFIG) SET(PSWDMC(ON))	SETPWMC ON	UPDATE MEMBER TYPE(IMSCON) SET(PSWDMC(ON))
UPDATE IMSCON TYPE(CONFIG) SET(PSWDMC(OFF))	SETPWMC OFF	UPDATE MEMBER TYPE(IMSCON) SET(PSWDMC(OFF))
UPDATE IMSCON TYPE(CONFIG) SET(PSWDMC(RCF))	SETPWMC RCF	UPDATE MEMBER TYPE(IMSCON) SET(PSWDMC(RCF))
UPDATE IMSCON TYPE(CONFIG) SET(RACF(ON))	SETRACF ON	UPDATE MEMBER TYPE(IMSCON) SET(RACF(ON))

Table 32. WTOR and IMS Connect z/OS equivalents for the UPDATE IMSCON TYPE(CONFIG) command (continued)

UPDATE IMSCON TYPE(CONFIG) command	Equivalent IMS Connect WTOR command	Equivalent IMS Connect z/OS command
UPDATE IMSCON TYPE(CONFIG) SET(RACF(OFF))	SETRACF OFF	UPDATE MEMBER TYPE(IMSCON) SET(RACF(OFF))
UPDATE IMSCON TYPE(CONFIG) SET(RRS(ON))	SETRRS ON	UPDATE MEMBER TYPE(IMSCON) SET(RRS(ON))
UPDATE IMSCON TYPE(CONFIG) SET(RRS(OFF))	SETRRS OFF	UPDATE MEMBER TYPE(IMSCON) SET(RRS(OFF))
UPDATE IMSCON TYPE(CONFIG) SET(UIDCACHE(ON))	SETUIDC ON	UPDATE MEMBER TYPE(IMSCON) SET(UIDCACHE(ON))
UPDATE IMSCON TYPE(CONFIG) SET(UIDCACHE(OFF))	SETUIDC OFF	UPDATE MEMBER TYPE(IMSCON) SET(UIDCACHE(OFF))
UPDATE IMSCON TYPE(CONFIG) START(RECORDER)	RECORDER OPEN	UPDATE MEMBER TYPE(IMSCON) START(TRACE)
UPDATE IMSCON TYPE(CONFIG) STOP(RECORDER)	RECORDER CLOSE	UPDATE MEMBER TYPE(IMSCON) STOP(TRACE)

Table 33. WTOR and IMS Connect z/OS equivalents for the UPDATE IMSCON TYPE(CONVERTER) command

UPDATE IMSCON TYPE(CONVERTER) command	Equivalent IMS Connect WTOR command	Equivalent IMS Connect z/OS command
UPDATE IMSCON TYPE(CONVERTER) NAME(*converter_name*) OPTION(REFRESH)	REFRESH CONVERTER NAME(*converter_name*)	UPDATE CONVERTER NAME(*converter_name*) OPTION(REFRESH)

Table 34. WTOR and IMS Connect z/OS equivalents for the UPDATE IMSCON TYPE(DATASTORE) command.

UPDATE IMSCON TYPE(DATASTORE) command	Equivalent IMS Connect WTOR command	Equivalent IMS Connect z/OS command
UPDATE IMSCON TYPE(DATASTORE) NAME(*datastore_name*) START(COMM)	OPENDS *datastore_id* STARTDS *datastore_id*	UPDATE DATASTORE NAME(*datastoreName*) START(COMM)
UPDATE IMSCON TYPE(DATASTORE) NAME(*datastore_name*) STOP(COMM)	STOPDS *datastore_id*	UPDATE DATASTORE NAME(*datastoreName*) STOP(COMM)
UPDATE IMSCON TYPE(DATASTORE) NAME(*datastore_name*) SET()	None	None

Table 35. WTOR and IMS Connect z/OS equivalents for the UPDATE IMSCON TYPE(IMSPLEX) command.

UPDATE IMSCON TYPE(IMSPLEX) command	Equivalent IMS Connect WTOR command	Equivalent IMS Connect z/OS command
UPDATE IMSCON TYPE(IMSPLEX) NAME(*imsplex_name*) START(COMM)	OPENIP *imsplex_id* STARTIP *imsplex_id*	UPDATE IMSPLEX NAME(*imsplex_name*) START(COMM)
UPDATE IMSCON TYPE(IMSPLEX) NAME(*imsplex_name*) STOP(COMM)	STOPIP *imsplex_id*	UPDATE IMSPLEX NAME(*imsplex_name*) STOP(COMM)

Table 36. WTOR and IMS Connect z/OS equivalents for the UPDATE IMSCON TYPE(LINK) command

UPDATE IMSCON TYPE(LINK) command	Equivalent IMS Connect WTOR command	Equivalent IMS Connect z/OS command
UPDATE IMSCON TYPE(LINK) NAME(*logical_link_name*) STOP(COMM)	STOPLINK *logical_link_name*	DELETE LINK NAME(*linkName*)
UPDATE IMSCON TYPE(LINK) NAME(*logical_link_name*) MSC(*lclplk_id*) STOP(COMM)	STOPLINK *logical_link_name lclplk_id*	DELETE LINK NAME(*linkname*) LCLPLKID(*lclPlkid*)

Table 37. WTOR and IMS Connect z/OS equivalents for the UPDATE IMSCON TYPE(MSC) command

UPDATE IMSCON TYPE(MSC) command	Equivalent IMS Connect WTOR command	Equivalent IMS Connect z/OS command
UPDATE IMSCON TYPE(MSC) NAME(*lclPlkid*) START(COMM)	STARTMSC *lclPlkid*	UPDATE MSC NAME(*lclPlkid*) START(COMM)
UPDATE IMSCON TYPE(MSC) NAME(*lclPlkid*) STOP(COMM)	STOPMSC *lclPlkid*	UPDATE MSC NAME(*lclPlkid*) STOP(COMM)

Table 38. WTOR and IMS Connect z/OS equivalents for the UPDATE IMSCON TYPE(ODBM) command

UPDATE IMSCON TYPE(ODBM) command	Equivalent IMS Connect WTOR command	Equivalent IMS Connect z/OS command
UDPATE IMSCON TYPE(ODBM) NAME(*odbm_name*) START(COMM)	STARTOD *odbm_name*	UPDATE ODBM NAME(*odbmName*) START(COMM)
UDPATE IMSCON TYPE(ODBM) NAME(*odbm_name*) STOP(COMM)	STOPOD *odbm_name*	UPDATE ODBM NAME(*odbmName*) STOP(COMM)

Table 39. WTOR and IMS Connect z/OS equivalents for the UPDATE IMSCON TYPE(PORT) command.

UPDATE IMSCON TYPE(PORT) command	Equivalent IMS Connect WTOR command	Equivalent IMS Connect z/OS command
UDPATE IMSCON TYPE(PORT) NAME(*port_name*) START(COMM)	OPENPORT *port_id* STARTPT *port_id*	UPDATE PORT NAME(*port_name*) START(COMM)

Table 39. WTOR and IMS Connect z/OS equivalents for the UPDATE IMSCON TYPE(PORT) command (continued).

UPDATE IMSCON TYPE(PORT) command	Equivalent IMS Connect WTOR command	Equivalent IMS Connect z/OS command
UDPATE IMSCON TYPE(PORT) NAME(*port_name*) STOP(COMM)	STOPPORT *port_id*	UPDATE PORT NAME(*port_name*) STOP(COMM)

Table 40. WTOR and IMS Connect z/OS equivalents for the UPDATE IMSCON TYPE(RACFUID) command

UPDATE IMSCON TYPE(RACFUID) command	Equivalent IMS Connect WTOR command	Equivalent IMS Connect z/OS command
UPDATE IMSCON TYPE(RACFUID) NAME(*userid*) OPTION(REFRESH)	REFRESH RACFUID NAME(*userid*)	UPDATE RACFUID NAME(*userid*) OPTION(REFRESH)

Table 41. WTOR and IMS Connect z/OS equivalents for the UPDATE IMSCON TYPE(RMTIMSCON) command

UPDATE IMSCON TYPE(RMTIMSCON) command	Equivalent IMS Connect WTOR command	Equivalent IMS Connect z/OS command
UPDATE IMSCON TYPE(RMTIMSCON) NAME(*rmtimscon_name*) START(COMM)	STARTRMT *rmtimsconName*	UPDATE RMTIMSCON NAME(*rmtimsconName*) START(COMM)
UPDATE IMSCON TYPE(RMTIMSCON) NAME(*rmtimscon_name*) STOP(COMM)	STOPRMT *rmtimsconName*	UPDATE RMTIMSCON NAME(*rmtimsconName*) STOP(COMM)

Table 42. WTOR and MS Connect z/OS equivalents for the UPDATE IMSCON TYPE(SENDCLNT) command

UPDATE IMSCON TYPE(SENDCLNT) command	Equivalent IMS Connect WTOR command	Equivalent IMS Connect z/OS command
UPDATE IMSCON TYPE(SENDCLNT) NAME(*sendclient_name*) RMTIMSCON(*rmtimscon_name*) STOP(COMM)	STOPSCLN *rmtimscon sendclient*	DELETE RMTIMSCON NAME(*rmtimsconname*) SENDCLNT(*clientid*)
UPDATE IMSCON TYPE(SENDCLNT) NAME(*sendclient_name*) RMTCICS(*rmtcics_name*) STOP(COMM)	STOPSCLN *rmtcics sendclient*	No equivalent z/OS command

Chapter 22. IMS Connect type-2 commands

IMS Connect type-2 commands are provided to simplify the management of IMS Connect resources in an IMSplex environment. Like all type-2 commands, IMS Connect type-2 commands are supported only through the Operations Manager (OM) API, such as TSO SPOC, REXX SPOC API, and Batch SPOC.

IMS Connect type-2 commands are described with all of the other IMS type-2 commands.

Related reference:

- CREATE IMSCON commands (Commands)
- DELETE IMSCON commands (Commands)
- QUERY IMSCON commands (Commands)
- UPDATE IMSCON commands (Commands)

© Copyright IBM Corp. 1974, 2017

Chapter 23. IMS Connect WTOR commands

All IMS Connect WTOR commands must be immediately preceded on the command line of the z/OS system console by the reply number of the outstanding IMS Connect reply message (for example, *nn*HWSCMD where *nn* is the reply number).

Related reference:

Equivalent IMS Connect WTOR, z/OS, and type-2 commands (Commands)

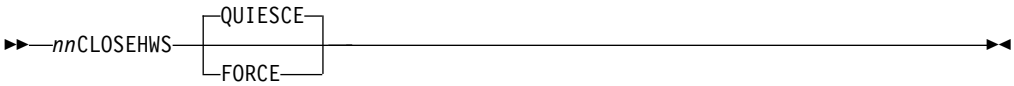
CLOSEHWS command

Use the CLOSEHWS command to terminate IMS Connect.

Subsections:

- "Syntax"
- "Keywords"
- "Usage notes" on page 446
- "Equivalent IMS Connect type-2 and z/OS commands" on page 446
- "Examples" on page 447

Syntax



Keywords

The following keywords are valid for the CLOSEHWS command:

nn 2-digit number of z/OS system console outstanding display.

FORCE
Specifies that termination is to end all client and data store connections immediately, which forces any IMS applications that are executing for the connected clients to abnormally terminate.

QUIESCE
Specifies that termination is to end all client and data store connections in a controlled manner. If no parameter is specified for the CLOSEHWS command, this parameter is used by default.

All work that is currently in progress, or that is queued for processing, is completed before IMS Connect is terminated. No new work is accepted after this command has been entered and accepted.

IMS Connect shuts down in the following order:

1. All active units of work for clients/browsers are completed.
2. Communication between IMS Connect and IMS is terminated.
3. IMS Connect terminates.

© Copyright IBM Corp. 1974, 2017

Usage notes

Use the CLOSEHWS command to terminate IMS Connect.

For IMS-to-IMS TCP/IP connections, CLOSEHWS closes all send socket connections that are in CONN state. For any send socket connections that are in RECV state IMS Connect waits indefinitely for an ACK or NAK. IMS Connect terminates only after an ACK or NAK is received for all connections that are in a RECV state.

Using the FORCE parameter terminates client and data store activity immediately. Using the QUIESCE parameter enables client and IMS host applications to run to completion. You can issue a CLOSEHWS FORCE command after issuing a CLOSEHWS QUIESCE command.

For IMS-to-IMS TCP/IP connections, the CLOSEHWS FORCE closes all send socket connections. For any send socket connections that are in RECV state, IMS Connect sends a NAK to OTMA that directs OTMA to reroute the message to the OTMA timeout queue, DFS$$TOQ.

Equivalent IMS Connect type-2 and z/OS commands

The following table lists IMS Connect type-2 commands and IMS Connect z/OS commands that perform similar functions as the CLOSEHWS command.

CLOSEHWS command	Equivalent IMS Connect type-2 command	Equivalent IMS Connect z/OS command
CLOSEHWS	UPDATE IMSCON TYPE(CONFIG) SHUTDOWN(COMM)	SHUTDOWN MEMBER
CLOSEHWS FORCE	UPDATE IMSCON TYPE(CONFIG) SHUTDOWN(COMM) OPTION(FORCE)	SHUTDOWN MEMBER OPTION(FORCE)
CLOSEHWS QUIESCE	UPDATE IMSCON TYPE(CONFIG) SHUTDOWN(COMM) OPTION(QUIESCE)	SHUTDOWN MEMBER OPTION(QUIESCE)

You can also use the z/OS MVS system commands STOP or CANCEL to terminate IMS Connect.

If you need to use a z/OS MVS system command, use the STOP command instead of CANCEL command, because the CANCEL command can leave IMS Connect functions, such as RRS, in unknown states. The results of the STOP command are the same as the IMS Connect CLOSEHWS QUIESCE command.

The following examples show the STOP MVS system command:

```
STOP ims_connect_jobname,dump
```

or

```
STOP ims_connect_jobname
```

The following examples show the CANCEL MVS system command:

```
CANCEL ims_connect_jobname,dump
```

or

```
CANCEL ims_connect_jobname
```

Examples

To close IMS Connect, you can use either of the following commands:

- *nn*CLOSEHWS QUIESCE
- *nn*CLOSEHWS FORCE

After issuing *nn*CLOSEHWS QUIESCE, if IMS Connect does not terminate because work in progress does not complete quickly enough, you can subsequently issue the *nn*CLOSEHWS FORCE command.

Where *nn* is the 2-digit number of z/OS system console outstanding display.

Related reference:

UPDATE IMSCON TYPE(CONFIG) command (Commands)

"IMS Connect SHUTDOWN MEMBER command" on page 587

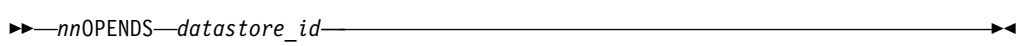
OPENDS command

Use the OPENDS command to start communication between the IMS Connect and a data store.

Subsections:

- "Syntax"
- "Keywords"
- "Usage notes"
- "Equivalent IMS Connect type-2 and z/OS commands" on page 448
- "Examples" on page 448

Syntax



Keywords

The following keywords are valid for the OPENDS command:

nn 2-digit number of z/OS system console outstanding display.

datastore_id
Specifies the name of the data store. This name must be defined to the IMS Connect through the configuration member HWSCFGxx, and must match one of the IDs that is defined in the DATASTORE configuration statement or statements.

Usage notes

Use this command to reestablish communication with a data store after communication fails between the IMS Connect and the data store. For example, use this command to restart communication when all activity for the data store in the IMS Connect is terminated, or after a STOPDS command has terminated communication with the data store.

Use the VIEWDS command to display information about data stores if you are not sure about the activity of a particular data store.

The OPENDS command does not affect a data store that is already active or a data store that is not defined to the IMS Connect in the configuration member HWSCFGxx.

Equivalent IMS Connect type-2 and z/OS commands

The following table lists IMS Connect type-2 commands and IMS Connect z/OS commands that perform similar functions as the OPENDS command.

OPENDS command	Equivalent IMS Connect type-2 command	Equivalent IMS Connect z/OS command
OPENDS *datastore_id*	UPDATE IMSCON TYPE(DATASTORE) NAME(*datastore_name*) START(COMM)	UPDATE DATASTORE NAME(*datastoreName*) START(COMM)

Examples

To open communication to data store IMSA:

```
nnOPENDS IMSA
```

Where *nn* is the 2-digit number of z/OS system console outstanding display.

Related reference:

UPDATE IMSCON TYPE(DATASTORE) command (Commands)

"IMS Connect UPDATE DATASTORE command" on page 591

"STARTDS command" on page 460

OPENIP command

Use the OPENIP command to start communication between the IMS Connect and the IMSplex that contains OM which is connected to SCI.

Subsections:

- "Syntax"
- "Keywords"
- "Usage notes" on page 449
- "Equivalent IMS Connect type-2 and z/OS commands" on page 449
- "Examples" on page 449

Syntax

Keywords

The following keywords are valid for the OPENIP command:

nn 2-digit number of z/OS system console outstanding display.

imsplex_id
Specifies the name of the IMSplex. This name must be defined to the IMS Connect through the configuration member HWSCFGxx, and must match the TMEMBER that is defined in the IMSplex configuration statement.

Usage notes

Use this command to reestablish communication with the IMSplex that is being used to communicate with OM if communication has failed between IMS Connect and the IMSplex. For example, use this command to restart communication when all activity for the IMSplex in the IMS Connect is terminated, or after a STOPIP command has terminated communication with the IMSplex IMS that contains OM.

Use the VIEWIP command to display information about the IMSplex if you are not sure about the activity of the IMSplex.

The OPENIP command does not affect the IMSplex if the IMSplex is already active or if the IMSplex is not defined to IMS Connect in the configuration member HWSCFGxx.

Equivalent IMS Connect type-2 and z/OS commands

The following table lists IMS Connect type-2 commands and IMS Connect z/OS commands that perform similar functions as the OPENIP command.

OPENIP command	Equivalent IMS Connect type-2 command	Equivalent IMS Connect z/OS command
OPENIP *imsplex_id*	UPDATE IMSCON TYPE(IMSPLEX) NAME(*imsplex_name*) START(COMM)	UPDATE IMSPLEX NAME(*imsplex_name*) START(COMM)

Examples

To open communication to IMSplex with TMEMBER name of IMSPLEX1:

```
nnOPENIP IMSPLEX1
```

Related reference:

UPDATE IMSCON TYPE(IMSPLEX) command (Commands)

"IMS Connect UPDATE IMSPLEX command" on page 592

"STARTIP command" on page 462

OPENPORT command

Use the OPENPORT command to reestablish IMS Connect communication with TCP/IP to enable listening on TCP/IP ports.

Subsections:

- "Syntax" on page 450
- "Keywords" on page 450
- "Usage notes" on page 450
- "Equivalent IMS Connect type-2 and z/OS commands" on page 450
- "Examples" on page 450

Syntax

Keywords

The following keywords are valid for the OPENPORT command:

nn 2-digit number of z/OS system console outstanding display.

portid
Identifies the number of the port to be opened. This port number must match one of the port numbers that is defined in the PORTID substatement of the TCPIP configuration statement in the HWSCFGxx configuration member. For the local option port, specify a portid value of LOCAL. Wildcards (* and %) are supported in the number.

Usage notes

Use this command to reestablish a TCP/IP connection to enable listening on a TCP/IP port. Use this command when communication stops between the IMS Connect and a TCP/IP port, but the IMS Connect has not terminated.

Equivalent IMS Connect type-2 and z/OS commands

The following table lists IMS Connect type-2 commands and IMS Connect z/OS commands that perform similar functions as the OPENPORT command.

OPENPORT command	Equivalent IMS Connect type-2 command	Equivalent IMS Connect z/OS command
OPENPORT *port_id*	UDPATE IMSCON TYPE(PORT) NAME(*port_name*) START(COMM)	UPDATE PORT NAME(*port_name*) START(COMM)

Examples

To reestablish the TCP/IP connection between the IMS Connect and port 9999 so that the IMS Connect can listen on that port:

```
nnOPENPORT 9999
```

Where *nn* is the 2-digit number of z/OS system console outstanding display.

Related reference:

UPDATE IMSCON TYPE(PORT) command (Commands)

"IMS Connect UPDATE PORT command" on page 601

"STARTPT command" on page 466

RECORDER command

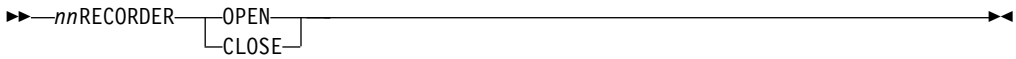
Use the RECORDER command to open and close the line trace data set.

Recommendation: Use BPE to manage the output of the IMS Connect Recorder Trace facility in an external trace data set.

Subsections:

- "Syntax"
- "Keywords"
- "Equivalent IMS Connect type-2 and z/OS commands"
- "Examples"

Syntax



Keywords

The following keywords are valid for the RECORDER command:

nn 2-digit number of z/OS system console outstanding display.

OPEN
Specifies to open the line trace data set.

CLOSE
Specifies to close the line trace data set.

Equivalent IMS Connect type-2 and z/OS commands

The following table lists IMS Connect type-2 commands and IMS Connect z/OS commands that perform similar functions as the RECORDER command.

RECORDER command	Equivalent IMS Connect type-2 command	Equivalent IMS Connect z/OS command
RECORDER OPEN	UPDATE IMSCON TYPE(CONFIG) START(RECORDER)	UPDATE MEMBER TYPE(IMSCON) START(TRACE)
RECORDER CLOSE	UPDATE IMSCON TYPE(CONFIG) STOP(RECORDER)	UPDATE MEMBER TYPE(IMSCON) STOP(TRACE)

Examples

```
nnRECORDER OPEN
nnRECORDER CLOSE
```

Where *nn* is the 2-digit number of z/OS system console outstanding display.

Related concepts:

IMS Connect traces (Diagnosis)

Related reference:

UPDATE IMSCON TYPE(CONFIG) command (Commands)

"IMS Connect UPDATE MEMBER command" on page 594

REFRESH CONVERTER command

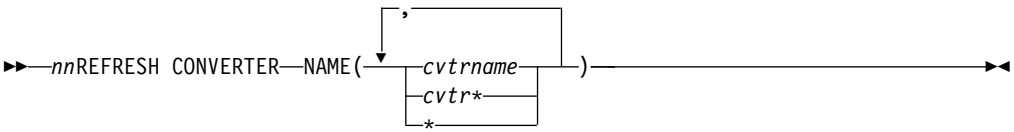
Use the REFRESH CONVERTER command to refresh one or more XML converters without restarting IMS Connect.

This command is equivalent to the z/OS Modify command UPDATE CONVERTER.

Subsections:

- "Syntax"
- "Keywords"
- "Usage notes"
- "Equivalent IMS Connect type-2 and z/OS commands"
- "Examples"

Syntax



Keywords

The following keywords are valid for the REFRESH CONVERTER command:

nn 2-digit number of z/OS system console outstanding display.

NAME(*cvtrname*)
Specify one or more XML converters to be refreshed. The wildcard symbol (*) is supported for pattern matching. Specifying only a wildcard refreshes all XML converters. The total length of all converter names specified on the NAME parameter, including the comma separators, cannot exceed 26 characters in length.

Usage notes

When the REFRESH CONVERTER command is issued, IMS Connect refreshes the specified XML converter. If a wildcard is specified, IMS Connect refreshes all XML converters that match the mask.

Equivalent IMS Connect type-2 and z/OS commands

The following table lists IMS Connect type-2 commands and IMS Connect z/OS commands that perform similar functions as the REFRESH CONVERTER command.

REFRESH CONVERTER command	Equivalent IMS Connect type-2 command	Equivalent IMS Connect z/OS command
REFRESH CONVERTER NAME(*converter_name*)	UPDATE IMSCON TYPE(CONVERTER) NAME(*converter_name*) OPTION(REFRESH)	UPDATE CONVERTER NAME(*converter_name*) OPTION(REFRESH)

Examples

Example input for the REFRESH CONVERTER command:

```
nnREFRESH CONVERTER NAME(IMSPHBKD)
```

Example output for the REFRESH CONVERTER command:

```
HWSA0375I XML CONVERTER IMSPHBKD WAS SUCCESSFULLY REFRESHED,M=XMLA
```

Related tasks:

Configuring XML conversion support for IMS Connect clients (System Definition)

Related reference:

UPDATE IMSCON TYPE(CONVERTER) command (Commands)

"IMS Connect UPDATE CONVERTER command" on page 589

REFRESH RACFUID command

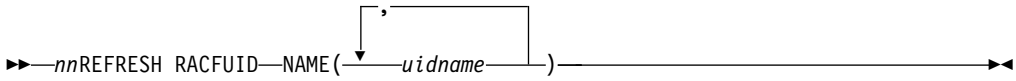
Use the REFRESH RACFUID command to refresh one or more cached IDs stored in the IMS Connect RACF user ID cache.

This command is equivalent to the UPDATE RACFUID z/OS Modify command.

Subsections:

- "Syntax"
- "Keywords"
- "Usage notes"
- "Equivalent IMS Connect type-2 and z/OS commands"
- "Examples" on page 454

Syntax



Keywords

The following keywords are valid for the REFRESH RACFUID command:

nn 2-digit number of z/OS system console outstanding display.

NAME(*uidname*)
Specify one or more RACF user IDs to be refreshed. The total length of all user IDs, including the comma separators, cannot exceed 26 characters in length.

Usage notes

When the REFRESH RACFUID command is issued, IMS Connect refreshes the specified ID by issuing the RACROUTE REQUEST=VERIFY command to RACF for the specified ID.

Equivalent IMS Connect type-2 and z/OS commands

The following table lists IMS Connect type-2 commands and IMS Connect z/OS commands that perform similar functions as the REFRESH RACFUID command.

REFRESH RACFUID command	Equivalent IMS Connect type-2 command	Equivalent IMS Connect z/OS command
REFRESH RACFUID NAME(*userid*)	UPDATE IMSCON TYPE(RACFUID) NAME(*userid*) OPTION(REFRESH)	UPDATE RACFUID NAME(*userid*) OPTION(REFRESH)

Examples

Example input for the REFRESH RACFUID command:

```
nnREFRESH RACFUID NAME(USRT001)
```

Example output for the REFRESH RACFUID command:

```
HWSP1504I RACF USERID USRT001 WAS SUCCESSFULLY REFRESHED,M=CUPR
```

Related concepts:

IMS Connect RACF user ID cache (Communications and Connections)

Related reference:

UPDATE IMSCON TYPE(RACFUID) command (Commands)

"IMS Connect UPDATE RACFUID command" on page 602

SETOAUTO command

Use the SETOAUTO command to turn on and off automatic connection with an Open Database Manager (ODBM).

Subsections:

- "Syntax"
- "Keywords"
- "Usages"
- "Equivalent IMS Connect type-2 and z/OS commands" on page 455
- "Examples" on page 455

Syntax

Keywords

The following keywords are valid for the SETOAUTO command:

nn 2-digit number of z/OS system console outstanding display.

YES | NO
Specifies to turn on or off automatic connection with ODBM.

Usages

You can use the SETOAUTO command to specify whether IMS Connect automatically connects to ODBM when either IMS Connect is started or an instance of ODBM is activated in the IMSplex.

When automatic connection with ODBM is set to YES, IMS Connect will connect to all the future ODBMs that enter the IMSplex. This option can be specified in the ODBMAUTOCONN= parameter of the IMS Connect configuration member.

When automatic connection with ODBM is set to NO, IMS Connect will not register with any future ODBMs that enter the IMSplex. After turning off the automatic connection of IMS Connect to ODBM, you can connect to ODBM manually by using the STARTOD command to open communication with an instance of ODBM.

Equivalent IMS Connect type-2 and z/OS commands

The following table lists IMS Connect type-2 commands and IMS Connect z/OS commands that perform similar functions as the SETOAUTO command.

SETOAUTO command	Equivalent IMS Connect type-2 command	Equivalent IMS Connect z/OS command
SETOAUTO YES	UPDATE IMSCON TYPE(CONFIG) SET(OAUTO(ON))	UPDATE MEMBER TYPE(IMSCON) SET(OAUTO(ON))
SETOAUTO NO	UPDATE IMSCON TYPE(CONFIG) SET(OAUTO(OFF)	UPDATE MEMBER TYPE(IMSCON) SET(OAUTO(OFF))

Examples

To set automatic connections to future ODBMs that enter the IMSplex:

```
nnSETOAUTO YES
```

Related reference:

UPDATE IMSCON TYPE(CONFIG) command (Commands)

"IMS Connect UPDATE MEMBER command" on page 594

SETPWMC command

If the security product that you use, such as RACF, supports mixed-case passwords, the SETPWMC command enables IMS Connect support for mixed-case passwords.

By default, IMS Connect determines the need for mixed-case password support based on the specification for mixed-case password support in the RACF configuration.

You can also use the SETPWMC command to disable IMS Connect support for mixed-case passwords. When IMS Connect support for mixed-case passwords is disabled, IMS Connect converts all lowercase characters in a password to uppercase characters.

Subsections:

- "Syntax" on page 456
- "Keywords" on page 456
- "Equivalent IMS Connect type-2 and z/OS commands" on page 456
- "Examples" on page 456

Syntax

Keywords

The following keywords are valid for the SETPWMC command:

nn 2-digit number of z/OS system console outstanding display.

ON | OFF | RCF
Identifies if IMS Connect support for mixed-case passwords is turned on or off, or if it depends on the mixed-case password specification in RACF. If mixed-case password support is turned off, IMS Connect converts any lowercase characters in passwords to uppercase characters. If mixed-case password support depends on the mixed-case password specification in RACF and if the mixed-case password specification in RACF is not set, IMS Connect converts any lowercase characters in passwords to uppercase characters.

Equivalent IMS Connect type-2 and z/OS commands

The following table lists IMS Connect type-2 commands and IMS Connect z/OS commands that perform similar functions as the SETPWMC command.

SETPWMC command	Equivalent IMS Connect type-2 command	Equivalent IMS Connect z/OS command
SETPWMC ON	UPDATE IMSCON TYPE(CONFIG) SET(PSWDMC(ON))	UPDATE MEMBER TYPE(IMSCON) SET(PSWDMC(ON))
SETPWMC OFF	UPDATE IMSCON TYPE(CONFIG) SET(PSWDMC(OFF))	UPDATE MEMBER TYPE(IMSCON) SET(PSWDMC(OFF))
SETPWMC RCF	UPDATE IMSCON TYPE(CONFIG) SET(PSWDMC(RCF))	UPDATE MEMBER TYPE(IMSCON) SET(PSWDMC(RCF))

Examples

To enable support for mixed-case passwords, type the following command:

```
nnSETPWMC ON
```

Where *nn* is the 2-digit number of z/OS system console outstanding display.

Related reference:

UPDATE IMSCON TYPE(CONFIG) command (Commands)

"IMS Connect UPDATE MEMBER command" on page 594

SETRACF command

Use the SETRACF command to turn on and off the RACF flag.

Subsections:

- "Syntax"
- "Keywords"
- "Usage notes"
- "Equivalent IMS Connect type-2 and z/OS commands"
- "Examples"

Syntax

Keywords

The following keywords are valid for the SETRACF command:

nn 2-digit number of z/OS system console outstanding display.

ON | OFF
Identifies whether the RACF flag is turned on or off.

Usage notes

Use this command to enable or disable the RACF user identification and verification.

Equivalent IMS Connect type-2 and z/OS commands

The following table lists IMS Connect type-2 commands and IMS Connect z/OS commands that perform similar functions as the SETRACF command.

SETRACF command	Equivalent IMS Connect type-2 command	Equivalent IMS Connect z/OS command
SETRACF ON	UPDATE IMSCON TYPE(CONFIG) SET(RACF(ON))	UPDATE MEMBER TYPE(IMSCON) SET(RACF(ON))
SETRACF OFF	UPDATE IMSCON TYPE(CONFIG) SET(RACF(OFF))	UPDATE MEMBER TYPE(IMSCON) SET(RACF(OFF))

Examples

To turn on the RACF:

```
nnSETRACF ON
```

Where *nn* is the 2-digit number of z/OS system console outstanding display.

Related reference:

UPDATE IMSCON TYPE(CONFIG) command (Commands)

"IMS Connect UPDATE MEMBER command" on page 594

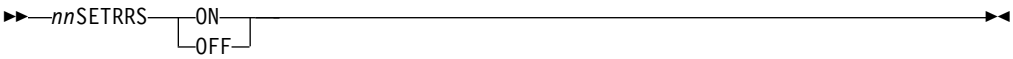
SETRRS command

Use the SETRRS command to enable or disable communication between IMS Connect and z/OS Resource Recovery Services (RRS).

Subsections:

- "Syntax"
- "Keywords"
- "Usage notes"
- "Equivalent IMS Connect type-2 and z/OS commands"
- "Examples"

Syntax



Keywords

The following keywords are valid for the SETRRS command:

nn 2-digit number of z/OS system console outstanding display.

ON | OFF
Identifies whether to enable RRS communication.

Usage notes

To enable or disable communication between IMS Connect and RRS. RRS is required for two-phase-commit support.

Equivalent IMS Connect type-2 and z/OS commands

The following table lists IMS Connect type-2 commands and IMS Connect z/OS commands that perform similar functions as the SETRRS command.

SETRRS command	Equivalent IMS Connect type-2 command	Equivalent IMS Connect z/OS command
SETRRS ON	UPDATE IMSCON TYPE(CONFIG) SET(RRS(ON))	UPDATE MEMBER TYPE(IMSCON) SET(RRS(ON))
SETRRS OFF	UPDATE IMSCON TYPE(CONFIG) SET(RRS(OFF))	UPDATE MEMBER TYPE(IMSCON) SET(RRS(OFF))

Examples

To disable communication between IMS Connect and RRS:

```
nnSETRRS OFF
```

Where *nn* is the 2-digit number of z/OS system console outstanding display.

Related reference:

UPDATE IMSCON TYPE(CONFIG) command (Commands)

"IMS Connect UPDATE MEMBER command" on page 594

SETUIDC command

Use the SETUIDC command to enable or disable the IMS Connect RACF user ID cache.

This command is equivalent to the SET UIDCACHE statement of the UPDATE MEMBER z/OS Modify command.

Subsections:

- "Syntax"
- "Keywords"
- "Usage notes"
- "Equivalent IMS Connect type-2 and z/OS commands"
- "Examples" on page 460

Syntax

Keywords

The following keywords are valid for the SETUIDC command:

nn 2-digit number of z/OS system console outstanding display.

ON | OFF
Enables or disables the IMS Connect RACF user ID cache.

Usage notes

This command is valid only if IMS Connect support for RACF is enabled.

Equivalent IMS Connect type-2 and z/OS commands

The following table lists IMS Connect type-2 commands and IMS Connect z/OS commands that perform similar functions as the SETUIDC command.

SETUIDC command	Equivalent IMS Connect type-2 command	Equivalent IMS Connect z/OS command
SETUIDC ON	UPDATE IMSCON TYPE(CONFIG) SET(UIDCACHE(ON))	UPDATE MEMBER TYPE(IMSCON) SET(UIDCACHE(ON))
SETUIDC OFF	UPDATE IMSCON TYPE(CONFIG) SET(UIDCACHE(OFF))	UPDATE MEMBER TYPE(IMSCON) SET(UIDCACHE(OFF))

Examples

Example input for the SETUIDC command:

```
nnSETUIDC ON
```

Example output for the SETUIDC command:

```
HWSP1501I RACF USERID CACHING ENABLED,M=SDRC
```

Related concepts:

IMS Connect RACF user ID cache (Communications and Connections)

Related reference:

UPDATE IMSCON TYPE(CONFIG) command (Commands)

"IMS Connect UPDATE MEMBER command" on page 594

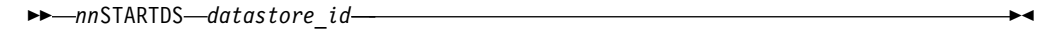
STARTDS command

Use the STARTDS command to start communication between the IMS Connect and a data store.

Subsections:

- "Syntax"
- "Keywords"
- "Usage notes"
- "Equivalent IMS Connect type-2 and z/OS commands" on page 461
- "Examples" on page 461

Syntax



Keywords

The following keywords are valid for the STARTDS command:

nn 2-digit number of z/OS system console outstanding display.

datastore_id
Specifies the name of the data store. This name must be defined to the IMS Connect through the configuration member HWSCFGxx, and must match one of the IDs that is defined in the DATASTORE configuration statement or statements.

Usage notes

Use this command to reestablish communication with a data store after communication fails between the IMS Connect and the data store. For example, use this command to restart communication when all activity for the data store in the IMS Connect is terminated, or after a STOPDS command has terminated communication with the data store.

Use the VIEWDS command to display information about data stores if you are not sure about the activity of a particular data store.

The STARTDS command does not affect a data store that is already active or a data store that is not defined to the IMS Connect in the configuration member HWSCFGxx.

Equivalent IMS Connect type-2 and z/OS commands

The following table lists IMS Connect type-2 commands and IMS Connect z/OS commands that perform similar functions as the STARTDS command.

STARTDS command	Equivalent IMS Connect type-2 command	Equivalent IMS Connect z/OS command
STARTDS *datastore_id*	UPDATE IMSCON TYPE(DATASTORE) NAME(*datastore_name*) START(COMM)	UPDATE DATASTORE NAME(*datastoreName*) START(COMM)

Examples

To open communication to data store IMSA:

```
nnSTARTDS IMSA
```

Where *nn* is the 2-digit number of z/OS system console outstanding display.

Related reference:

UPDATE IMSCON TYPE(DATASTORE) command (Commands)

"OPENDS command" on page 447

"IMS Connect UPDATE DATASTORE command" on page 591

STARTIA command

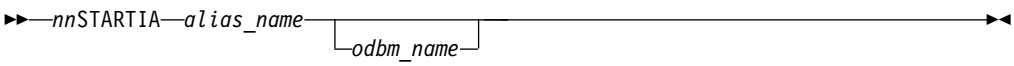
Use the STARTIA command to enable message routing to an IMS alias that is managed by one or more Open Database Manager (ODBM) instances. Alias represents the IMS data store that the client wants to send the message to.

The STARTIA command sets the specified alias to active so that IMS Connect can route to that alias.

Subsections:

- "Syntax"
- "Keywords"
- "Usage notes" on page 462
- "Equivalent IMS Connect type-2 and z/OS commands" on page 462
- "Examples" on page 462

Syntax



Keywords

The following keywords are valid for the STARTIA command:

nn 2-digit number of z/OS system console outstanding display.

alias_name
Specifies the name of the IMS alias. Wildcards (* and %) are supported in the name.

odbm_name
Specifies the name of the ODBM. Wildcards (* and %) are supported in the name. This parameter is optional.

Usage notes

Whereas the STARTOD command starts communication between the IMS Connect and an IMS Open Database Manager (ODBM), the STARTIA command sets the specified IMS alias to active so that IMS Connect can route to that alias.

Equivalent IMS Connect type-2 and z/OS commands

The following table lists IMS Connect type-2 commands and IMS Connect z/OS commands that perform similar functions as the STARTIA command.

STARTIA command	Equivalent IMS Connect type-2 command	Equivalent IMS Connect z/OS command
STARTIA *alias_name odbm_name*	UDPATE IMSCON TYPE(ALIAS) NAME(*alias_name*) ODBM(*odbm_name*) START(COMM)	UPDATE ALIAS NAME(*aliasName*) ODBM(*odbmName*) START(ROUTE)

Examples

To activate an alias named IMS1 for ODBM named ODBM1OD:

```
nnSTARTIA IMS1 ODBM1OD
```

Related reference:

"STOPIA command" on page 471

UPDATE IMSCON TYPE(ALIAS) command (Commands)

"IMS Connect UPDATE ALIAS command" on page 588

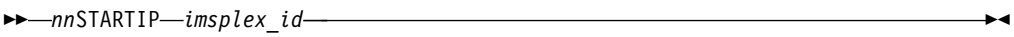
STARTIP command

Use the STARTIP command to start communication between the IMS Connect and the IMSplex that contains OM which is connected to SCI.

Subsections:

- "Syntax"
- "Keywords" on page 463
- "Usage notes" on page 463
- "Equivalent IMS Connect type-2 and z/OS commands" on page 463
- "Examples" on page 463

Syntax



Keywords

The following keywords are valid for the STARTIP command:

nn 2-digit number of z/OS system console outstanding display.

imsplex_id
Specifies the name of the IMSplex. This name must be defined to the IMS Connect through the configuration member HWSCFGxx, and must match the TMEMBER that is defined in the IMSplex configuration statement.

Usage notes

Use this command to reestablish communication with the IMSplex that is being used to communicate with OM if communication has failed between IMS Connect and the IMSplex. For example, use this command to restart communication when all activity for the IMSplex in the IMS Connect is terminated, or after a STOPIP command has terminated communication with the IMSplex IMS that contains OM.

If the IMS Connect instance supports ISC or MSC TCP/IP links, IMS Connect changes the status of the links that it supports in the IMSplex to ACTIVE.

Use the VIEWIP command to display information about the IMSplex if you are not sure about the activity of the IMSplex.

The STARTIP command does not affect the IMSplex if the IMSplex is already active or if the IMSplex is not defined to IMS Connect in the configuration member HWSCFGxx.

Equivalent IMS Connect type-2 and z/OS commands

The following table lists IMS Connect type-2 commands and IMS Connect z/OS commands that perform similar functions as the STARTIP command.

STARTIP command	Equivalent IMS Connect type-2 command	Equivalent IMS Connect z/OS command
STARTIP *imsplex_id*	UPDATE IMSCON TYPE(IMSPLEX) NAME(*imsplex_name*) START(COMM)	UPDATE IMSPLEX NAME(*imsplex_name*) START(COMM)

Examples

To open communication to IMSplex with TMEMBER name of IMSPLEX1:

```
nnSTARTIP IMSPLEX1
```

Where *nn* is the 2-digit number of z/OS system console outstanding display.

Related reference:

UPDATE IMSCON TYPE(IMSPLEX) command (Commands)

"OPENIP command" on page 448

"IMS Connect UPDATE IMSPLEX command" on page 592

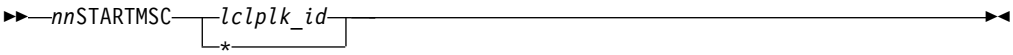
STARTMSC command

Use the STARTMSC command in a local instance of IMS Connect to resume communications on an MSC physical link after the communications on the physical link are stopped by a STOPMSC command or an UPDATE MSC STOP(COMM) command.

Subsections:

- "Syntax"
- "Keywords"
- "Usage notes"
- "Equivalent IMS Connect type-2 and z/OS commands"
- "Examples"

Syntax



Keywords

The following keywords are valid for the STARTMSC command:

nn 2-digit number of z/OS system console outstanding display.

lclplk_id
The ID of the MSC physical link, as specified on the LCLPLKID parameter of the MSC statement that defines the physical link to IMS Connect.

Usage notes

When the STARTMSC command is issued, IMS Connect resumes communications on the specified MSC physical link and changes the status of the physical link to ACTIVE.

Use the VIEWMSC command or the QUERY MSC command to display information about MSC physical links that are defined to IMS Connect.

Equivalent IMS Connect type-2 and z/OS commands

The following table lists IMS Connect type-2 commands and IMS Connect z/OS commands that perform similar functions as the STARTMSC command.

STARTMSC command	Equivalent IMS Connect type-2 command	Equivalent IMS Connect z/OS command
STARTMSC *lclplk_id*	UPDATE IMSCON TYPE(MSC) NAME(*lclPlkid*) START(COMM)	UPDATE MSC NAME(*lclPlkid*) START(COMM)

Examples

Example input for the STARTMSC command:

```
nnSTARTMSC MSC12
```

Example output for the STARTMSC command:

```
HWSF3300I COMMUNICATIONS ON MSC PHYSICAL LINK MSC12 STARTED; M=ISC1
```

Related reference:

"IMS Connect UPDATE MSC command" on page 597

UPDATE IMSCON TYPE(MSC) command (Commands)

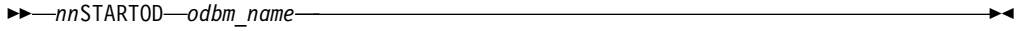
STARTOD command

Use the STARTOD command to start communication between the IMS Connect and an IMS Open Database Manager (ODBM).

Subsections:

- "Syntax"
- "Keywords"
- "Usage notes"
- "Equivalent IMS Connect type-2 and z/OS commands" on page 466
- "Examples" on page 466

Syntax



Keywords

The following keywords are valid for the STARTOD command:

nn 2-digit number of z/OS system console outstanding display.

odbm_name
Specifies the name of the ODBM. Wildcards (* and %) are supported in the name.

Usage notes

Use the STARTOD command to reestablish communication with an ODBM after communication fails between the IMS Connect and the ODBM. For example, use this command to restart communication when all activities for the ODBM in the IMS Connect is terminated, or after a STOPOD command has terminated communication with the ODBM. The STARTOD command makes a connection to ODBM and automatically sets all the aliases of that ODBM to active so that messages can be routed to those aliases. After the STARTOD command is issued, the ODBM status will be REGISTERED.

If ODBMAUTOCONN=N is set, you can manually start communication with an ODBM by issuing the STARTOD command. If, for whatever reason, you want to stop routing messages to an ODBM, issue the STOPOD command. If you want to resume routing messages to that ODBM, issue the STARTOD command to start the communication. If you want to stop routing messages to a particular alias, issue the STOPIA command. You can resume that alias routing by issuing the STARTIA command, which sets the specified IMS alias to active so that IMS Connect can route to that alias.

Note: If you stop routing messages to a particular alias by using the STOPIA command, and then issue the STOPOD command and the STARTOD command, the alias status will be lost across STOPOD and STARTOD. In other words, the alias will be active again after the STARTOD command.

Use the VIEWDS command to display information about ODBMs if you are not sure about the activity of a particular ODBM. The STARTOD command does not affect an ODBM that is already active.

Equivalent IMS Connect type-2 and z/OS commands

The following table lists IMS Connect type-2 commands and IMS Connect z/OS commands that perform similar functions as the STARTOD command.

STARTOD command	Equivalent IMS Connect type-2 command	Equivalent IMS Connect z/OS command
STARTOD *odbm_name*	UDPATE IMSCON TYPE(ODBM) NAME(*odbm_name*) START(COMM)	UPDATE ODBM NAME(*odbmName*) START(COMM)

Examples

To open communication to ODBM named ODBM1OD:

```
nnSTARTOD ODBM1OD
```

Related reference:

"STOPOD command" on page 477

UPDATE IMSCON TYPE(ODBM) command (Commands)

"IMS Connect UPDATE ODBM command" on page 599

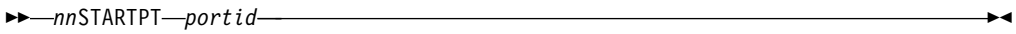
STARTPT command

Use the STARTPT command to reestablish IMS Connect communication with TCP/IP to enable listening on TCP/IP ports.

Subsections:

- "Syntax"
- "Keywords"
- "Usage notes" on page 467
- "Equivalent IMS Connect type-2 and z/OS commands" on page 467
- "Examples" on page 467

Syntax



Keywords

The following keywords are valid for the STARTPT command:

nn 2-digit number of z/OS system console outstanding display.

portid
Identifies the number of the port to be opened. This port number must match one of the port numbers that is defined in either the PORTID or CICSPORT substatement of the TCPIP configuration statement in the `HWSCFGxx` configuration member. For the local option port, specify a *portid* value of LOCAL. Wildcards (* and %) are supported in the number.

Usage notes

Use this command to reestablish a TCP/IP connection to enable listening on a TCP/IP port. Use this command when communication stops between the IMS Connect and a TCP/IP port, but the IMS Connect has not terminated.

Equivalent IMS Connect type-2 and z/OS commands

The following table lists IMS Connect type-2 commands and IMS Connect z/OS commands that perform similar functions as the STARTPT command.

STARTPT command	Equivalent IMS Connect type-2 command	Equivalent IMS Connect z/OS command
STARTPT *port_id*	UDPATE IMSCON TYPE(PORT) NAME(*port_name*) START(COMM)	UPDATE PORT NAME(*port_name*) START(COMM)

Examples

To reestablish the TCP/IP connection between the IMS Connect and port 9999 so that the IMS Connect can listen on that port:

```
nnSTARTPT 9999
```

Where *nn* is the 2-digit number of z/OS system console outstanding display.

Related reference:

"OPENPORT command" on page 449

"IMS Connect UPDATE PORT command" on page 601

UPDATE IMSCON TYPE(PORT) command (Commands)

STARTRMT command

Use the STARTRMT command in a local instance of IMS Connect to resume communications with a remote instance of IMS Connect after communications have been stopped by either a STOPRMT or a UPDATE RMTIMSCON STOP(COMM) command.

Subsections:

- "Syntax" on page 468
- "Keywords" on page 468
- "Usage notes" on page 468
- "Equivalent IMS Connect type-2 and z/OS commands" on page 468
- "Examples" on page 469

Syntax

Keywords

The following keywords are valid for the STARTRMT command:

nn The two-digit number of the z/OS system console outstanding display.

rmtimscon_id
The name of the remote IMS Connect connection to start, as specified on the ID parameter of the RMTIMSCON statement that defines the connection to the local IMS Connect instance.

Usage notes

When the STARTRMT command is issued, IMS Connect resumes communications on the specified connection to a remote IMS Connect instance. IMS Connect changes the status of the connection to NOT ACTIVE until new sockets are opened for communications with the remote instance of IMS Connect.

The STARTRMT command is required only when communications on a connection with a remote IMS Connect instance have been previously stopped by either a STOPRMT command or an UPDATE RMTIMSCON STOP(COMM) command.

If the specified RMTIMSCON connection is defined with AUTOCONN=Y, when the STARTRMT is issued, IMS Connect automatically creates socket connections to the remote IMS Connect. The number of socket connections IMS Connect opens is determined by the RESVSOC parameter.

OTMA communications can resume as soon as IMS Connect executes the STARTRMT command.

For MSC communications, the STARTRMT command resumes only TCP/IP communications in IMS Connect, and does not change the stopped status of MSC links. If the MSC links have been stopped by either a STOPMSC command or an UPDATE MSC STOP(COMM) command, after you issue the STARTRMT command, you also need to restart the MSC links in IMS Connect by issuing either the STARTMSC command or the UPDATE MSC START(COMM) command.

Recommendation: When restarting MSC communications, to prevent IMS from sending MSC messages to IMS Connect before TCP/IP communications have been restarted, always issue the STARTRMT command before issuing the STARTMSC command.

Use the VIEWRMT command, the QUERY IMSCON TYPE(RMTIMSCON) command, or the QUERY RMTIMSCON command to display information about the current connections to a remote IMS Connect instance that are defined to the local IMS Connect instance.

Equivalent IMS Connect type-2 and z/OS commands

The following table lists IMS Connect type-2 commands and IMS Connect z/OS commands that perform similar functions as the STARTRMT command.

STARTRMT command	Equivalent IMS Connect type-2 command	Equivalent IMS Connect z/OS command
STARTRMT *rmtimsconName*	UPDATE IMSCON TYPE(RMTIMSCON) NAME(*rmtimscon_name*) START(COMM)	UPDATE RMTIMSCON NAME(*rmtimsconName*) START(COMM)

Examples

The following example of the STARTRMT command starts the connection CONNECT2:

```
nnSTARTRMT CONNECT2
```

STARTRMT command output:

```
HWST3500I COMMUNICATIONS WITH REMOTE IMS CONNECT CONNECT2 STARTED; M=TSCH
```

Related reference:

"STOPRMT command" on page 480

"IMS Connect UPDATE RMTIMSCON command" on page 603

UPDATE IMSCON TYPE(RMTIMSCON) command (Commands)

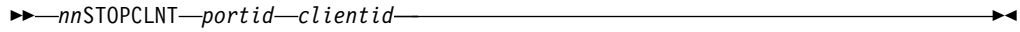
STOPCLNT command

Use the STOPCLNT command to immediately terminate communication with a client using a specific TCP/IP port.

Subsections:

- "Syntax"
- "Keywords"
- "Usage notes" on page 470
- "Equivalent IMS Connect type-2 and z/OS commands" on page 470
- "Examples" on page 470

Syntax



Keywords

The following keywords are valid for the STOPCLNT command:

nn 2-digit number of z/OS system console outstanding display.

portid
Identifies the port that the client is using for the TCP/IP connection with the IMS Connect. This port number must match a port number that is defined in either the PORTID or CICSPORT substatement of the TCPIP configuration statement in the HWSCFGxx configuration member. For the local option port, specify a *portid* value of LOCAL.

clientid
Specifies the name of the client (the client name is dynamically generated by IMS TM Resource Adapter).

Usage notes

Work currently in progress for that client is ended.

Use this command whenever a client is unable to accept response messages being sent to it, or when a client is waiting for a nonexistent response message (for example, when an error occurred that caused a response message to be lost before it was sent back to the client).

Use the VIEWPORT command to display the name and state of the client.

Equivalent IMS Connect type-2 and z/OS commands

The following table lists IMS Connect type-2 commands and IMS Connect z/OS commands that perform similar functions as the STOPCLNT command.

STOPCLNT command	Equivalent IMS Connect type-2 command	Equivalent IMS Connect z/OS command
STOPCLNT *portid clientid*	UPDATE IMSCON TYPE(CLIENT) NAME(*client_name*) PORT(*portid*) STOP(COMM)	DELETE PORT NAME(*portName*) CLIENT(*clientName*)

Examples

To force the IMS Connect to terminate communication with client `CLIENT01`, who is communicating with the IMS Connect using port `9999`:

```
nnSTOPCLNT 9999 CLIENT01
```

Where *nn* is the 2-digit number of z/OS system console outstanding display.

Related reference:

UPDATE IMSCON TYPE(CLIENT) command (Commands)

"IMS Connect DELETE CLIENT command" on page 534

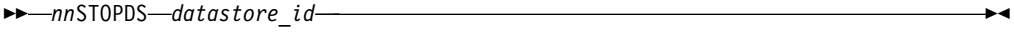
STOPDS command

Use the STOPDS command to immediately terminate communication between the IMS Connect and a data store.

Subsections:

- "Syntax"
- "Keywords"
- "Usage notes" on page 471
- "Equivalent IMS Connect type-2 and z/OS commands" on page 471
- "Examples" on page 471

Syntax



Keywords

The following keywords are valid for the STOPDS command:

nn 2-digit number of z/OS system console outstanding display.

datastore_id
Specifies the name of the data store. This name must match an ID that is defined in a DATASTORE configuration statement of the `HWSCFGxx` configuration member.

Usage notes

Work currently in progress for a data store is ended and communications with that data store and its threads are terminated. Messages that are queued for the data store are released and the originator of the queued messages is notified. No new messages are accepted after the STOPDS command is accepted.

Use this command to release messages that are queued for an unavailable data store or for a data store whose queued work belongs to unavailable clients. It can also be used for any type of error situation that requires immediate termination of communication with a data store.

Use the OPENDS command to open communication with the data store at a later time.

Equivalent IMS Connect type-2 and z/OS commands

The following table lists IMS Connect type-2 commands and IMS Connect z/OS commands that perform similar functions as the STOPDS command.

STOPDS command	Equivalent IMS Connect type-2 command	Equivalent IMS Connect z/OS command
STOPDS *datastore_id*	UPDATE IMSCON TYPE(DATASTORE) NAME(*datastore_name*) STOP(COMM)	UPDATE DATASTORE NAME(*datastoreName*) STOP(COMM)

Examples

To stop communication to data store IMSA:

```
nnSTOPDS IMSA
```

Where *nn* is the 2-digit number of z/OS system console outstanding display.

Related reference:

UPDATE IMSCON TYPE(DATASTORE) command (Commands)

"IMS Connect UPDATE DATASTORE command" on page 591

STOPIA command

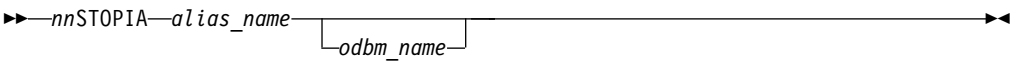
Use the STOPIA command to disable message routing to an IMS alias that is managed by one or more Open Database Manager (ODBM) instances. The STOPIA command sets the specified alias to inactive so that IMS Connect cannot route to that alias.

Subsections:

- "Syntax" on page 472
- "Keywords" on page 472
- "Usage notes" on page 472

- "Equivalent IMS Connect type-2 and z/OS commands"
- "Examples"

Syntax



Keywords

The following keywords are valid for the STOPIA command:

nn 2-digit number of z/OS system console outstanding display.

alias_name
Specifies the name of the IMS alias. Wildcards (* and %) are supported in the name.

odbm_name
Specifies the name of the ODBM. Wildcards (* and %) are supported in the name. This parameter is optional.

Usage notes

If you want to stop routing messages to a particular alias, then issue the STOPIA command. You can resume that alias routing by issuing the STARTIA command, which sets the specified IMS alias to active so that IMS Connect can route to that alias.

Note: If you stop routing messages to a particular alias by using the STOPIA command, and then issue the STOPOD command and the STARTOD command, the alias status will be lost across STOPOD and STARTOD. In other words, the alias will be active again after the STARTOD command.

Equivalent IMS Connect type-2 and z/OS commands

The following table lists IMS Connect type-2 commands and IMS Connect z/OS commands that perform similar functions as the STOPIA command.

STOPIA command	Equivalent IMS Connect type-2 command	Equivalent IMS Connect z/OS command
STOPIA *alias_name odbm_name*	UDPATE IMSCON TYPE(ALIAS) NAME(*alias_name*) ODBM(*odbm_name*) STOP(COMM)	UPDATE ALIAS NAME(*aliasName*) ODBM(*odbmName*) STOP(ROUTE)

Examples

To deactivate an alias named IMS1 for ODBM named ODBM1OD:

```
nnSTOPIA IMS1 ODBM1OD
```

Related reference:

"STARTIA command" on page 461

UPDATE IMSCON TYPE(ALIAS) command (Commands)

"IMS Connect UPDATE ALIAS command" on page 588

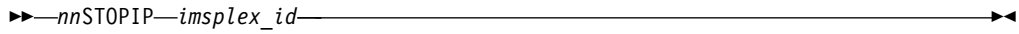
STOPIP command

Use the STOPIP command to stop communication between the IMS Connect and the IMSplex that contains OM which is connected to SCI.

Subsections:

- "Syntax"
- "Keywords"
- "Usage notes"
- "Equivalent IMS Connect type-2 and z/OS commands" on page 474
- "Examples" on page 474

Syntax



Keywords

The following keywords are valid for the STOPIP command:

nn 2-digit number of z/OS system console outstanding display.

imsplex_id
Specifies the name of the IMSplex. This name must be defined to the IMS Connect, through the configuration member HWSCFGxx, and must match the TMEMBER that is defined in the IMSplex configuration statement.

Usage notes

Work currently in progress for the IMSplex ends and communication with the IMSplex and its threads are terminated. Any messages in progress are rejected and an error message is returned to the requester. STOPIP can also be used for any error situation that requires immediate termination of communication with the IMSplex.

If IMS Connect supports either ISC or MSC TCP/IP links and the STOPIP command is used, IMS Connect sends a notification to IMS for each ISC link and MSC physical link. For ISC links, IMS Connect informs IMS that all of the ISC parallel sessions are terminated on the ISC link. For MSC links, IMS Connect informs IMS that all of the MSC logical links on the physical link are terminated.

Stopping IMSplex communications breaks all ISC links and MSC TCP/IP physical links that are supported by the local IMS Connect in the local IMSplex.

When a STOPIP or similar command is issued, the following actions occur for ISC and MSC TCP/IP links:

- The local IMS Connect disconnects the parallel send and receive sockets that support the links and sends a shutdown directive to IMS for ISC and MSC in the local IMS system.
- For ISC links, when IMS receives the shutdown directive, ISC cleans up the ISC link.
- For MSC links, when IMS receives the shutdown directive, MSC cleans up the physical link and issues message DFS3176E.

- IMS Connect changes the status of the ISC links and MSC TCP/IP physical links to DISCONNECTED.

Use the OPENIP command to start communication with the IMSplex at a later time.

Equivalent IMS Connect type-2 and z/OS commands

The following table lists IMS Connect type-2 commands and IMS Connect z/OS commands that perform similar functions as the STOPIP command.

STOPIP command	Equivalent IMS Connect type-2 command	Equivalent IMS Connect z/OS command
STOPIP *imsplex_id*	UPDATE IMSCON TYPE(IMSPLEX) NAME(*imsplex_name*) STOP(COMM)	UPDATE IMSPLEX NAME(*imsplex_name*) STOP(COMM)

Examples

To stop communication to IMSplex with TMEMBER name of IMSPLEX1:

```
nnSTOPIP IMSPLEX1
```

Where *nn* is the 2-digit number of z/OS system console outstanding display.

Related reference:

UPDATE IMSCON TYPE(IMSPLEX) command (Commands)

"IMS Connect UPDATE IMSPLEX command" on page 592

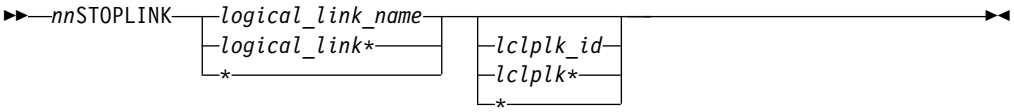
STOPLINK command

Use the IMS Connect WTOR command STOPLINK to stop communications on an MSC logical link that is assigned to an MSC physical link in IMS Connect. STOPLINK performs a similar action as the DELETE LINK z/OS MODIFY command.

Subsections:

- "Syntax"
- "Keywords"
- "Usage notes" on page 475
- "Equivalent IMS Connect type-2 and z/OS commands" on page 475
- "Examples" on page 476

Syntax



Keywords

The following keywords are valid for the STOPLINK command:

nn The two-digit number of the z/OS system console outstanding display.

logical_link_name
The name of the MSC logical link that is being stopped.

lclplk_id
The ID of the MSC physical link on which to stop the logical link is assigned to.

This value is optional. If no physical link ID is specified, IMS Connect searches all the MSC physical links and stops all logical links that have the specified name.

Usage notes

Use the STOPLINK command to clean up the resources associated with an MSC logical link when the link has already been terminated, but the IMS Connect resources associated with the link were not cleaned up correctly.

Recommendation: Use the IMS command /PSTOP to terminate MSC logical links. Use the STOPLINK command only when the IMS Connect resources associated with an MSC logical link that has already been terminated have failed to clean up correctly.

When the STOPLINK command is issued, IMS Connect:

- Stops communication on the MSC logical link
- Informs IMS that communication has stopped on the logical link
- Deletes the control blocks associated with the logical link and frees the associated storage
- Issues message HWSF3310I

To avoid accidentally stopping a logical link that uses the same logical link name on another physical link, limit the command processing to a specific physical link by specifying the name of the target physical link as the *lclplk_id* value.

If the STOPLINK command is issued against two or more physical links, or the *lclplk_id* value is omitted, IMS Connect stops communication on all MSC logical links that match the *logical_link_name* specified on the STOPLINK command.

To display information about the MSC logical links that are assigned to the MSC physical links that are defined to an IMS Connect instance, use any of the following commands:

- In the IMS type-2 command format, either QUERY IMSCON TYPE(LINK) or QUERY IMSCON TYPE(MSC)
- In the WTOR command format, VIEWMSC
- In the z/OS MODIFY command format, QUERY MSC

Equivalent IMS Connect type-2 and z/OS commands

The following table lists IMS Connect type-2 commands and IMS Connect z/OS commands that perform similar functions as the STOPLINK command.

STOPLINK command	Equivalent IMS Connect type-2 command	Equivalent IMS Connect z/OS command
STOPLINK *linkName*	UPDATE IMSCON TYPE(LINK) NAME(*link_name*) STOP(COMM)	DELETE LINK NAME(*linkName*)
STOPLINK *linkname lclPlkid*	UPDATE IMSCON TYPE(LINK) NAME(*linkname*) MSC(*lclPlkid*) STOP(COMM)	DELETE LINK NAME(*linkname*) LCLPLKID(*lclPlkid*)

Examples

In the following example, the STOPLINK command searches all MSC physical links and stops communication on all MSC logical links that are named MSCLINK1:

```
nnSTOPLINK MSCLINK1
```

Example output for the STOPLINK command:

```
HWSF3310I LOGICAL LINK MSCLINK1 TERMINATED; MSC=MSC12, M=ISC2
```

Related reference:

UPDATE IMSCON TYPE(LINK) command (Commands)

"IMS Connect DELETE LINK command" on page 535

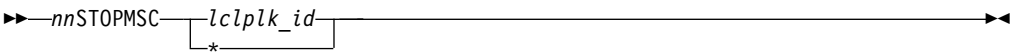
STOPMSC command

Use the WTOR command STOPMSC to stop communications on an MSC physical link that is defined to a local instance of IMS Connect. The STOPMSC command performs similar functions as the z/OS MODIFY command UPDATE MSC STOP(COMM).

Subsections:

- "Syntax"
- "Keywords"
- "Usage notes" on page 477
- "Equivalent IMS Connect type-2 and z/OS commands" on page 477
- "Examples" on page 477

Syntax



Keywords

The following keywords are valid for the STOPMSC command:

nn The two-digit number of the z/OS system console outstanding display.

lclplk_id
The ID of the physical link on which communications are stopped, as specified on the LCLPLKID parameter of the MSC statement that defines the physical link to IMS Connect.

Usage notes

When the STOPMSC command is issued, IMS Connect:

- Stops communication on the specified MSC physical link, including communications on all the MSC logical links that are assigned to the physical link
- Informs IMS that communication has stopped on the physical link so that IMS can also terminate the physical link and any logical links that are assigned to the physical link
- Changes the status of the MSC physical link to NOT ACTIVE
- For TCP/IP generic resources, clears affinity of a physical link to the IMS system
- Issues message HWSF3305I

Use the VIEWMSC command or the QUERY MSC command to display information about MSC physical links that are defined to IMS Connect.

Equivalent IMS Connect type-2 and z/OS commands

The following table lists IMS Connect type-2 commands and IMS Connect z/OS commands that perform similar functions as the STOPMSC command.

STOPMSC command	Equivalent IMS Connect type-2 command	Equivalent IMS Connect z/OS command
STOPMSC *lclPlkid*	UPDATE IMSCON TYPE(MSC) NAME(*lclPlkid*) STOP(COMM)	UPDATE MSC NAME(*lclPlkid*) STOP(COMM)

Examples

The following example of the STOPMSC command stops the MSC physical link MSC12:

```
nnSTOPMSC MSC12
```

Example output for the STOPMSC command:

```
HWSF3305I COMMUNICATIONS ON MSC PHYSICAL LINK MSC12 STOPPED; M=DSCM
```

Related reference:

"STARTMSC command" on page 464

"IMS Connect UPDATE MSC command" on page 597

UPDATE IMSCON TYPE(MSC) command (Commands)

STOPOD command

Use the STOPOD command to immediately terminate communication between the IMS Connect and an IMS Open Database Manager (ODBM).

Subsections:

- "Syntax" on page 478
- "Keywords" on page 478
- "Usage notes" on page 478
- "Equivalent IMS Connect type-2 and z/OS commands" on page 478
- "Examples" on page 478

Syntax

Keywords

The following keywords are valid for the STOPOD command:

nn 2-digit number of z/OS system console outstanding display.

odbm_name
Specifies the name of the ODBM. Wildcards (* and %) are supported in the name.

Usage notes

Use the STOPOD command for any type of error situation that requires immediate termination of communication with an ODBM. Work currently in progress for an ODBM is ended and communications with that ODBM and its threads are terminated.

Use the STARTOD command to open communication with the ODBM at a later time.

After the STOPOD command, the ODBM status will be UNREGISTERED.

Equivalent IMS Connect type-2 and z/OS commands

The following table lists IMS Connect type-2 commands and IMS Connect z/OS commands that perform similar functions as the STOPOD command.

STOPOD command	Equivalent IMS Connect type-2 command	Equivalent IMS Connect z/OS command
STOPOD *odbm_name*	UDPATE IMSCON TYPE(ODBM) NAME(*odbm_name*) STOP(COMM)	UPDATE ODBM NAME(*odbmName*) STOP(COMM)

Examples

To stop communication to ODBM named ODBM1OD:

```
nnSTOPOD ODBM1OD
```

Related reference:

UPDATE IMSCON TYPE(ODBM) command (Commands)

"IMS Connect UPDATE ODBM command" on page 599

STOPPORT command

Use the STOPPORT command to immediately terminate listening on a TCP/IP port.

Subsections:

- "Syntax" on page 479
- "Keywords" on page 479

- "Usage notes"
- "Equivalent IMS Connect type-2 and z/OS commands"
- "Examples"

Syntax

Keywords

The following keywords are valid for the STOPPORT command:

nn 2-digit number of z/OS system console outstanding display.

portid
Identifies the number of the port on which listening is to stop. This port number must match a port number that is defined in either the PORTID or the CICSPORT substatement of the TCPIP configuration statement in the HWSCFG*xx* configuration member. For the local option port, specify a *portid* value of LOCAL.

Usage notes

Work currently in progress is allowed to continue for existing clients. Only the listening for new request messages on the port is terminated immediately. When existing work has completed, the port is no longer active.

Use the VIEWPORT command to display the state of the port and any clients using that port.

Equivalent IMS Connect type-2 and z/OS commands

The following table lists IMS Connect type-2 commands and IMS Connect z/OS commands that perform similar functions as the STOPPORT command.

STOPPORT command	Equivalent IMS Connect type-2 command	Equivalent IMS Connect z/OS command
STOPPORT *port_id*	UDPATE IMSCON TYPE(PORT) NAME(*port_name*) STOP(COMM)	UPDATE PORT NAME(*port_name*) STOP(COMM)

Examples

To stop listening on port 9999:

```
nnSTOPPORT 9999
```

Where *nn* is the 2-digit number of z/OS system console outstanding display.

Related reference:

UPDATE IMSCON TYPE(PORT) command (Commands)

"IMS Connect UPDATE PORT command" on page 601

STOPRMT command

Use the WTOR command STOPRMT in a local instance of IMS Connect to stop communications with a remote instance of IMS Connect. The STOPRMT command performs a similar function to the z/OS MODIFY command UPDATE RMTIMSCON STOP(COMM).

Subsections:

- "Syntax"
- "Keywords"
- "Usage notes"
- "Equivalent IMS Connect type-2 and z/OS commands" on page 481
- "Examples" on page 481

Syntax

Keywords

The following keywords are valid for the STOPRMT command:

nn The two-digit number of the z/OS system console outstanding display.

rmtimscon_id
The name of the remote IMS Connect connection to stop, as specified on the ID parameter of the RMTIMSCON statement that defines the connection to the local IMS Connect instance.

Usage notes

When the STOPRMT command is issued, IMS Connect:

- Stops all communication with a remote IMS Connect instance on the connection specified in the command.
- Informs IMS that the connection to the remote IMS Connect instance has been stopped so that IMS can shut down any affected MSC logical links.
- Terminates existing socket connections and prevents new socket connections until either a STARTRMT command or an UPDATE RMTIMSCON START(COMM) command is issued.
- Changes the status of the connection to `STOPPED`.
- Issues message HWST3505I.

After the STOPRMT command is issued, if IMS Connect receives a message for the connection in the `STOPPED` state, IMS Connect returns a NAK response to OTMA and issues message HWST3575W. OTMA retains the message at the front of the queue.

Use the VIEWRMT command to display the current status of the remote IMS Connect connections that are defined to the local IMS Connect instance.

Equivalent IMS Connect type-2 and z/OS commands

The following table lists IMS Connect type-2 commands and IMS Connect z/OS commands that perform similar functions as the STOPRMT command.

STOPRMT command	Equivalent IMS Connect type-2 command	Equivalent IMS Connect z/OS command
STOPRMT *rmtimsconName*	UPDATE IMSCON TYPE(RMTIMSCON) NAME(*rmtimscon_name*) STOP(COMM)	UPDATE RMTIMSCON NAME(*rmtimsconName*) STOP(COMM)

Examples

In the following example, the STOPRMT command stops communication on the connection CONNECT2:

```
nnSTOPRMT CONNECT2
```

STOPRMT command output:

```
HWST3505I COMMUNICATIONS WITH REMOTE IMS CONNECT CONNECT2 STOPPED; M=TSCH
```

Related reference:

"STARTRMT command" on page 467

"IMS Connect UPDATE RMTIMSCON command" on page 603

UPDATE IMSCON TYPE(RMTIMSCON) command (Commands)

STOPSCLN command

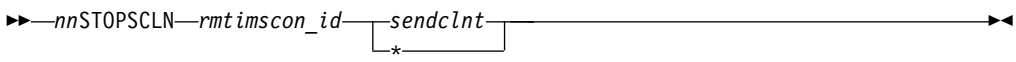
For connections that are used to send OTMA messages to a remote IMS system, use the IMS Connect WTOR command STOPSCLN to terminate send sockets on the connection to the remote IMS Connect instance.

Do not use the STOPSCLN command for connections that are used for ISC parallel sessions or MSC links, unless the send sockets failed to clean up properly after the connection was terminated.

Subsections:

- "Syntax"
- "Keywords"
- "Usage notes" on page 482
- "Equivalent IMS Connect type-2 and z/OS commands" on page 482
- "Examples" on page 483

Syntax



Keywords

The following keywords are valid for the STOPSCLN command:

nn The two-digit number of the z/OS system console outstanding display.

rmtimscon_id
The ID of the remote IMS Connect connection that the send socket is on, as specified on the ID parameter of the RMTIMSCON statement.

sendclnt
The ID of the send socket to terminate. An asterisk (*) can be used as a wildcard character.

Usage notes

Use the STOPSCLN command to terminate send sockets on:

- An ISC link to a CICS subsystem.
- An IMS Connect-to-IMS Connect connection that is used to send messages to a remote IMS system.

To identify the IDs of the send sockets that you need to terminate, use any one of the following IMS Connect commands:

- In the IMS type-2 command format, QUERY IMSCON TYPE(SENDCLNT)
- For OTMA connections to a remote IMS system:
  - In the WTOR command format, VIEWRMT
  - In the z/OS MODIFY command format, QUERY RMTIMSCON

For ISC parallel sessions and MSC links, the STOPSCLN command should be used only if the send client was not properly cleaned up during link termination. Otherwise, do not use the STOPSCLN command to terminate sockets that are used by ISC or MSC because it breaks the sessions or links.

If you need to stop ISC parallel sessions that are still active in IMS, stop them by issuing the IMS command /QUIESCE NODE, which automatically terminates the associated send and receive sockets in IMS Connect. An alternative method of terminating send sockets that are used for ISC parallel sessions is to use the UPDATE IMSCON TYPE(ISCUSER) type-2 IMS command, which terminates both the send and receive sockets that are used by the parallel session.

If you need to stop MSC links that are still active in IMS, stop them by issuing the IMS command /PSTOP LINK, which automatically terminates the associated send and receive sockets in IMS Connect. An alternative method of terminating send sockets that are used for MSC messages is to use the IMS Connect WTOR command STOPLINK or the IMS Connect z/OS modify command DELETE LINK. These commands terminate the send sockets and receive sockets used by the MSC link. The IMS type-2 command equivalent is UPDATE IMSCON TYPE(LINK).

If a send socket connection is in RECV state when the STOPSCLN command is issued, IMS Connect issues a NAK to OTMA and then terminates the send socket connection. OTMA reroutes the message to the dead letter queue, HWS$DLQ.

Equivalent IMS Connect type-2 and z/OS commands

The following table lists IMS Connect type-2 commands and IMS Connect z/OS commands that perform similar functions as the STOPSCLN command.

STOPSCLN command	Equivalent IMS Connect type-2 command	Equivalent IMS Connect z/OS command
STOPSCLN *rmtimscon sendclient*	UPDATE IMSCON TYPE(SENDCLNT) NAME(*sendclient_name*) RMTIMSCON(*rmtimscon_name*) STOP(COMM)	DELETE RMTIMSCON NAME(*rmtimsconname*) SENDCLNT(*clientid*)

Examples

In the following example, the STOPSCLN command terminates send socket OTM11111 on connection CONNECT2:

```
nnSTOPSCLN CONNECT2 OTM11111
```

Example output for the STOPSCLN command:

```
HWST3525I THE SEND CLIENT OTM11111 WAS TERMINATED FOR REMOTE IMS CONNECT CONNECT2; M=TCVC
```

Related reference:

"VIEWRMT command" on page 526

"IMS Connect QUERY RMTIMSCON command" on page 581

"IMS Connect DELETE SENDCLNT command" on page 537

UPDATE IMSCON TYPE(SENDCLNT) command (Commands)

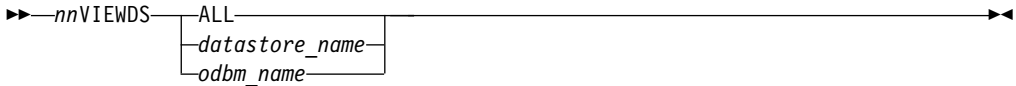
VIEWDS command

Use the VIEWDS command to display the current status and activity of one or all data stores and ODBM instances that are known to IMS Connect.

Subsections:

- "Syntax"
- "Keywords"
- "Usage notes" on page 484
- "Equivalent IMS Connect type-2 and z/OS commands" on page 484
- "Output fields" on page 484
- "Examples" on page 487

Syntax



Keywords

The following keywords are valid for the VIEWDS command:

ALL
Displays information about all data stores and ODBM instances that are known to IMS Connect.

datastore_name
The name of the data store for which information is to be displayed. This name

must match the ID parameter of a DATASTORE configuration statement of the HWSCFGxx configuration member. Only the information for this data store is displayed.

nn The 2-digit number of z/OS system console outstanding display.

odbm_name
The name of the ODBM instance for which information is to be displayed. This name must match the name of an ODBM instance as defined in the NAME parameter of the ODBM statement in the CSLDCxxx PROCLIB member. Only the information for this ODBM instance is displayed.

Usage notes

Use the VIEWDS command to display the current activity of a data store or an ODBM instance.

To view information about only a single data store or ODBM instance, specify the name of a specific data store or ODBM instance. To view information about all data stores and ODBM instances associated with an instance of IMS Connect, specify the ALL keyword.

The information displayed by *nn*`VIEWDS ALL` is the same as the information returned for all data stores by the VIEWHWS command.

Equivalent IMS Connect type-2 and z/OS commands

The following table lists IMS Connect type-2 commands and IMS Connect z/OS commands that perform similar functions as the VIEWDS command.

VIEWDS command	Equivalent IMS Connect type-2 command	Equivalent IMS Connect z/OS command
VIEWDS ALL	QUERY IMSCON TYPE(DATASTORE) NAME(*) SHOW(ALL|*show_parm*)	QUERY DATASTORE NAME(*) SHOW(ALL)
VIEWDS *datastore_name*	QUERY IMSCON TYPE(DATASTORE) NAME(*datastore_name*) SHOW(ALL|*show_parm*)	QUERY DATASTORE NAME(*datastore_name*)

Output fields

The following list shows the output fields of the VIEWDS command.

DATASTORE=
Name of the data store, as defined in the ID substatement of the DATASTORE configuration statement in the `HWSCFGxx` configuration member, or `No active Datastores`. The data store is an IMS system or, more precisely, a connection to an IMS system.

STATUS
State of the data store: ACTIVE, NOT ACTIVE or DISCONNECT.

If the data store goes down, IMS Connect is notified (from IMS OTMA through z/OS cross-system coupling facility (XCF)) of the status of the data store. When the data store is brought back up and restarted, IMS Connect is notified and automatically reconnects to the data store.

GROUP
XCF group name for the group to which the IMS Connect and IMS OTMA belong.

MEMBER
IMS Connect member name in the XCF group listed.

TARGET MEMBER
IMS OTMA member name in the XCF group listed.

STATE
Indicates the state of the OTMA server. The STATE value can have one of the following values:

AVAIL
Indicates that the OTMA server is available.

WARN
Indicates that the OTMA server has one or more resources in warning state.

SEVERE
Indicates that the OTMA server is experiencing some severe resource issues.

N/A Indicates that OTMA has not reported status for this data store.

DEFAULT REROUTE NAME
The name of an alternative destination specified in a client reroute request.

RACF APPL NAME
The RACF APPL name for PassTicket and data store access control.

MULTIRTP
Shows the current specification of the MULTIRTP parameter in the DATASTORE statement for the currently displayed DATASTORE connection. Possible values are:

Y The client bid requests for this DATASTORE connection request a TPIPE that supports multiple active resume TPIPE requests.

N The client bid requests for this DATASTORE connection request a TPIPE that supports only a single active resume TPIPE request.

The client bid requests for this DATASTORE connection do not contain a MULTIRTP specification. OTMA determines whether the TPIPEs for this DATASTORE connection support multiple active resume TPIPE requests.

b A MULTIRTP value is not specified in the DATASTORE statement. TPIPE support for multiple resume tpipe requests is determined by either the MULTIRTP specification in the HWS statement or in the OTMA client descriptor.

CASCADE
Shows the specification in the data store definition for support for cascading global RRS transactions to IMS TM data stores that are located on different LPARs than IMS Connect.

This specification was set either by the CASCADE= parameter of the DATASTORE= statement in the HWSCFGxx member of the IMS PROCLIB data set or by the SET(CASCADE() keyword on type-2 command CREATE IMSCON TYPE(DATASTORE) or UPDATE IMSCON TYPE(DATASTORE).

The value shown here overrides the IMS Connect default value for CASCADE.

One of the following values is displayed:

blank A blank value indicates that cascaded transaction support was not specified for this data store. Support for this data store is determined by the IMS Connect default specification.

Y Indicates that when IMS Connect and the IMS data store are on different z/OS images (LPARs), this data store connection supports cascading global IMS TM transactions.

N Indicates that by default, this data store connection does not support cascading global IMS TM transactions.

OTMA ACEE AGING VALUE
The current OTMA accessor environment element (ACEE) aging value specified in the IMS Connect DATASTORE configuration statement. The OTMA ACEE aging value determines how frequently OTMA refreshes the security definitions that are cached in an ACEE for IMS Connect.

OTMA ACK TIMEOUT VALUE
The OTMA ACK timeout value in the IMS Connect configuration member.

OTMA MAX INPUT MESSAGE
The OTMA input message flood control value in the IMS Connect configuration file.

SUPER MEMBER NAME
The name of the OTMA super member to which this data store belongs.

ODBM=
Identifies an instance of Open Database Manager (ODBM) present in the IMSplex.

STATUS
Shows whether this instance of IMS Connect is registered with the identified instance of ODBM. STATUS can have one of the following values:

- `REGISTERED` - IMS Connect is registered with the identified instance of ODBM.
- `NOT ACTIVE` - The identified instance of ODBM is not available.

ODBMRRS
Indicates the RRS specification in an ODBM. When IMS Connect registers with ODBM, ODBM provides IMS Connect with its RRS setting in addition to the aliases.

ALIAS
The alias name of an IMS data store defined to the instance of ODBM identified in the preceding ODBM output field.

STATUS
Shows whether the IMS system associated with the preceding alias name is active or inactive. The status can have one of the following values:

- `ACTIVE` - IMS Connect can route messages to the identified alias.
- `NOT ACTIVE` - IMS Connect cannot route messages to the identified alias.

Examples

In the following examples, *nn* is the 2-digit number of z/OS system console outstanding display.

Example 1 for VIEWDS command

To view the information for a single data store, IMS1:

```
nnVIEWDS IMS1
```

This command produces the following output:

```
HWSC0001I  DATASTORE=IMS1     STATUS=ACTIVE
HWSC0001I    GROUP=XCFGRP1  MEMBER=HWS1
HWSC0001I    TARGET MEMBER=IMS1           STATE=AVAIL
HWSC0001I    DEFAULT REROUTE NAME=CFG$DEF
HWSC0001I    RACF APPL NAME=APPLID1       MULTIRTP=Y   CASCADE=Y
HWSC0001I    OTMA ACEE AGING VALUE=2147483647
HWSC0001I    OTMA ACK TIMEOUT VALUE=120
HWSC0001I    OTMA MAX INPUT MESSAGE=5000
HWSC0001I    SUPER MEMBER NAME=SM01  CM0 ACK TOQ=
```

Example 2 for VIEWDS command

To view the information for an ODBM instance ODBMAOD:

```
nnVIEWDS ODBMAOD
```

This command produces the following output:

```
HWSC0001I  ODBM=ODBMAOD  STATUS=REGISTERED   ODBMRRS=Y
HWSC0001I    ALIAS=IMS1  STATUS=ACTIVE, TRACE
HWSC0001I    ALIAS=IMS2  STATUS=NOT ACTIVE(ODBM)
```

Example 3 for VIEWDS command

To view the information for all data stores defined to IMS Connect:

```
nnVIEWDS ALL
```

This command produces the following output:

```
HWSC0001I  DATASTORE=IMS1     STATUS=ACTIVE
HWSC0001I    GROUP=XCFGRP1  MEMBER=HWS1
HWSC0001I    TARGET MEMBER=IMS1           STATE=AVAIL
HWSC0001I    DEFAULT REROUTE NAME=CFG$DEF
HWSC0001I    RACF APPL NAME=APPLID1       MULTIRTP=Y   CASCADE=Y
HWSC0001I    OTMA ACEE AGING VALUE=2147483647
HWSC0001I    OTMA ACK TIMEOUT VALUE=120
HWSC0001I    OTMA MAX INPUT MESSAGE=5000
HWSC0001I    SUPER MEMBER NAME=SM01  CM0 ACK TOQ=
HWSC0001I  ODBM=ODBMAOD  STATUS=REGISTERED   ODBMRRS=Y
HWSC0001I    ALIAS=IMS1  STATUS=ACTIVE, TRACE
HWSC0001I    ALIAS=IMS2  STATUS=NOT ACTIVE(IMSCON)
```

```
HWSC0001I  ODBM=ODBMBOD  STATUS=REGISTERED ODBMRRS=N
HWSC0001I    ALIAS=IMS1  STATUS=ACTIVE
HWSC0001I    ALIAS=IMS2  STATUS=ACTIVE
HWSC0001I  ODBM=ODBMCOD  STATUS=NOT ACTIVE
HWSC0001I  ODBM=ODBMDOD  STATUS=NOT ACTIVE
```

Related reference:

QUERY IMSCON TYPE(DATASTORE) command (Commands)

"IMS Connect QUERY DATASTORE command" on page 540

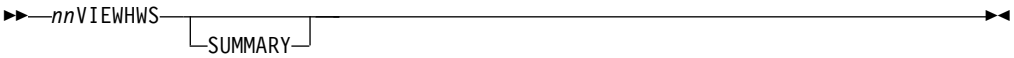
VIEWHWS command

Use the VIEWHWS command to display the current activity of IMS Connect.

Subsections:

- "Syntax"
- "Keywords"
- "Usage notes"
- "Equivalent IMS Connect type-2 and z/OS commands" on page 489
- "Output fields" on page 489
- "Example 1: VIEWHWS command" on page 504

Syntax



Keywords

The following keywords are valid for the VIEWHWS command:

nn The 2-digit number of the z/OS system console outstanding display.

SUMMARY
Excludes detailed client information. This option provides the same HWS and data store information but bypasses listing each individual client for the ports. The port status and client totals are still reported.

Usage notes

You can use the VIEWDS command to display information for data stores only or the VIEWPORT command to display information for ports only.

The VIEWHWS command displays the CONV status if a client is running an IMS conversational transaction. However, if an application sets the InteractionSpec convTypeNew property IMS to true, IMS Connect will not keep track of the conversational status of the client across iterations of the conversational transaction. Therefore, the VIEWHWS command will not display the CONV status unless the command is entered during a conversational iteration.

When IMS Connect connects to an ODBM, the status of the ODBM is REGISTERED. When a STOPOD command is issued to stop the connection to the ODBM, the status of the ODBM is changed to UNREGISTERED. When a REGISTERED ODBM goes down, the status of that ODBM is changed to NOT ACTIVE. If an alias is ACTIVE, then IMS Connect can route DRDA messages to it.

If an alias is NOT ACTIVE, then IMS Connect cannot route DRDA messages to it.

Equivalent IMS Connect type-2 and z/OS commands

The following table lists IMS Connect type-2 commands and IMS Connect z/OS commands that perform similar functions as the VIEWHWS command.

VIEWHWS command	Equivalent IMS Connect type-2 command	Equivalent IMS Connect z/OS command
VIEWHWS	QUERY IMSCON TYPE(CONFIG) SHOW(ALL|*show_parm*)	QUERY MEMBER TYPE(IMSCON) SHOW(ALL)

Output fields

The following list shows the output fields of the VIEWHWS command.

HWS ID=
Name of the IMS Connect, as defined in the ID substatement of the HWS configuration statement in the `HWSCFGxx` configuration member.

RACF Indicates whether RACF support is currently enabled for the instance of IMS Connect identified in the HWS ID= field.

UIDCACHE
Indicates whether the RACF user ID cache is enabled for the instance of IMS Connect identified in the HWS ID= field.

UIDAGE
The length of time in seconds that a RACF user ID is used before IMS Connect refreshes it.

PSWDMC
Indicates whether mixed-case password support is currently enabled for the instance of IMS Connect identified in the HWS ID= field. “R” indicates that the IMS Connect support for mixed-case passwords depends on the mixed-case password specification in RACF.

MAXSOC
The maximum total number of sockets that this instance of IMS Connect can open. The maximum number of physical connections that can be made is the MAXSOC= value less the number of ports, because IMS Connect uses one socket on each port for listening.

TCPIPQ
The maximum number of queued requests that IMS Connect can maintain without assigned sockets.

TIMEOUT
The interval of time IMS Connect waits before disconnecting a client application program after either waiting for a response for the client from IMS or waiting for data from the client after the client opens a socket connection.

NUMSOC
The sum of all active sockets on all defined ports, RMTIMSCON connections, and RMTCICS connections.

WARNSOC
A decimal value between 1 and 50 that indicates the warning level as a percentage of the max sockets limit (MAXSOC= value in the TCPIP configuration statement). When the number of sockets increases to this warning level, IMS Connect issues an HWSS0772W message.

WARNINC
A decimal value between 1 and 50 that indicates the warning incremental percentage. After the warning level (WARNSOC value in the TCPIP configuration statement) is reached, each time the number of sockets increases by the warning incremental percentage, IMS Connect issues an HWSS0772W message again.

RRS Shows if RRS is set to Y or N in the HWS configuration file.

STATUS
State of RRS. The RRS state can be one of the following:

- `ACTIVE` - IMS Connect restart with RRS is complete.
- `NOT ACTIVE` - IMS Connect is not registered with RRS.
- `REGISTERED` - IMS Connect is registered with RRS.

MULTIRTP
Shows the current specification of the MULTIRTP parameter in the HWS statement. Possible values are:

Y MULTIRTP=Y is the default value for all DATASTORE connections from this IMS Connect instance. Unless this value is overridden by the MULTIRTP parameter in the DATASTORE statement, the client bid requests for DATASTORE connections made from this IMS Connect instance will request TPIPEs that support multiple active resume TPIPE requests.

N MULTIRTP=N is the default value for all DATASTORE connections from this IMS Connect instance. Unless this value is overridden by the MULTIRTP parameter in the DATASTORE statement, the client bid requests for DATASTORE connections made from this IMS Connect instance will request TPIPEs that support only a single active resume TPIPE request.

b A MULTIRTP value is not specified in the HWS statement. TPIPE support for multiple resume tpipe requests is determined by either the MULTIRTP specification in the DATASTORE statement or in the OTMA client descriptor.

CASCADE

Shows the IMS Connect default specification for support for cascading global RRS transactions to IMS TM data stores that are located on different LPARs than IMS Connect. This specification was set either by the CASCADE= parameter of the HWS= statement in the HWSCFGxx member of the IMS PROCLIB data set or by the type-2 UPDATE IMSCON TYPE(CONFIG) SET(CASCADE()) command.

The value shown here can be overridden in the definition of an individual data store connection.

One of the following values is displayed:

Y Indicates that when IMS Connect and the IMS data store are on different z/OS images (LPARs), IMS Connect enables cascaded transaction support for all data store connections that do not have a different CASCADE value in their definitions.

N Indicates that by default, cascaded transaction support is disabled for data store connections that do not have a different CASCADE value in their definitions.

VERSION
The version of this instance of IMS Connect.

IP-ADDRESS
The IP address for this instance of IMS Connect. Client application programs use this IP address to connect to IMS Connect.

SUPER MEMBER NAME
1-4 character OTMA super member name.

ADAPTER
Shows whether XML adapter support is enabled or disabled.

MAXCVRT
The maximum number of XML converters that this instance of IMS Connect can load concurrently. The actual number of converters that IMS Connect can load depends on the size of the converters and the IMS Connect region size.

NUMCVRT
The current number of XML converters loaded in this instance of IMS Connect.

MAXLSSSZ
Displays the maximum language structure segment size that is passed to the XML converter.

ODBM AUTO CONNECTION
Indicates whether the instance of IMS Connect identified in the ID field is configured to automatically connect to the instances of ODBM in the IMSplex in which IMS Connect is a member.

ODBM TIMEOUT
The time interval that IMS Connect waits before disconnecting a client application program that uses the Open Database architecture.

ODBM IMSPLEX
The names of IMS Connect and SCI that are used for communication with ODBM within an IMSplex.

MEMBER
The name of IMS Connect that SCI uses to manage communications between IMS Connect and ODBM.

TARGET MEMBER
The name of the SCI that manages the communication between IMS Connect and ODBM.

DATASTORE=
Name of the data store, as defined in the ID substatement of the DATASTORE configuration statement in the `HWSCFGxx` configuration member, or `No active Datastores`. The data store is an IMS system or, more precisely, a connection to an IMS system.

STATUS
State of the data store: ACTIVE, NOT ACTIVE or DISCONNECT.

If the data store goes down, IMS Connect is notified (from IMS OTMA through z/OS cross-system coupling facility (XCF)) of the status of the data store. When the data store is brought back up and restarted, IMS Connect is notified and automatically reconnects to the data store.

GROUP
XCF group name for the group to which the IMS Connect and IMS OTMA belong.

MEMBER
IMS Connect member name in the XCF group listed.

TARGET MEMBER
IMS OTMA member name in the XCF group listed.

STATE
Indicates the state of the OTMA server. The STATE value can have one of the following values:

AVAIL
Indicates that the OTMA server is available.

WARN
Indicates that the OTMA server has one or more resources in warning state.

SEVERE
Indicates that the OTMA server is experiencing some severe resource issues.

N/A Indicates that OTMA has not reported status for this data store.

DEFAULT REROUTE NAME
The name of an alternative destination specified in a client reroute request.

RACF APPL NAME
The RACF APPL name for PassTicket and data store access control.

MULTIRTP
Shows the current specification of the MULTIRTP parameter in the DATASTORE statement for the currently displayed DATASTORE connection. Possible values are:

Y The client bid requests for this DATASTORE connection request a TPIPE that supports multiple active resume TPIPE requests.

N The client bid requests for this DATASTORE connection request a TPIPE that supports only a single active resume TPIPE request.

The client bid requests for this DATASTORE connection do not contain a MULTIRTP specification. OTMA determines whether the TPIPEs for this DATASTORE connection support multiple active resume TPIPE requests.

b A MULTIRTP value is not specified in the DATASTORE statement. TPIPE support for multiple resume tpipe

requests is determined by either the MULTIRTP specification in the HWS statement or in the OTMA client descriptor.

CASCADE
Shows the specification in the data store definition for support for cascading global RRS transactions to IMS TM data stores that are located on different LPARs than IMS Connect.

This specification was set either by the CASCADE= parameter of the DATASTORE= statement in the HWSCFGxx member of the IMS PROCLIB data set or by the SET(CASCADE() keyword on type-2 command CREATE IMSCON TYPE(DATASTORE) or UPDATE IMSCON TYPE(DATASTORE).

The value shown here overrides the IMS Connect default value for CASCADE.

One of the following values is displayed:

blank A blank value indicates that cascaded transaction support was not specified for this data store. Support for this data store is determined by the IMS Connect default specification.

Y Indicates that when IMS Connect and the IMS data store are on different z/OS images (LPARs), this data store connection supports cascading global IMS TM transactions.

N Indicates that by default, this data store connection does not support cascading global IMS TM transactions.

OTMA ACEE AGING VALUE
The current OTMA accessor environment element (ACEE) aging value specified in the IMS Connect DATASTORE configuration statement. The OTMA ACEE aging value determines how frequently OTMA refreshes the security definitions that are cached in an ACEE for IMS Connect.

OTMA ACK TIMEOUT VALUE
The OTMA ACK timeout value in the IMS Connect configuration member.

OTMA MAX INPUT MESSAGE
The OTMA input message flood control value in the IMS Connect configuration file.

SUPER MEMBER NAME
The name of the OTMA super member to which this data store belongs.

IMSPLEX=
Name of the IMSplex as defined in the TMEMBER parameter of the IMSplex statement in the IMS Connect configuration member HWSCFGxx.

STATUS
The state of the IMSplex.

MEMBER
IMS Connect member name as defined in the IMS Connect configuration file IMSplex statement for MEMBER= parameter.

TARGET
IMS Connect target member name as defined in the IMS Connect configuration file IMSplex statement for TMEMBER= parameter.

ODBM=
Identifies an instance of Open Database Manager (ODBM) present in the IMSplex.

STATUS
Shows whether this instance of IMS Connect is registered with the identified instance of ODBM. STATUS can have one of the following values:

- `REGISTERED` - IMS Connect is registered with the identified instance of ODBM.
- `NOT ACTIVE` - The identified instance of ODBM is not available.

ODBMRRS
Indicates the RRS specification in an ODBM. When IMS Connect registers with ODBM, ODBM provides IMS Connect with its RRS setting in addition to the aliases.

ALIAS
The alias name of an IMS data store defined to the instance of ODBM identified in the preceding ODBM output field.

STATUS
Shows whether the IMS system associated with the preceding alias name is active or inactive. The status can have one of the following values:

- `ACTIVE` - IMS Connect can route messages to the identified alias.
- `NOT ACTIVE` - IMS Connect cannot route messages to the identified alias.

MSC= Name of the displayed MSC physical link, as specified on the LCLPLKID parameter of the MSC statement in the IMS Connect configuration member. The following additional output fields describe the status, attributes, and logical links that belong to the physical link identified in the MSC= output field:

STATUS
The state of the MSC physical link displayed in the MSC output field. A physical link can be in the following states:

ACTIVE
The MSC physical link is enabled for communications between IMS Connect and the local IMS system.

NOT ACTIVE
The MSC physical link is disabled for communications between IMS Connect and the local IMS system.

RMTPLKID
The ID of the MSC physical link at the remote IMS site, as defined in the RMTPLKID parameter of the MSC statement in the IMS Connect configuration member.

LCLIMSID
The ID of the local MSC-enabled IMS system, as defined to IMS Connect on the LCLIMS parameter of the MSC statement in the

IMS Connect configuration member. This LCLIMS value must match the IMS ID that the local IMS system registers with SCI in the IMSplex. You can determine the IMS ID that is registered with SCI by issuing the QUERY IMSPLEX command in the local IMSplex.

If the second IMS ID is shown in the output, it identifies the other IMS system in an XRF couple.

RMTIMSID
The name of the remote MSC-enabled IMS system, as defined on the RMTIMS parameter of the MSC statement in the IMS Connect configuration member.

GENIMSID
The generic IMS ID that is being used for TCP/IP generic resources for MSC, as defined in the GENIMSID parameter of the MSC statement in the IMS Connect configuration member.

AFFINITY
The name of the IMS system with which the MSC logical link currently has affinity for TCP/IP generic resources.

IMSPLEX
Name of the IMSplex that manages communications between IMS Connect and the local, MSC-enabled IMS system. This name is the IMSplex name that is specified on the IMSPLEX parameter of MSC statement in the IMS Connect configuration member.

This IMSPLEX name must match the name specified on the IMSPLEX(NAME=) statement of the CSLSIxxx PROCLIB member of the SCI instance that is managing communications between IMS Connect and the IMSplex.

MEMBER
The name by which this IMS Connect is registered in the IMSplex, as defined on the MEMBER parameter in the IMSPLEX substatement of the MSC statement in the IMS Connect configuration member.

This MEMBER name must also be specified on the LCLICON parameter in the MSPLINK macro that defines this MSC physical link in the local IMS system.

TARGET MEMBER
Name of the IMSplex as defined in the TMEMBER subparameter of the IMSPLEX parameter of the MSC statement in the IMS Connect configuration member.

RMTIMSCON
The name of the IMS Connect connection that is used by this MSC physical link, as defined in the RMTIMSCON parameter of the MSC statement in the IMS Connect configuration member.

IP-ADDRESS
The IP address of the remote IMS Connect instance.

PORT The port number used by the remote IMS Connect instance.

HOSTNAME
The host name of the remote IMS Connect instance, if one is available.

LINK The ID of an MSC logical link that is assigned to this physical link.

The values returned for this output field are displayed in column format.

If no logical links are currently assigned to this physical link, this output field is not displayed.

PARTNERID

The ID of the remote partner for this logical link, as specified on the PARTNER parameter in the MSLINK macro that defines this logical link.

The values returned for this output field are displayed in column format.

If no logical links are currently assigned to this physical link, this output field is not displayed.

STATUS
The current state of this logical link. A logical link can have the following state:

ACTIVE
This logical link can send messages to the remote IMS Connect.

The values returned for this output field are displayed in column format.

If no logical links are currently assigned to this physical link, this output field is not displayed.

SENDCLNT
The send socket ID generated by this instance of IMS Connect to identify a send socket opened on this connection. The SENDCLNT ID is used as the client ID when establishing a socket connection with the remote IMS Connect instance.

The send socket IDs generated for sockets on connections that are used for OTMA begin with "OTM." The send socket IDs that are generated for sockets on connections that are used for MSC begin with "MSC."

The values returned for this output field are displayed in column format.

If no logical links are currently assigned to this physical link, this output field is not displayed.

RECVCLNT
The client ID that identifies the receive socket for this logical link. The client ID is generated by the remote IMS Connect instance.

The values returned for this output field are displayed in column format.

If no logical links are currently assigned to this physical link, this output field is not displayed.

ISC= ID of the ISC configuration statement in the IMS Connect configuration member that defines the ISC link to which the following output fields apply.

STATUS
Status of this ISC link. Possible values are:

ACTIVE
This ISC link is enabled for communications between IMS Connect and the local IMS system.

NOT ACTIVE
This ISC link is disabled for communications between IMS Connect and the local IMS system.

NODE
The name of this ISC node as defined to the local IMS.

LCLIMSID
Name of the local IMS as defined in the LCLIMS parameter of the ISC statement in the IMS Connect configuration member.

IMSPLEX=

Name of the IMSplex as defined in the IMSPLEX parameter of ISC statement in the IMS Connect configuration member. If IMSPLEX is not specified in the ISC statement, the name from the first IMSPLEX statement in the configuration is displayed.

MEMBER=
Name of the member as defined in the MEMBER subparameter of the IMSPLEX parameter of the ISC statement in the IMS Connect configuration member. This is the IMS Connect member name registered to the IMSplex.

TARGET MEMBER=
Name of the IMSplex as defined in the TMEMBER subparameter of the IMSPLEX parameter of the ISC statement in the IMS Connect configuration member.

RMTCICS
Name of the remote CICS connection that this ISC link is using. The value shown here is the value specified on the RMTCICS parameter of the ISC statement in the IMS Connect configuration member.

IP-ADDRESS
The IP address of this remote CICS.

PORT The port of this remote CICS.

HOSTNAME
The hostname of this remote CICS. This field is displayed if HOSTNAME parameter is specified in the RMTCICS statement.

CICSAPPL
The value specified on the APPLID= parameter of the DFHSIT macro definition in the remote CICS subsystem.

CICSPORT
The ID of the port that IMS Connect uses to receive messages from CICS on this ISC link. The CICSPORT is defined on the TCPIP configuration statement.

CICSNETID
The network ID of the remote CICS subsystem.

ISCUSER STATUS RECVCLNT SENDCLNT

If one or more ISC parallel sessions (ISCUSERs) are active on this ISC link, the command output contains information about each ISCUSER. The information for multiple ISCUSERs is displayed in a column format under a heading that identifies the information type. The information for each ISCUSER is displayed on its own row.

ISCUSER
Name of the ISC parallel session.

STATUS
Status of this ISCUSER.

SENDCLNT
Name of the client ID that IMS Connect uses to send messages to the remote CICS subsystem for this ISCUSER.

RECVCLNT
Name of the client ID of the remote CICS subsystem that this IMS Connect receives messages from for this ISCUSER.

RMTCICS=
Name of the remote CICS connection as defined in the ID parameter of the RMTCICS configuration statement in the IMS Connect configuration member.

STATUS=
State of the connection to the remote CICS subsystem that is identified in the RMTCICS field. Possible values include:

ACTIVE
This IMS Connect has at least one socket connection to the remote CICS subsystem identified in the RMTCICS field.

NOT ACTIVE
This IMS Connect does not have any socket connections to the remote CICS identified in the RMTCICS field.

STOPPED
Communications between this IMS Connect and the remote CICS subsystem is stopped. Any messages sent to the remote subsystem while the connection is stopped are rejected and an error message is returned to the sender.

IP-ADDRESS
The IP address of the remote CICS.

PORT The port number of the remote CICS that this IMS Connect instance communicates on.

HOSTNAME
The host name of the remote CICS or remote IMS Connect, if available.

NUMSOC
The number of sockets that are currently open.

SENDCLNT ISCNAME STATUS SECOND SENDPORT
If this instance of IMS Connect has one or more active connections to this remote CICS subsystem, the command output includes information about each connection. The information for an active connection is displayed immediately under the heading that describes the value.

The headings are described in the next list of fields.

SENDCLNT
The client ID that this instance of IMS Connect used to connect to the remote CICS.

ISCNAME
Name of the ISC configuration statement in the IMS Connect configuration member.

STATUS
State of this remote IMS Connect connection. Possible values include:

CONN
This connection is in connect state. It can send messages to the remote CICS.

RECV This connection is in receive state. It is waiting to receive a response from the remote CICS.

SECOND
Number of seconds that this connection is in the specified state.

SENDPORT
The port that this instance of IMS Connect used to connect to the remote CICS.

RMTIMSCON=
Identifies a connection between this local IMS Connect instance and a remote instance of IMS Connect. The following additional output fields describe the status, attributes, and sockets that belong to the connection identified in the RMTIMSCON output field:

STATUS
The state of the connection displayed in the RMTIMSCON output field. A connection can be in the following states:

ACTIVE
This connection has at least one send socket open to a remote IMS Connect instance.

NOT ACTIVE
This connection does not have any send sockets open to a remote IMS Connect instance.

RETRY CONN
IMS Connect is attempting to automatically reconnect to the remote IMS Connect instance after a previous attempt to connect failed. If the current reconnection attempt fails, IMS Connect attempts to reconnect again every 2 minutes.

While a connection has a RETRY CONN status, the number of open sockets fluctuates. IMS Connect opens a socket when actively trying to reconnect. IMS Connect closes the socket during the 2-minute interval between reconnection attempts.

STOPPED
A STOPRMT command stopped communication on this connection. Any messages sent to this connection are rejected and an error is sent back to the sender.

IP-ADDRESS
The IP address of the IMS Connect instance at the remote end of this connection.

PORT The port number used by the remote IMS Connect instance for this connection.

HOSTNAME
The host name of the remote IMS Connect instance, if available.

AUTOCONN
Indicates whether IMS Connect automatically establishes send sockets for OTMA communications on this connection during IMS Connect startup. Possible values are:

Y Send sockets are automatically established on this connection when IMS Connect starts up. The number of sockets that IMS Connect establishes during startup is displayed in the RESVSOC output field

N No sockets are established on this connection during IMS Connect startup. For OTMA messages, send sockets are established upon receipt of a message from OTMA for delivery to the remote IMS system.

PERSISTENT
Indicates whether the sockets on this connection are persistent.

Y The sockets established on this connection are persistent.

N The sockets established on this connection are not persistent. After messages are sent on this connection, the sockets that they are sent on are closed.

IDLETO
The time interval that open sockets on this connection can remain idle before they are closed by IMS Connect.

RESVSOC
The number of send sockets that are reserved for use by this connection.

NUMSOC
The number of sockets that are currently open on this connection.

SENDCLNT
The randomly generated send socket ID. The local instance of IMS Connect uses the ID to uniquely identify the socket. The send socket ID is used as the client ID when establishing a socket connection with the remote IMS Connect instance.

The send socket IDs that are used for OTMA begin with "OTM." The send socket IDs that are used for MSC begin with "MSC."

The values returned for this output field are displayed in tabular format.

If no sockets are currently open on this connection, this output field is not displayed.

LCLPLKID
The ID of a local MSC physical link that is using this connection. This value is specified on the LCLPLKID parameter of the MSC statement in the IMS Connect configuration member. The LCLPLKID output field is displayed only for connections that are used for MSC communications.

The values returned for this output field are displayed in tabular format.

If no sockets are currently open on this connection, this output field is not displayed.

USERID
The user ID provided by the local IMS system for transaction authorization by the remote IMS system. The USERID output field is displayed only for connections that are used for OTMA communications.

If no sockets are currently open on this connection, this output field is not displayed.

The values returned for this output field are displayed in tabular format.

STATUS
The current state of this socket connection. A socket connection can have the following states:

CONN
This socket connection can send messages to the remote IMS Connect.

RECV This socket connection is waiting to receive a response from the remote IMS Connect.

The values returned for this output field are displayed in tabular format.

If no sockets are currently open on this connection, this output field is not displayed.

SECOND
The number of seconds that this connection has been in the state shown on the same row under the STATUS output field heading.

The values returned for this output field are displayed in tabular format.

If no sockets are currently open on this connection, this output field is not displayed.

SENDPORT
The local port used for this connection.

The values returned for this output field are displayed in tabular format.

If no sockets are currently open on this connection, this output field is not displayed.

PORT=

Identifies the port or ports to which the following output fields and statuses apply. The port numbers are defined in the IMS Connect configuration member in the IMS.PROCLIB data set.

You can determine the type of a port by the port-type indicator that is appended to the end of the port number. IMS Connect uses the following port-type indicators:

No type indicator
Indicates a port that can be used for IMS TM OTMA clients and IMS-to-IMS MSC or OTMA communications. Ports that do not have a type indicator are defined on either the PORT keyword or the PORTID keyword of the TCPIP statement in the IMS Connect configuration member.

C Indicates an IBM CICS Transaction Server for z/OS port for ISC communications that is defined to IMS Connect on the CICSPORT parameter of the TCPIP and ISC statements. CICS ports are used exclusively by CICS subsystems that access IMS TM via an ISC TCP/IP link.

D Indicates a Distributed Relational Database Architecture™ (DRDA) port that is defined to IMS Connect on the DRDAPORT parameter. DRDA ports are used exclusively by client application programs that access IMS DB through the Open Database Manager (ODBM).

S Indicates an SSL port that is defined to IMS Connect on the SSLPORT parameter. SSL ports are used only when IMS Connect support for SSL is enabled.

The following output fields apply to the port identified in the preceding PORT output field.

STATUS
State of the port, whether `ACTIVE` or `INACTIVE`.

KEEPAV
The amount of time a connection remains idle before the z/OS TCP/IP layer sends a packet to maintain the connection.

NUMSOC
The number of sockets used on each port.

TIMEOUT
The amount of time IMS Connect waits before terminating a client connection if no messages are received from the client.

CLIENTID ISCNAME STATUS SECOND CLNTPORT IP-ADDRESS
or
CLIENTID LCLPLKID STATUS SECOND CLNTPORT IP-ADDRESS
or
CLIENTID USERID TRANCODE DATASTORE STATUS SECOND CLNTPORT IP-ADDRESS APSB-TOKEN
The ID and information for an active client on a receive socket.

The information for each active client is displayed on a single row. Clients are grouped together by client type under a row of headings that identify the information that is returned for that client type.

For ISC clients, the following information is displayed: `CLIENTID ISCNAME STATUS SECOND CLNTPORT IP-ADDRESS`.

For MSC clients, the following information is displayed: `CLIENTID LCLPLKID STATUS SECOND CLNTPORT IP-ADDRESS`.

For OTMA and ODBM clients, the following information is displayed: `CLIENTID USERID TRANCODE DATASTORE STATUS SECOND CLNTPORT IP-ADDRESS APSB-TOKEN`.

The following example shows the information returned for three OTMA clients:

```
HWSC0001I  PORT=9999     STATUS=ACTIVE     KEEPAV=0 NUMSOC=4 EDIT=         TIMEOUT=0
HWSC0001I    CLIENTID USERID   TRANCODE DATASTORE STATUS    SECOND  CLNTPORT IP-ADDRESS       APSB-TOKEN
HWSC0001I    CLIENT63 USRT003  apol12   IMSY      RECV WFCM     55      4118 192.0.2.0
HWSC0001I    CLIENT62 USRT002           IMS1      CONN         210      4115 192.0.2.0
HWSC0001I    CLIENT61 USRT001  apol12   IMSN      RECV          13      4112 192.0.2.0
HWSC0001I    TOTAL CLIENTS=3  RECV=2 CONN=1 XMIT=0 OTHER=0
```

If a port has no active clients on it, these output fields are not displayed. Instead, the command output displays `NO ACTIVE CLIENTS` after the row that identifies the port.

The information returned for active clients includes, in alphabetical order:

APSB-TOKEN
Open Database Manager (ODBM) APSB token.

CLIENTID
The name of the client.

For connections from other IMS Connect instances, the other IMS Connect instance generates this value.

CLNTPORT
A random number that TCP/IP generates to represent a connection from a client.

DATASTORE
The data store associated with this client.

The IP address of the client for this connection.

If IPV6 is enabled, the IP address format consists of eight hexadecimal numbers divided by colons. If IPV6 is not enabled, the IP address format of IPV4 is used. The following example is for an IPV6 IP address that is displayed in IPV6 format:

```
FEDC:ABCD:2222:3333:FEDC:DB55:6666:3322
```

The following example is for an IPV4 IP address that is displayed in IPV6 format:

```
0:0:0:0:0:FFFF:945:33FF
```

ISCNAME
The name of the ISC statement in the IMS Connect configuration member.

LCLPLKID
The MSC physical link in the local IMS system to which IMS Connect routes the messages from the client. This field exists only for MSC communications.

SECOND
Number of seconds that the client has been in the specified status.

STATUS
State of the thread of the client. The client thread state can be one of the following values:

RECV In a receive state waiting for input from client.

READ In a partial read state. IMS Connect is reading a message from the client but has not yet received the entire length of the message.

CONN
Waiting for output from IMS.

XMIT Sending data to client.

CONV
In a conversational state.

WFCM
Waiting for confirmation (ACK, NAK, or DEALLOCATE) from client.

TRANCODE
The transaction code submitted by the client.

USERID
The USERID name passed to IMS Connect.

Example 1: VIEWHWS command

In the following example, *nn* is the 2-digit number of the z/OS system console outstanding display.

To view the information for an instance of IMS Connect:

```
nnVIEWHWS
```

This command produces the following output:

```
HWSC0001I HWS ID=HWS1     RACF=N  PSWDMC=N
HWSC0001I    UIDCACHE=Y   UIDAGE=2147483647
HWSC0001I    MAXSOC=2000  TIMEOUT=5000 TCPIPQ=50
HWSC0001I    NUMSOC=7     WARNSOC=80%  WARNINC=5%
HWSC0001I    RRS=Y  STATUS=REGISTERED  MULTIRTP=  CASCADE=N
HWSC0001I    VERSION=V14 IP-ADDRESS=127.0.0.1
HWSC0001I    SUPER MEMBER NAME=SM01  CM0 ACK TOQ=

HWSC0001I   ADAPTER=N    MAXCVRT=100  NUMCVRT=0
HWSC0001I    MAXLSSSZ=32767
HWSC0001I  ODBM AUTO CONNECTION=Y
HWSC0001I  ODBM TIMEOUT=18000
HWSC0001I  ODBM IMSPLEX MEMBER=IMSPLEX1 TARGET MEMBER=PLEX1
HWSC0001I  DATASTORE=IMS1     STATUS=ACTIVE
HWSC0001I    GROUP=XCFGRP1  MEMBER=HWS1
HWSC0001I    TARGET MEMBER=IMS1           STATE=AVAIL
HWSC0001I    DEFAULT REROUTE NAME=CFG$DEF
HWSC0001I    RACF APPL NAME=APPLID1       MULTIRTP=  CASCADE=Y
HWSC0001I    OTMA ACEE AGING VALUE=2147483647
HWSC0001I    OTMA ACK TIMEOUT VALUE=120
HWSC0001I    OTMA MAX INPUT MESSAGE=5000
HWSC0001I    SUPER MEMBER NAME=SM01  CM0 ACK TOQ=
HWSC0001I      STATUS=ACTIVE
HWSC0001I    MEMBER=IMSPLEX1 TARGET=PLEX1
HWSC0001I  ODBM=ODBMAOD  STATUS=REGISTERED   ODBMRRS=Y
```

```
HWSC0001I    ALIAS=IMS1  STATUS=ACTIVE, TRACE
HWSC0001I    ALIAS=IMS2  STATUS=NOT ACTIVE
HWSC0001I  ODBM=ODBMBOD  STATUS=REGISTERED ODBMRRS=N
HWSC0001I    ALIAS=IMS1  STATUS=ACTIVE
HWSC0001I    ALIAS=IMS2  STATUS=ACTIVE
HWSC0001I  ODBM=ODBMCOD  STATUS=NOT ACTIVE
HWSC0001I  ODBM=ODBMDOD  STATUS=NOT ACTIVE
HWSC0001I  PORT=9999     STATUS=ACTIVE     KEEPAV=0 NUMSOC=4 EDIT=         TIMEOUT=0
HWSC0001I    CLIENTID USERID   TRANCODE DATASTORE STATUS    SECOND  CLNTPORT IP-ADDRESS       APSB-TOKEN
HWSC0001I    CLIENT63 USRT003  apol12   IMSY      RECV WFCM     55      4118 192.0.2.0
HWSC0001I    CLIENT62 USRT002           IMS1      CONN         210      4115 192.0.2.0
HWSC0001I    CLIENT61 USRT001  apol12   IMSN      RECV          13      4112 192.0.2.0
HWSC0001I    TOTAL CLIENTS=3  RECV=2 CONN=1 XMIT=0 OTHER=0
HWSC0001I  PORT=LOCAL    STATUS=ACTIVE     KEEPAV=0 NUMSOC=0 EDIT=         TIMEOUT=0
HWSC0001I    NO ACTIVE CLIENTS
HWSC0001I  PORT=9998S    STATUS=NOT ACTIVE KEEPAV=0 NUMSOC=1 EDIT=         TIMEOUT=0
HWSC0001I  PORT=1111D    STATUS=ACTIVE     KEEPAV=5 NUMSOC=1 EDIT=         TIMEOUT=6000
HWSC0001I    NO ACTIVE CLIENTS
HWSC0001I  PORT=2222D    STATUS=ACTIVE     KEEPAV=0 NUMSOC=1 EDIT=         TIMEOUT=100
HWSC0001I    NO ACTIVE CLIENTS
```

Example 2: SUMMARY

In the following example, *nn* is the 2-digit number of the z/OS system console outstanding display.

To view the summary information for an instance of IMS Connect:

```
nnVIEWHWS SUMMARY
```

This command produces the same output as the VIEWHWS command without the SUMMARY keyword except for the PORT= sections, which give only the total counts for the ports, eliminating individual client displays.

```
HWSC0001I HWS ID=HWS1     RACF=N  PSWDMC=N
HWSC0001I    UIDCACHE=Y   UIDAGE=2147483647
HWSC0001I    MAXSOC=50    TIMEOUT=5000 TCPIPQ=50
HWSC0001I    NUMSOC=7     WARNSOC=80%  WARNINC=5%
HWSC0001I    RRS=Y   STATUS=REGISTERED MULTIRTP=  CASCADE=Y
HWSC0001I    VERSION=V14 IP-ADDRESS=127.0.0.1
HWSC0001I    SUPER MEMBER NAME=SM01  CM0 ACK TOQ=

HWSC0001I   ADAPTER=N    MAXCVRT=100  NUMCVRT=0
HWSC0001I    MAXLSSSZ=32767
HWSC0001I  ODBM AUTO CONNECTION=Y
HWSC0001I  ODBM TIMEOUT=18000
HWSC0001I  ODBM IMSPLEX MEMBER=IMSPLEX1 TARGET MEMBER=PLEX1
HWSC0001I  DATASTORE=IMS1     STATUS=ACTIVE
HWSC0001I    GROUP=XCFGRP1  MEMBER=HWS1
HWSC0001I    TARGET MEMBER=IMS1           STATE=AVAIL
HWSC0001I    DEFAULT REROUTE NAME=CFG$DEF
HWSC0001I    RACF APPL NAME=APPLID1       MULTIRTP=  CASCADE=
HWSC0001I    OTMA ACEE AGING VALUE=2147483647
HWSC0001I    OTMA ACK TIMEOUT VALUE=120
HWSC0001I    OTMA MAX INPUT MESSAGE=5000
HWSC0001I    SUPER MEMBER NAME=SM01  CM0 ACK TOQ=
HWSC0001I  IMSPLEX=PLEX1    STATUS=ACTIVE
HWSC0001I    MEMBER=IMSPLEX1 TARGET=PLEX1
HWSC0001I  ODBM=ODBMAOD  STATUS=REGISTERED   ODBMRRS=Y
HWSC0001I    ALIAS=IMS1  STATUS=ACTIVE, TRACE
HWSC0001I    ALIAS=IMS2  STATUS=NOT ACTIVE(IMSCON,ODBM)
HWSC0001I  ODBM=ODBMBOD  STATUS=REGISTERED ODBMRRS=N
HWSC0001I    ALIAS=IMS1  STATUS=ACTIVE
HWSC0001I    ALIAS=IMS2  STATUS=ACTIVE
HWSC0001I  ODBM=ODBMCOD  STATUS=NOT ACTIVE
HWSC0001I  ODBM=ODBMDOD  STATUS=NOT ACTIVE
HWSC0001I  PORT=9999     STATUS=ACTIVE     KEEPAV=0 NUMSOC=4 EDIT=         TIMEOUT=0
HWSC0001I    TOTAL CLIENTS=3  RECV=2 CONN=1 XMIT=0 OTHER=0
HWSC0001I  PORT=LOCAL    STATUS=ACTIVE     KEEPAV=0 NUMSOC=0 EDIT=         TIMEOUT=0
HWSC0001I    NO ACTIVE CLIENTS
HWSC0001I  PORT=9998S    STATUS=NOT ACTIVE KEEPAV=0 NUMSOC=1 EDIT=         TIMEOUT=0
```

```
HWSC0001I  PORT=1111D    STATUS=ACTIVE     KEEPAV=5 NUMSOC=1 EDIT=         TIMEOUT=6000
HWSC0001I    NO ACTIVE CLIENTS
HWSC0001I  PORT=2222D    STATUS=ACTIVE     KEEPAV=0 NUMSOC=1 EDIT=         TIMEOUT=100
HWSC0001I    NO ACTIVE CLIENTS
```

Example 3: IMS Connect to IMS Connect connections for MSC

If connections between IMS Connect instances are used for MSC, the command output presents different perspectives of the connections, depending on which instance of IMS Connect the command is issued against.

For MSC communications, one instance of IMS Connect has a connection defined to send MSC messages to the other instance of IMS Connect. The other instance of IMS Connect has a connection defined to send the reply messages.

In the following example output, the command was issued for an IMS Connect instance, HWS1, at the IMS installation where the MSC transactions are input to IMS. HWS1 is connected to another IMS Connect instance, HWS2. HWS1 knows HWS2 only by the RMTIMSCON name ICON2. HWS1 has two active client connections for sending MSC transactions to ICON2 for processing, MSCBB435 and MSC84CF7, shown under SENDCLNT in the RMTIMSCON information. HWS1 receives the response messages to the MSC transactions on the client connections MSCC73E0 and MSC0EBB0, shown under CLIENTID in the PORT information.

A DATASTORE statement is defined for this instance of IMS Connect; however, DATASTORE statements are not used for MSC communications and can be omitted if your instance of IMS Connect does not communicate with OTMA.

```
HWSC0001I   HWS ID=HWS1     RACF=N  PSWDMC=R
HWSC0001I      UIDCACHE=N   UIDAGE=2147483647
HWSC0001I      MAXSOC=2000  TIMEOUT=5000 TCPIPQ=50
HWSC0001I      NUMSOC=5     WARNSOC=80%  WARNINC=5%
HWSC0001I      RRS=N   STATUS=REGISTERED MULTIRTP=  CASCADE=N
HWSC0001I      VERSION=V14 IP-ADDRESS=192.0.2.0
HWSC0001I      SUPER MEMBER NAME=      CM0 ACK TOQ=

HWSC0001I     ADAPTER=N   MAXCVRT=100  NUMCVRT=0
HWSC0001I      MAXLSSSZ=32767
HWSC0001I    DATASTORE=IMS1     STATUS=DISCONNECT
HWSC0001I      GROUP=XCFGRP1  MEMBER=HWS1
HWSC0001I      TARGET MEMBER=IMS1             STATE=N/A
HWSC0001I      DEFAULT REROUTE NAME=HWS$DEF
HWSC0001I      RACF APPL NAME=APPLID1         MULTIRTP=  CASCADE=
HWSC0001I      OTMA ACEE AGING VALUE=2147483647
HWSC0001I      OTMA ACK TIMEOUT VALUE=120
HWSC0001I      OTMA MAX INPUT MESSAGE=5000
HWSC0001I      SUPER MEMBER NAME=      CM0 ACK TOQ=
HWSC0001I    IMSPLEX=PLEX1    STATUS=ACTIVE
HWSC0001I      MEMBER=HWS1     TARGET=PLEX1
HWSC0001I    MSC=MSC12    STATUS=ACTIVE
HWSC0001I      RMTPLKID=MSC21
HWSC0001I      LCLIMSID=IMS1      RMTIMSID=IMS2
HWSC0001I      IMSPLEX=PLEX1
HWSC0001I        MEMBER=HWS1              TARGET MEMBER=PLEX1
HWSC0001I      RMTIMSCON=ICON2
HWSC0001I        IP-ADDRESS=127.0.0.1        PORT=5555
HWSC0001I        HOSTNAME=WWW.EXAMPLE.COM
HWSC0001I      LINK       PARTNERID   STATUS       SENDCLNT   RECVCLNT
HWSC0001I      LNK12T01   TA          ACTIVE       MSC84CF7   MSC0EBB0
HWSC0001I      LNK12T02   TB          ACTIVE       MSCBB435   MSCC73E0
HWSC0001I    NO ACTIVE ODBM
HWSC0001I    PORT=9999     STATUS=ACTIVE     KEEPAV=0 NUMSOC=3      EDIT=     TIMEOUT=0
HWSC0001I      CLIENTID LCLPLKID  STATUS      SECOND CLNTPORT IP-ADDRESS
HWSC0001I      MSCC73E0 MSC12     RECV             6 1026     0:0:0:0:0:FFFF:7F00:1
HWSC0001I      MSC0EBB0 MSC12     RECV            56 1025     0:0:0:0:0:FFFF:7F00:1
```

```
HWSC0001I      TOTAL CLIENTS=2  RECV=2 READ=0 CONN=0 XMIT=0 OTHER=0
HWSC0001I    RMTIMSCON=ICON2     STATUS=ACTIVE
HWSC0001I      IP-ADDRESS=127.0.0.1        PORT=5555
HWSC0001I      HOSTNAME=WWW.EXAMPLE.COM
HWSC0001I      AUTOCONN=N  PERSISTENT=Y
HWSC0001I      IDLETO=0
HWSC0001I      RESVSOC=2   NUMSOC=2
HWSC0001I      SENDCLNT LCLPLKID STATUS       SECOND SENDPORT
HWSC0001I      MSCBB435 MSC12    CONN              6 1028
HWSC0001I      MSC84CF7 MSC12    CONN             56 1027
HWSC0001I      TOTAL SENDCLNTS=2 RECV=0 CONN=2 XMIT=0 OTHER=0
```

The following example output corresponds to the preceding example output, and was produced by executing the command on IMS Connect instance HWS2, at the remote IMS installation where the MSC transactions are processed. HWS2 knows HWS1 only by the RMTIMSCON name ICON1. HWS2 receives MSC transactions on two active client connections, MSCBB435 and MSC84CF7, shown in the PORT information under CLIENTID. These IDs are the same IDs shown in the HWS1 example under SENDCLNT. After processing the MSC transactions, HWS2 sends the response messages back to ICON1 on the two active client connections shown under SENDCLNT in the RMTIMSCON information: MSCC73E0 and MSC0EBB0, the same IDs shown in the HWS1 example under CLIENTID.

```
HWSC0001I   HWS ID=HWS2     RACF=N  PSWDMC=R
HWSC0001I      UIDCACHE=N   UIDAGE=2147483647
HWSC0001I      MAXSOC=50  TIMEOUT=5000    TCPIPQ=50
HWSC0001I      NUMSOC=5      WARNSOC=80%  WARNINC=5%
HWSC0001I      RRS=N   STATUS=REGISTERED  MULTIRTP=  CASCADE=N
HWSC0001I      VERSION=V14 IP-ADDRESS=127.0.0.1
HWSC0001I      SUPER MEMBER NAME=      CM0 ACK TOQ=

HWSC0001I    ADAPTER=N     MAXCVRT=100   NUMCVRT=0
HWSC0001I      MAXLSSSZ=32767
HWSC0001I    DATASTORE=IMS2     STATUS=DISCONNECT
HWSC0001I      GROUP=XCFGRP1  MEMBER=HWS2
HWSC0001I      TARGET MEMBER=IMS2             STATE=N/A
HWSC0001I      DEFAULT REROUTE NAME=HWS$DEF
HWSC0001I      RACF APPL NAME=APPLID1         MULTIRTP=  CASCADE=
HWSC0001I      OTMA ACEE AGING VALUE=2147483647
HWSC0001I      OTMA ACK TIMEOUT VALUE=120
HWSC0001I      OTMA MAX INPUT MESSAGE=5000
HWSC0001I      SUPER MEMBER NAME=      CM0 ACK TOQ=
HWSC0001I    IMSPLEX=PLEX2    STATUS=ACTIVE
HWSC0001I      MEMBER=HWS2     TARGET=PLEX2
HWSC0001I    MSC=MSC21    STATUS=ACTIVE
HWSC0001I      RMTPLKID=MSC12
HWSC0001I      LCLIMSID=IMS2      RMTIMSID=IMS1
HWSC0001I      IMSPLEX=PLEX2
HWSC0001I        MEMBER=HWS2              TARGET MEMBER=PLEX2
HWSC0001I      RMTIMSCON=ICON1
HWSC0001I        IP-ADDRESS=192.0.2.0        PORT=9999
HWSC0001I        HOSTNAME=WWW.EXAMPLE.COM
HWSC0001I      LINK       PARTNERID   STATUS       SENDCLNT   RECVCLNT
HWSC0001I      LNK21T01   TA          ACTIVE       MSC0EBB0   MSC84CF7
HWSC0001I      LNK21T02   TB          ACTIVE       MSCC73E0   MSCBB435
HWSC0001I    NO ACTIVE ODBM
HWSC0001I    PORT=5555     STATUS=ACTIVE     KEEPAV=0 NUMSOC=3     EDIT=      TIMEOUT=0
HWSC0001I      CLIENTID LCLPLKID  STATUS      SECOND CLNTPORT IP-ADDRESS
HWSC0001I      MSCBB435 MSC21     RECV            18 1028     0:0:0:0:0:FFFF:C000:200
HWSC0001I      MSC84CF7 MSC21     RECV            69 1027     0:0:0:0:0:FFFF:C000:200
HWSC0001I      TOTAL CLIENTS=2  RECV=2 READ=0 CONN=0 XMIT=0 OTHER=0
HWSC0001I    RMTIMSCON=ICON1     STATUS=ACTIVE
HWSC0001I      IP-ADDRESS=192.0.2.0        PORT=9999
HWSC0001I      HOSTNAME=WWW.EXAMPLE.COM
HWSC0001I      AUTOCONN=N  PERSISTENT=Y
HWSC0001I      IDLETO=0
HWSC0001I      RESVSOC=2   NUMSOC=2
HWSC0001I      SENDCLNT LCLPLKID STATUS       SECOND SENDPORT
```

```
HWSC0001I      MSCC73E0 MSC21    CONN             18 1026
HWSC0001I      MSC0EBB0 MSC21    CONN             69 1025
HWSC0001I      TOTAL SENDCLNTS=2 RECV=0 CONN=2 XMIT=0 OTHER=0
```

Example 4: IMS Connect to IMS Connect connections for OTMA

If a connection between IMS Connect instances is used for OTMA messages, the command output shows different perspectives of the connection, depending on which instance of IMS Connect the command is executed on.

For OTMA, because the connections between IMS Connect instances are one-way connections, the connection is defined only in the sending IMS Connect instance. The connection is not defined in the remote IMS Connect instance.

In the following example output, the command is executed on HWS1, the IMS Connect instance at the IMS installation that sends the OTMA output messages. HWS1 is connected to a receiving IMS Connect instance, HWS2; however, HWS1 knows HWS2 only by the RMTIMSCON name ICON2B shown in the example. HWS1 sends the OTMA output messages to ICON2 for processing on the active client connection, OTM924FA, shown under SENDCLNT in the RMTIMSCON information. The SENDCLNT ID is auto-generated by HWS1 when establishing the connection with HWS2.

```
HWSC0001I   HWS ID=HWS1     RACF=N  PSWDMC=R
HWSC0001I      UIDCACHE=N   UIDAGE=2147483647
HWSC0001I      MAXSOC=2000  TIMEOUT=5000  TCPIPQ=50
HWSC0001I      NUMSOC=2      WARNSOC=80%  WARNINC=5%
HWSC0001I      RRS=N   STATUS=REGISTERED  MULTIRTP=  CASCADE=N
HWSC0001I      VERSION=V14 IP-ADDRESS=192.0.2.0
HWSC0001I      SUPER MEMBER NAME=      CM0 ACK TOQ=
HWSC0001I    ADAPTER=N  MAXCVRT=100  NUMCVRT=0
HWSC0001I      MAXLSSSZ=32767
HWSC0001I    DATASTORE=IMS1     STATUS=ACTIVE
HWSC0001I      GROUP=XCFGRP1  MEMBER=HWS1
HWSC0001I      TARGET MEMBER=IMS1             STATE=AVAIL
HWSC0001I      DEFAULT REROUTE NAME=HWS$DEF
HWSC0001I      RACF APPL NAME=APPLID1         MULTIRTP=  CASCADE=
HWSC0001I      OTMA ACEE AGING VALUE=2147483647
HWSC0001I      OTMA ACK TIMEOUT VALUE=120
HWSC0001I      OTMA MAX INPUT MESSAGE=5000
HWSC0001I      SUPER MEMBER NAME=      CM0 ACK TOQ=
HWSC0001I    NO ACTIVE IMSPLEX
HWSC0001I    NO ACTIVE ODBM
HWSC0001I    NO ACTIVE MSC
HWSC0001I    PORT=9999     STATUS=ACTIVE     KEEPAV=0 NUMSOC=1     EDIT=     TIMEOUT=0
HWSC0001I      NO ACTIVE CLIENTS
HWSC0001I    RMTIMSCON=ICON2B    STATUS=ACTIVE
HWSC0001I      IP-ADDRESS=127.0.0.1        PORT=5555
HWSC0001I      HOSTNAME=WWW.EXAMPLE.COM
HWSC0001I      AUTOCONN=N  PERSISTENT=Y
HWSC0001I      IDLETO=3000
HWSC0001I      RESVSOC=4   NUMSOC=1
HWSC0001I      SENDCLNT USERID   STATUS       SECOND SENDPORT
HWSC0001I      OTM924FA APOL1    CONN           5941 1026
HWSC0001I      TOTAL SENDCLNTS=1 RECV=0 CONN=1 XMIT=0 OTHER=0
```

The following example output corresponds to the preceding example output, and was produced by executing the command on HWS2, the IMS Connect instance at the IMS installation that receives the OTMA output messages. HWS2 knows HWS1 only by the auto-generated client ID, OTM924FA, shown under CLIENTID in the PORT information. The CLIENTID value is the same value shown under SENDCLNT at the local IMS installation. HWS2 receives the OTMA transactions on the connection identified by the client ID OTM924FA. After processing the OTMA

message, the remote IMS system queues any response messages to the tpipe hold queue identified by the client ID OTM924FA for later asynchronous retrieval.

```
HWSC0001I   HWS ID=HWS2     RACF=N  PSWDMC=R
HWSC0001I     UIDCACHE=N   UIDAGE=2147483647
HWSC0001I     MAXSOC=50  TIMEOUT=5000  TCPIPQ=50
HWSC0001I     NUMSOC=5      WARNSOC=80%  WARNINC=5%
HWSC0001I     RRS=N   STATUS=REGISTERED  MULTIRTP=  CASCADE=N
HWSC0001I     VERSION=V14 IP-ADDRESS=127.0.0.1
HWSC0001I     SUPER MEMBER NAME=      CM0 ACK TOQ=
HWSC0001I    ADAPTER=N  MAXCVRT=100  NUMCVRT=0
HWSC0001I     MAXLSSSZ=32767
HWSC0001I    DATASTORE=IMS2     STATUS=ACTIVE
HWSC0001I     GROUP=XCFGRP2  MEMBER=HWS2
HWSC0001I     TARGET MEMBER=IMS2             STATE=AVAIL
HWSC0001I     DEFAULT REROUTE NAME=HWS$DEF
HWSC0001I     RACF APPL NAME=APPLID1         MULTIRTP=  CASCADE=
HWSC0001I     OTMA ACEE AGING VALUE=2147483647
HWSC0001I     OTMA ACK TIMEOUT VALUE=120
HWSC0001I     OTMA MAX INPUT MESSAGE=5000
HWSC0001I     SUPER MEMBER NAME=      CM0 ACK TOQ=
HWSC0001I    NO ACTIVE IMSPLEX
HWSC0001I    NO ACTIVE ODBM
HWSC0001I    NO ACTIVE MSC
HWSC0001I    PORT=9999     STATUS=ACTIVE     KEEPAV=0 NUMSOC=1         EDIT=         TIMEOUT=0
HWSC0001I     NO ACTIVE CLIENTS
HWSC0001I    PORT=8888     STATUS=ACTIVE     KEEPAV=0 NUMSOC=1         EDIT=         TIMEOUT=0
HWSC0001I     NO ACTIVE CLIENTS
HWSC0001I    PORT=7777     STATUS=ACTIVE     KEEPAV=0 NUMSOC=1         EDIT=         TIMEOUT=0
HWSC0001I     NO ACTIVE CLIENTS
HWSC0001I    PORT=5555     STATUS=ACTIVE     KEEPAV=0 NUMSOC=2         EDIT=         TIMEOUT=0
HWSC0001I     CLIENTID USERID TRANCODE DATASTORE STATUS SECOND CLNTPORT IP-ADDRESS              APSB-TOKEN
HWSC0001I     OTM924FA APOL1  APOL11   IMS2      RECV       11 1026     0:0:0:0:0:FFFF:C000:200
HWSC0001I     TOTAL CLIENTS=1  RECV=1 READ=0 CONN=0 XMIT=0 OTHER=0
HWSC0001I    RMTIMSCON=ICON1B    STATUS=NOT ACTIVE
HWSC0001I     IP-ADDRESS=192.0.2.0        PORT=9999
HWSC0001I     HOSTNAME=WWW.EXAMPLE.COM
HWSC0001I     AUTOCONN=N  PERSISTENT=Y
HWSC0001I     IDLETO=0
HWSC0001I     RESVSOC=4   NUMSOC=0
HWSC0001I     NO ACTIVE CLIENTS
```

Example 5: ISC connections between IMS Connect and CICS

In the following example output, an ISC link, IMS1CIC4, links to a remote CICS subsystem by way of the RMTCICS connection, CICS4.

Currently, three parallel sessions, or *ISCUSERs*, are active: CIC4U01, CIC4U02, and CIC4U03. The status, the send client ID, and the receive client ID are shown on the row for each parallel session.

IMS Connect receives input messages from CICS on port 9876C, as defined by the CICSPORT keyword in the IMS Connect configuration member. Each receive socket for each parallel session on the CICSPORT is identified by a receive client ID that is generated by IMS Connect.

IMS Connect sends output messages to CICS through send sockets on ports that are automatically selected by IMS Connect. Each send socket for each parallel session is identified by a send client ID that is automatically generated by IMS Connect.

```
HWSC0001I HWS ID=HWS1     RACF=N  PSWDMC=R
HWSC0001I    UIDCACHE=N   UIDAGE=2147483647
HWSC0001I    MAXSOC=50  TIMEOUT=5000  TCPIPQ=50
HWSC0001I    NUMSOC=7      WARNSOC=80%  WARNINC=5%
HWSC0001I    RRS=N   STATUS=NOT ACTIVE  MULTIRTP=  CASCADE=N
```

```
HWSC0001I    VERSION=V14 IP-ADDRESS=127.0.0.1
HWSC0001I    SUPER MEMBER NAME=      CM0 ACK TOQ=

HWSC0001I ADAPTER=N MAXCVRT=100 NUMCVRT=0
HWSC0001I    MAXLSSSZ=32767
HWSC0001I  IMSPLEX=PLEX1    STATUS=ACTIVE
HWSC0001I    MEMBER=HWS1     TARGET=PLEX1
HWSC0001I  ISC=IMS1CIC4 STATUS=ACTIVE
HWSC0001I    NODE=CIC4
HWSC0001I    LCLIMSID=IMS1
HWSC0001I    IMSPLEX=PLEX1
HWSC0001I      MEMBER=HWS1              TARGET MEMBER=PLEX1
HWSC0001I    RMTCICS=CICS4
HWSC0001I      IP-ADDRESS=192.0.2.0        PORT=7777
HWSC0001I      HOSTNAME=WWW.EXAMPLE.COM
HWSC0001I    CICSAPPL=CICS1     CICSPORT=9876
HWSC0001I    CICSNETID=IMSNET
HWSC0001I    ISCUSER    STATUS       SENDCLNT   RECVCLNT
HWSC0001I    CIC4U01    ACTIVE       ISCE7811   ISCF4816
HWSC0001I    CIC4U02    ACTIVE       ISC3A819   ISC7121D
HWSC0001I    CIC4U03    ACTIVE       ISCECD99   ISC5C09F
HWSC0001I    TOTAL ISCUSERS=3 ACTIVE=3 OTHER=0
HWSC0001I  NO ACTIVE ODBM
HWSC0001I  NO ACTIVE MSC
HWSC0001I  PORT=9876C    STATUS=ACTIVE     KEEPAV=0 NUMSOC=4 EDIT=         TIMEOUT=0
HWSC0001I    CLIENTID ISCNAME   STATUS      SECOND CLNTPORT IP-ADDRESS
HWSC0001I    ISC5C09F IMS1CIC4  RECV             4 1037     0:0:0:0:0:FFFF:C000:200
HWSC0001I    ISC7121D IMS1CIC4  RECV            13 1035     0:0:0:0:0:FFFF:C000:200
HWSC0001I    ISCF4816 IMS1CIC4  RECV           200 1033     0:0:0:0:0:FFFF:C000:200
HWSC0001I    TOTAL CLIENTS=3  RECV=3 READ=0 CONN=0 XMIT=0 OTHER=0
HWSC0001I  NO ACTIVE RMTIMSCON
HWSC0001I  RMTCICS=CICS4     STATUS=ACTIVE
HWSC0001I    IP-ADDRESS=192.0.2.0      PORT=7777
HWSC0001I    HOSTNAME=WWW.EXAMPLE.COM
HWSC0001I    SENDCLNT ISCNAME  STATUS       SECOND SENDPORT
HWSC0001I    ISCECD99 IMS1CIC4 CONN             4 1036
HWSC0001I    ISC3A819 IMS1CIC4 CONN            13 1034
HWSC0001I    ISCE7811 IMS1CIC4 CONN           200 1032
HWSC0001I    TOTAL SENDCLNTS=3 RECV=0 CONN=3 XMIT=0 OTHER=0
```

Example 6: Multiple client types

In the following example output, IMS Connect supports many different client types, including:

- Clients that connect to IMS DB via a DRDA port and the Open Database Manager (ODBM) component of the IMS Common Service Layer (CSL).
- Clients that connect to IMS TM via a DATASTORE connection and the IMS Open Transaction Manager Access (OTMA) component.
- A CICS system that connects to IMS TM via ISC and the CICS IPIC protocol
- An IMS system that connects to another IMS system via OTMA
- An IMS system that connects in another IMS system via MSC

Certain output fields are always displayed while other output fields are displayed only if IMS Connect is actively supporting a specific client type.

```
HWSC0001I   HWS ID=ICON1    RACF=N  PSWDMC=R
HWSC0001I      MAXSOC=50  TIMEOUT=5000  TCPIPQ=50
HWSC0001I      NUMSOC=18     WARNSOC=80%  WARNINC=5%
HWSC0001I      RRS=N   STATUS=ACTIVE      MULTIRTP=  CASCADE=N
HWSC0001I      VERSION=V14 IP-ADDRESS=127.0.0.1
HWSC0001I      SUPER MEMBER NAME=      CM0 ACK TOQ=

HWSC0001I    ADAPTER=Y  MAXCVRT=100  NUMCVRT=0
HWSC0001I      MAXLSSSZ=32767
HWSC0001I    ODBM AUTO CONNECTION=Y
```

```
HWSC0001I    ODBM TIMEOUT=1200000
HWSC0001I    ODBM IMSPLEX MEMBER=                TARGET MEMBER=
HWSC0001I    DATASTORE=IMS1     STATUS=ACTIVE
HWSC0001I      GROUP=XCFGRP1  MEMBER=ICON1
HWSC0001I      TARGET MEMBER=IMS1             STATE=N/A
HWSC0001I      DEFAULT REROUTE NAME=THATWAY
HWSC0001I      RACF APPL NAME=APPLID1         MULTIRTP=  CASCADE=
HWSC0001I      OTMA ACEE AGING VALUE=2147483647
HWSC0001I      OTMA ACK TIMEOUT VALUE=120
HWSC0001I      OTMA MAX INPUT MESSAGE=5000
HWSC0001I      SUPER MEMBER NAME=      CM0 ACK TOQ=
HWSC0001I    IMSPLEX=PLEX1    STATUS=ACTIVE
HWSC0001I      MEMBER=ICON1    TARGET=PLEX1
HWSC0001I    ODBM=ODBM1OD  STATUS=REGISTERED   ODBMRRS=Y
HWSC0001I      ALIAS=ALI1     STATUS=ACTIVE
HWSC0001I      ALIAS=IMSA     STATUS=ACTIVE
HWSC0001I      ALIAS=IMS1     STATUS=ACTIVE
HWSC0001I      ALIAS=ALI2     STATUS=ACTIVE
HWSC0001I      ALIAS=IMS2     STATUS=ACTIVE
HWSC0001I    MSC=MSC12    STATUS=ACTIVE
HWSC0001I      RMTPLKID=MSC21
HWSC0001I      LCLIMS=IMS1 RMTIMS=IMS2 GENIMSID=IMS AFFINTY=IMS1
HWSC0001I      IMSPLEX=PLEX1
HWSC0001I        MEMBER=ICON1    TARGET MEMBER=PLEX1
HWSC0001I      RMTIMSCON=CONNECT2
HWSC0001I        IP-ADDRESS=192.0.2.0      PORT=5555
HWSC0001I        HOSTNAME=WWW.EXAMPLE.COM
HWSC0001I      LINK     PARTNERID STATUS     SENDCLNT RECVCLNT
HWSC0001I      LINK12A  AA        ACTIVE     MSC11111 MSC33333
HWSC0001I      LINK12B  BB        ACTIVE     MSC22222 MSC44444
HWSC0001I    ISC=CICS    STATUS=ACTIVE
HWSC0001I      NODE=CICSTRM1
HWSC0001I      LCLIMSID=IMS1
HWSC0001I      IMSPLEX=PLEX1
HWSC0001I        MEMBER=HWS1              TARGET MEMBER=PLEX1
HWSC0001I      RMTCICS=ICONNT1
HWSC0001I        IP-ADDRESS=192.0.2.1        PORT=9999
HWSC0001I        HOSTNAME=WWW.EXAMPLE.COM
HWSC0001I        CICSAPPL=CICSA1   CICSPORT=6666
HWSC0001I        CICSNETID=CICSNET
HWSC0001I      NO ACTIVE ISCUSER
HWSC0001I    ISC=LU6TCP2    STATUS=ACTIVE
HWSC0001I      NODE=CICSTRM2
HWSC0001I      LCLIMSID=IMS1
HWSC0001I      IMSPLEX=PLEX1
HWSC0001I        MEMBER=HWS1              TARGET MEMBER=PLEX1
HWSC0001I      RMTCICS=ICONNT2
HWSC0001I        IP-ADDRESS=192.0.2.2        PORT=9998
HWSC0001I        HOSTNAME=WWW.EXAMPLE.COM
HWSC0001I        CICSAPPL=CICSA2   CICSPORT=7777
HWSC0001I        CICSNETID=CICSNET
HWSC0001I      ISCUSER  STATUS    SENDCLNT RECVCLNT
HWSC0001I      IMSUS01  ACTIVE    ISC11111 ISC33333
HWSC0001I      IMSUS02  ACTIVE    ISC22222 ISC44444
HWSC0001I        TOTAL ISCUSERS=2 ACTIVE=2 OTHER=0
HWSC0001I    PORT=9996     STATUS=ACTIVE     KEEPAV=0 NUMSOC=4 EDIT=         TIMEOUT=0
HWSC0001I      CLIENTID USERID   TRANCODE DATASTORE STATUS  SECOND CLNTPORT IP-ADDRESS            APSB-TOKEN
HWSC0001I      CLIENT01 USRT003  ITOC04   IMS1       RECV        27 3464     0:0:0:0:0:FFFF:C000:204
HWSC0001I      CLIENTID LCLPLKID  STATUS SECOND CLNTPORT IP-ADDRESS
HWSC0001I      MSC33333 MSC12     CONN       14 1739     0:0:0:0:0:FFFF:C000:200
HWSC0001I      MSC44444 MSC12     CONN        9 2684     0:0:0:0:0:FFFF:C000:200
HWSC0001I      TOTAL CLIENTS=3  RECV=1 CONN=2 XMIT=0 OTHER=0
HWSC0001I    PORT=6666C    STATUS=NOT ACTIVE KEEPAV=0 NUMSOC=1 EDIT=         TIMEOUT=0
HWSC0001I      NO ACTIVE CLIENTS
HWSC0001I    PORT=7777C    STATUS=ACTIVE     KEEPAV=0 NUMSOC=3 EDIT=         TIMEOUT=0
HWSC0001I
HWSC0001I      CLIENTID ISCNAME STATUS SECOND CLNTPORT  IP-ADDRESS
HWSC0001I      ISC33333 LU6TCP2   CONN     14     3579  0:0:0:0:0:FFFF:C000:202
HWSC0001I      ISC44444 LU6TCP2   CONN      9     2468  0:0:0:0:0:FFFF:C000:202
HWSC0001I      TOTAL CLIENTS=3  RECV=1 CONN=2 XMIT=0 OTHER=0
```

```
HWSC0001I    PORT=LOCAL    STATUS=ACTIVE     KEEPAV=0 NUMSOC=0 EDIT=         TIMEOUT=0
HWSC0001I      NO ACTIVE CLIENTS
HWSC0001I    PORT=8899S    STATUS=NOT ACTIVE KEEPAV=0 NUMSOC=1 EDIT=         TIMEOUT=0
HWSC0001I      NO ACTIVE CLIENTS
HWSC0001I    PORT=8888D    STATUS=ACTIVE     KEEPAV=0 NUMSOC=1 EDIT=         TIMEOUT=600000
HWSC0001I      NO ACTIVE CLIENTS
HWSC0001I    RMTIMSCON=CONNECT2 STATUS=ACTIVE
HWSC0001I      IP-ADDRESS=192.0.2.0       PORT=5555
HWSC0001I      HOSTNAME=WWW.EXAMPLE.COM
HWSC0001I      AUTOCONN=N PERSISTENT=Y
HWSC0001I      IDLETO=6000
HWSC0001I      RESVSOC=10        NUMSOC=2
HWSC0001I      SENDCLNT LCLPLKID STATUS SECOND SENDPORT
HWSC0001I      MSC11111 MSC12    CONN      100 1234
HWSC0001I      MSC22222 MSC12    CONN       89 5678
HWSC0001I    RMTIMSCON=CONNECT3 STATUS=ACTIVE
HWSC0001I      IP-ADDRESS=192.0.2.4       PORT=9999
HWSC0001I      HOSTNAME=WWW.EXAMPLE.COM
HWSC0001I      AUTOCONN=Y PERSISTENT=Y
HWSC0001I      IDLETO=6000
HWSC0001I      RESVSOC=4         NUMSOC=4
HWSC0001I      SENDCLNT USERID   STATUS SECOND SENDPORT
HWSC0001I      OTM11111 USER01   CONN      100 1111
HWSC0001I      OTM22222 USER01   CONN       89 2222
HWSC0001I      OTM33333 USER02   CONN       81 3333
HWSC0001I      OTM44444 USER03   RECV       23 4444
HWSC0001I      TOTAL SENDCLNTS=4 RECV=1 CONN=3 XMIT=0 OTHER=0
HWSC0001I    RMTCICS=ICONNT1   STATUS=NOT ACTIVE
HWSC0001I      IP-ADDRESS=192.0.2.1        PORT=9997
HWSC0001I      HOSTNAME=WWW.EXAMPLE.COM
HWSC0001I      NO ACTIVE CLIENTS
HWSC0001I    RMTCICS=ICONNT2   STATUS=ACTIVE
HWSC0001I      IP-ADDRESS=192.0.2.2        PORT=9998
HWSC0001I      HOSTNAME=WWW.EXAMPLE.COM
HWSC0001I      SENDCLNT ISCNAME     STATUS SECOND SENDPORT
HWSC0001I      ISC11111 LU6TCP2     CONN      100 1111
HWSC0001I      ISC22222 LU6TCP2     CONN       89 2222
HWSC0001I      TOTAL SENDCLNTS=2 RECV=0 CONN=2 XMIT=0 OTHER=0
```

Related reference:

z/OS: IPv6 addressing

QUERY IMSCON TYPE(CONFIG) command (Commands)

"IMS Connect QUERY MEMBER command" on page 547

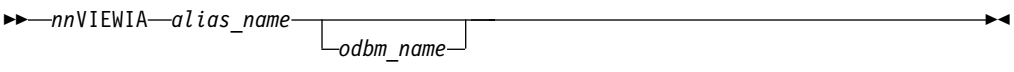
VIEWIA command

Use the VIEWIA command to display the status of IMS aliases and the associated ODBMs.

Subsections:

- "Syntax"
- "Keywords" on page 513
- "Equivalent IMS Connect type-2 and z/OS commands" on page 513
- "Examples" on page 513

Syntax



Keywords

The following keywords are valid for the VIEWIA command:

nn 2-digit number of z/OS system console outstanding display.

alias_name
Specifies the name of the IMS alias. Wildcards (* and %) are supported in the name.

odbm_name
Specifies the name of the ODBM. Wildcards (* and %) are supported in the name. This parameter is optional.

Equivalent IMS Connect type-2 and z/OS commands

The following table lists IMS Connect type-2 commands and IMS Connect z/OS commands that perform similar functions as the VIEWIA command.

VIEWIA command	Equivalent IMS Connect type-2 command	Equivalent IMS Connect z/OS command
VIEWIA ALL	QUERY IMSCON TYPE(ALIAS) NAME(*) SHOW(ALL|*show_parm*)	QUERY ALIAS NAME(*)
VIEWIA *alias_name*	QUERY IMSCON TYPE(ALIAS) NAME(*alias_name*) SHOW(ALL|*show_parm*)	QUERY ALIAS NAME(*aliasName*)
VIEWIA *alias_name odbm_name*	QUERY IMSCON TYPE(ALIAS) NAME(*alias_name*) ODBM(*odbm_name*)	QUERY ALIAS NAME(*aliasName*) ODBM(*odbmName*)

Examples

The following are examples of the VIEWIA command:

Example 1 for VIEWIA command

Command input:

```
nnVIEWIA IMSA
```

Command output:

```
HWSC0001I    ALIAS=IMSA ODBM=ODBM1OD  STATUS=ACTIVE
HWSC0001I    ALIAS=IMSA ODBM=ODBM2OD  STATUS=NOT ACTIVE(IMSCON)
```

Explanation: This example shows the status of IMS alias IMSA and the associated ODBMs. IMSA is active when routed through ODBM1OD, but not active when routed through ODBM2OD. The status `NOT ACTIVE(IMSCON)` indicates that the alias is deactivated by the IMS Connect STOPIA command.

Example 2 for VIEWIA command

Command input:

```
nnVIEWIA IMSA ODBM1OD
```

Command output:

```
HWSC0001I    ALIAS=IMSA ODBM=ODBM1OD  STATUS=ACTIVE
```

Explanation: This example shows the status of IMS alias IMSA associated with an ODBM named ODBM1OD. IMSA is active when routed through ODBM1OD.

Example 3 for VIEWIA command

Command input:

```
nnVIEWIA IMSA ODBM*
```

Command output:

```
HWSC0001I    ALIAS=IMSA ODBM=ODBM1OD  STATUS=ACTIVE
HWSC0001I    ALIAS=IMSA ODBM=ODBM2OD  STATUS=ACTIVE
```

Explanation: This example shows the status of IMS alias IMSA associated with ODBMs whose names start with "ODBM". IMSA is active when routed through ODBM1OD and ODBM2OD.

Example 4 for VIEWIA command

Command input:

```
nnVIEWIA I*
```

Command output:

```
HWSC0001I    ALIAS=IMS1 ODBM=ODBM1OD  STATUS=ACTIVE
HWSC0001I    ALIAS=IMSA ODBM=ODBM1OD  STATUS=NOT ACTIVE(IMSCON)
HWSC0001I    ALIAS=IM1A ODBM=ODBM1OD  STATUS=ACTIVE
HWSC0001I    ALIAS=IMS1 ODBM=ODBM2OD  STATUS=NOT ACTIVE(IMSCON,ODBM)
HWSC0001I    ALIAS=IMSA ODBM=ODBM2OD  STATUS=NOT ACTIVE(ODBM)
HWSC0001I    ALIAS=IMS2 ODBM=ODBM2OD  STATUS=ACTIVE
HWSC0001I    ALIAS=IMS3 ODBM=ODBM3OD  STATUS=NOT ACTIVE(IMSCON)
HWSC0001I    ALIAS=IO3A ODBM=ODBM3OD  STATUS=NOT ACTIVE(ODBM)
```

Explanation: This example shows the status of IMS aliases whose names start with "I" and the associated ODBMs. The status `NOT ACTIVE` indicates whether the alias is deactivated by the IMS Connect STOPIA command (IMSCON), or by the UPDATE ODBM STOP(CONNECTION) ALIAS command (ODBM), or both.

Example 5 for VIEWIA command

Command input:

```
nnVIEWIA IMS* ODBM2OD
```

Command output:

```
HWSC0001I    ALIAS=IMSA ODBM=ODBM2OD  STATUS=ACTIVE
HWSC0001I    ALIAS=IMSB ODBM=ODBM2OD  STATUS=ACTIVE
HWSC0001I    ALIAS=IMSC ODBM=ODBM2OD  STATUS=ACTIVE
HWSC0001I    ALIAS=IMSD ODBM=ODBM2OD  STATUS=ACTIVE
```

Explanation: This example shows the status of IMS aliases whose names start with "IMS" and are associated with an ODBM named ODBM2OD.

Example 6 for VIEWIA command

Command input:

```
nnVIEWIA IMS* ODBM*
```

Command output:

```
HWSC0001I    ALIAS=IMSA ODBM=ODBM1OD  STATUS=ACTIVE
HWSC0001I    ALIAS=IMSB ODBM=ODBM1OD  STATUS=ACTIVE
HWSC0001I    ALIAS=IMSA ODBM=ODBM2OD  STATUS=ACTIVE
HWSC0001I    ALIAS=IMSB ODBM=ODBM2OD  STATUS=ACTIVE
HWSC0001I    ALIAS=IMSC ODBM=ODBM2OD  STATUS=ACTIVE
HWSC0001I    ALIAS=IMSD ODBM=ODBM2OD  STATUS=ACTIVE
```

Explanation: This example shows the status of IMS aliases whose names start with "IMS" and are associated with ODBMs whose name start with "ODBM".

Example 7 for VIEWIA command

Command input:

```
nnVIEWIA IMS% ODBM%OD
```

Command output:

```
HWSC0001I    ALIAS=IMSA ODBM=ODBM1OD  STATUS=ACTIVE
HWSC0001I    ALIAS=IMSB ODBM=ODBM2OD  STATUS=ACTIVE
HWSC0001I    ALIAS=IMSA ODBM=ODBM3OD  STATUS=ACTIVE
HWSC0001I    ALIAS=IMSB ODBM=ODBM4OD  STATUS=ACTIVE
HWSC0001I    ALIAS=IMSC ODBM=ODBM5OD  STATUS=ACTIVE
HWSC0001I    ALIAS=IMSD ODBM=ODBM6OD  STATUS=ACTIVE
```

Explanation: This example shows the status of IMS aliases whose names are 4 characters long and start with "IMS", and are associated with ODBMs whose names are 7 characters long and start with "ODBM" and end with "OD".

Related reference:

QUERY IMSCON TYPE(ALIAS) command (Commands)

"IMS Connect QUERY ALIAS command" on page 539

VIEWIP command

Use the VIEWIP command to display the current activity for the IMSplex.

Subsections:

- "Syntax"
- "Keywords"
- "Usage notes" on page 516
- "Equivalent IMS Connect type-2 and z/OS commands" on page 516
- "Examples" on page 516

Syntax

Keywords

The following keywords are valid for the VIEWIP command:

nn 2-digit number of z/OS system console outstanding display.

imsplex_id
Specifies the name of the IMSplex for which information is to be displayed. If the IMSplex name is used, this name must match the ID parameter of the IMSplex configuration statement in the HWSCFGxx.

Usage notes

The VIEWIP command displays the current information for the IMSplex. The information displayed for the IMSplex is:

IMSPLEX=
Name of the IMSplex, as defined in the ID parameter of the IMSplex configuration statement in the IMS Connect configuration member, HWSCFGxx.

STATUS=
State of the IMSplex: ACTIVE, NOT ACTIVE, or DISCONNECT.

If the IMSplex does down, IMS Connect is notified (through SCI) of the status of the IMSplex. When the IMSplex is brought back up and restarted, IMS Connect is notified and automatically reconnects to IMSplex.

MEMBER=
Name of the member as defined in the Member parameter of the IMSplex configuration statement in the IMS Connect configuration member, HWSCFGxx.

TARGET MEMBER=
Name of the target member of the IMSplex SCI to which IMS has connected and defined in the TMEMBER parameter of the IMSplex configuration statement in the IMS Connect configuration member, HWSCFGxx.

Equivalent IMS Connect type-2 and z/OS commands

The following table lists IMS Connect type-2 commands and IMS Connect z/OS commands that perform similar functions as the VIEWIP command.

VIEWIP command	Equivalent IMS Connect type-2 command	Equivalent IMS Connect z/OS command
VIEWIP ALL	QUERY IMSCON TYPE(IMSPLEX) NAME(*) SHOW(ALL | *show_parm*)	QUERY IMSPLEX NAME(*) SHOW(ALL)
VIEWIP *IMSplex_name*	QUERY IMSCON TYPE(IMSPLEX) NAME(*IMSplex_name*) SHOW(ALL | *show_parm*)	QUERY IMSPLEX NAME(*imsplexName*) SHOW(ALL)

Examples

To view the information for the IMSplex, with TMEMBER name of IMSPLEX1:

```
nnVIEWIP IMSPLEX1
```

Where *nn* is the 2-digit number of z/OS system console outstanding display.

Related reference:

QUERY IMSCON TYPE(IMSPLEX) command (Commands)

"IMS Connect QUERY IMSPLEX command" on page 545

VIEWMSC command

Use the IMS Connect WTOR command VIEWMSC to display the attributes and status of the MSC links that IMS Connect supports.

The VIEWMSC command displays the same information as the z/OS MODIFY command QUERY MSC.

To display information about the sockets that are currently in use by MSC logical links on an MSC physical link that is defined to IMS Connect, use the IMS Connect WTOR command VIEWRMT or the IMS Connect z/OS MODIFY command QUERY RMTIMSCON.

Subsections:

- "Syntax"
- "Keywords"
- "Usage notes"
- "Equivalent IMS Connect type-2 and z/OS commands"
- "Output fields" on page 518
- "Examples" on page 520

Syntax

Keywords

The following keywords are valid for the VIEWMSC command:

ALL
Displays information about all MSC physical links that are defined to this instance IMS Connect.

nn The two-digit number of the z/OS system console outstanding display.

lclplk_id
The ID of the MSC physical link definition, as specified on the LCLPLKID parameter of the MSC statement in the HWSCFGxx PROCLIB member.

Usage notes

Use the VIEWMSC command to display information of the MSC physical link defined in the LCLPLKID parameter of the MSC statement in the IMS Connect configuration member.

Equivalent IMS Connect type-2 and z/OS commands

The following table lists IMS Connect type-2 commands and IMS Connect z/OS commands that perform similar functions as the VIEWMSC command.

VIEWMSC command	Equivalent IMS Connect type-2 command	Equivalent IMS Connect z/OS command
VIEWMSC ALL	QUERY IMSCON TYPE(MSC) NAME(*) SHOW(ALL | *show_parm*)	QUERY MSC NAME(*)
VIEWMSC *msc_id*	QUERY IMSCON TYPE(MSC) NAME(*msc_id*) SHOW(ALL | *show_parm*)	QUERY MSC NAME(*msc_id*)

Output fields

Output fields that return only a single value are displayed in the format `FIELD_LABEL=VALUE`, for example: `MSC=MSC12`.

Output fields that return multiple values, such as the output fields for multiple logical links on an MSC physical link, are displayed in a column format, where the field label is a column heading and the values appear under the heading on separate rows in the column. For example:

```
HWSC0001I      LINK     PARTNERID STATUS     SENDCLNT RECVCLNT
HWSC0001I      LINK12A  AA        ACTIVE     MSC11111 MSC33333
HWSC0001I      LINK12B  BB        ACTIVE     MSC22222 MSC44444
```

The following list describes the output fields of the VIEWMSC command.

MSC= Name of the displayed MSC physical link, as specified on the LCLPLKID parameter of the MSC statement in the IMS Connect configuration member. The following additional output fields describe the status, attributes, and logical links that belong to the physical link identified in the MSC= output field:

STATUS
The state of the MSC physical link displayed in the MSC output field. A physical link can be in the following states:

ACTIVE
The MSC physical link is enabled for communications between IMS Connect and the local IMS system.

NOT ACTIVE
The MSC physical link is disabled for communications between IMS Connect and the local IMS system.

RMTPLKID
The ID of the MSC physical link at the remote IMS site, as defined in the RMTPLKID parameter of the MSC statement in the IMS Connect configuration member.

LCLIMSID
The ID of the local MSC-enabled IMS system, as defined to IMS Connect on the LCLIMS parameter of the MSC statement in the IMS Connect configuration member. This LCLIMS value must match the IMS ID that the local IMS system registers with SCI in the IMSplex. You can determine the IMS ID that is registered with SCI by issuing the QUERY IMSPLEX command in the local IMSplex.

If the second IMS ID is shown in the output, it identifies the other IMS system in an XRF couple.

RMTIMSID
The name of the remote MSC-enabled IMS system, as defined on the RMTIMS parameter of the MSC statement in the IMS Connect configuration member.

GENIMSID
The generic IMS ID that is being used for TCP/IP generic resources for MSC, as defined in the GENIMSID parameter of the MSC statement in the IMS Connect configuration member.

AFFINITY
The name of the IMS system with which the MSC logical link currently has affinity for TCP/IP generic resources.

IMSPLEX
Name of the IMSplex that manages communications between IMS Connect and the local, MSC-enabled IMS system. This name is the IMSplex name that is specified on the IMSPLEX parameter of MSC statement in the IMS Connect configuration member.

This IMSPLEX name must match the name specified on the IMSPLEX(NAME=) statement of the CSLSIxxx PROCLIB member of the SCI instance that is managing communications between IMS Connect and the IMSplex.

MEMBER
The name by which this IMS Connect is registered in the IMSplex, as defined on the MEMBER parameter in the IMSPLEX substatement of the MSC statement in the IMS Connect configuration member.

This MEMBER name must also be specified on the LCLICON parameter in the MSPLINK macro that defines this MSC physical link in the local IMS system.

TARGET MEMBER
Name of the IMSplex as defined in the TMEMBER subparameter of the IMSPLEX parameter of the MSC statement in the IMS Connect configuration member.

RMTIMSCON
The name of the IMS Connect connection that is used by this MSC physical link, as defined in the RMTIMSCON parameter of the MSC statement in the IMS Connect configuration member.

IP-ADDRESS
The IP address of the remote IMS Connect instance.

PORT The port number used by the remote IMS Connect instance.

HOSTNAME
The host name of the remote IMS Connect instance, if one is available.

LINK The ID of an MSC logical link that is assigned to this physical link.

The values returned for this output field are displayed in column format.

If no logical links are currently assigned to this physical link, this output field is not displayed.

PARTNERID

The ID of the remote partner for this logical link, as specified on the PARTNER parameter in the MSLINK macro that defines this logical link.

The values returned for this output field are displayed in column format.

If no logical links are currently assigned to this physical link, this output field is not displayed.

STATUS
The current state of this logical link. A logical link can have the following state:

ACTIVE
This logical link can send messages to the remote IMS Connect.

The values returned for this output field are displayed in column format.

If no logical links are currently assigned to this physical link, this output field is not displayed.

SENDCLNT
The send socket ID generated by this instance of IMS Connect to identify a send socket opened on this connection. The SENDCLNT ID is used as the client ID when establishing a socket connection with the remote IMS Connect instance.

The send socket IDs generated for sockets on connections that are used for OTMA begin with "OTM." The send socket IDs that are generated for sockets on connections that are used for MSC begin with "MSC."

The values returned for this output field are displayed in column format.

If no logical links are currently assigned to this physical link, this output field is not displayed.

RECVCLNT
The client ID that identifies the receive socket for this logical link. The client ID is generated by the remote IMS Connect instance.

The values returned for this output field are displayed in column format.

If no logical links are currently assigned to this physical link, this output field is not displayed.

Examples

In the following example, the VIEWMSC command returns information about the MSC physical link MSC13:

```
nnVIEWMSC MSC13

HWSC0001I    MSC=MSC13    STATUS=ACTIVE
HWSC0001I      RMTPLKID=MSC31
HWSC0001I      LCLIMSID=IMS1      RMTIMSID=IMS3
HWSC0001I      GENIMSID=          AFFINITY=
HWSC0001I      IMSPLEX=PLEX1
HWSC0001I        MEMBER=HWS1              TARGET MEMBER=PLEX1
HWSC0001I      RMTIMSCON=ICONNT2
HWSC0001I        IP-ADDRESS=127.0.0.1        PORT=9992
HWSC0001I        HOSTNAME=
HWSC0001I      NO ACTIVE LINK
```

In the following example, the VIEWMSC command returns information about the MSC physical link MSC12:

```
nnVIEWMSC MSC12
```

```
HWSC0001I    MSC=MSC12    STATUS=ACTIVE
HWSC0001I      RMTPLKID=MSC21
HWSC0001I      LCLIMSID=IMS1    ,IMS3      RMTIMSID=IMS3
HWSC0001I      GENIMSID=IMS       AFFINITY=
HWSC0001I      IMSPLEX=PLEX1
HWSC0001I        MEMBER=HWS1              TARGET MEMBER=PLEX1
HWSC0001I      RMTIMSCON=ICONNT1
HWSC0001I        IP-ADDRESS=192.0.2.2        PORT=9999
HWSC0001I        HOSTNAME=WWW.EXAMPLE.COM
HWSC0001I      NO ACTIVE LINK
```

Related reference:

"IMS Connect QUERY MSC command" on page 571

QUERY IMSCON TYPE(MSC) command (Commands)

VIEWPORT command

Use the VIEWPORT command to display the status, clients, and activity of a port.

Subsections:

- "Syntax"
- "Keywords"
- "Usage notes" on page 522
- "Equivalent IMS Connect type-2 and z/OS commands" on page 522
- "Output fields" on page 522
- "Example 1 for VIEWPORT command" on page 524

Syntax

Keywords

The following keywords are valid for the VIEWPORT command:

nn 2-digit number of z/OS system console outstanding display.

portid Specifies the port for which information is to be displayed. The *portid* specified must match the port number defined on the PORTID, PORT, or SSLPORT substatement of the TCPIP configuration statement or on the DRDAPORT substatement of the ODACCESS configuration statement in the IMS Connect configuration PROCLIB member. Information displayed is for that port only.

ALL Displays information about all the ports defined in the TCPIP and ODACCESS configuration statements in the IMS Connect configuration PROCLIB member.

LOCAL
Displays information about all clients that are using local port communications through IMS TM Resource Adapter.

Usage notes

Use the VIEWPORT command to display the current status, client, and activity of one port or all ports.

The information displayed by *nn*`VIEWPORT ALL` is the same as the information returned for all ports by the VIEWHWS command.

Equivalent IMS Connect type-2 and z/OS commands

The following table lists IMS Connect type-2 commands and IMS Connect z/OS commands that perform similar functions as the VIEWPORT command.

VIEWPORT command	Equivalent IMS Connect type-2 command	Equivalent IMS Connect z/OS command
VIEWPORT ALL	QUERY IMSCON TYPE(PORT) NAME(*) SHOW(ALL | *show_parm*)	QUERY PORT NAME(*) SHOW(ALL)
VIEWPORT *port_id*	QUERY IMSCON TYPE(PORT) NAME(*portid*) SHOW(ALL | *show_parm*)	QUERY PORT NAME(*port_id*) SHOW(ALL)
VIEWPORT LOCAL	QUERY IMSCON TYPE(PORT) NAME(LOCAL) SHOW(ALL | *show_parm*)	QUERY PORT NAME(LOCAL) SHOW(ALL)

Output fields

The following list describes the output fields of the VIEWPORT command.

PORT=
Identifies the port or ports to which the following output fields and statuses apply. The port numbers are defined in the IMS Connect configuration member in the IMS.PROCLIB data set.

You can determine the type of a port by the port-type indicator that is appended to the end of the port number. IMS Connect uses the following port-type indicators:

No type indicator
Indicates a port that can be used for IMS TM OTMA clients and IMS-to-IMS MSC or OTMA communications. Ports that do not have a type indicator are defined on either the PORT keyword or the PORTID keyword of the TCPIP statement in the IMS Connect configuration member.

C Indicates an IBM CICS Transaction Server for z/OS port for ISC communications that is defined to IMS Connect on the CICSPORT parameter of the TCPIP and ISC statements. CICS ports are used exclusively by CICS subsystems that access IMS TM via an ISC TCP/IP link.

D Indicates a Distributed Relational Database Architecture (DRDA) port that is defined to IMS Connect on the DRDAPORT parameter. DRDA ports are used exclusively by client application programs that access IMS DB through the Open Database Manager (ODBM).

S Indicates an SSL port that is defined to IMS Connect on the SSLPORT parameter. SSL ports are used only when IMS Connect support for SSL is enabled.

The following output fields apply to the port identified in the preceding PORT output field.

STATUS
State of the port, whether ACTIVE or INACTIVE.

KEEPAV
The amount of time a connection remains idle before the z/OS TCP/IP layer sends a packet to maintain the connection.

NUMSOC
The number of sockets used on each port.

TIMEOUT
The amount of time IMS Connect waits before terminating a client connection if no messages are received from the client.

CLIENTID ISCNAME STATUS SECOND CLNTPORT IP-ADDRESS
or
CLIENTID LCLPLKID STATUS SECOND CLNTPORT IP-ADDRESS
or
CLIENTID USERID TRANCODE DATASTORE STATUS SECOND CLNTPORT IP-ADDRESS APSB-TOKEN
The ID and information for an active client on a receive socket.

The information for each active client is displayed on a single row. Clients are grouped together by client type under a row of headings that identify the information that is returned for that client type.

For ISC clients, the following information is displayed: CLIENTID ISCNAME STATUS SECOND CLNTPORT IP-ADDRESS.

For MSC clients, the following information is displayed: CLIENTID LCLPLKID STATUS SECOND CLNTPORT IP-ADDRESS.

For OTMA and ODBM clients, the following information is displayed: CLIENTID USERID TRANCODE DATASTORE STATUS SECOND CLNTPORT IP-ADDRESS APSB-TOKEN.

The following example shows the information returned for three OTMA clients:

```
HWSC0001I  PORT=9999     STATUS=ACTIVE     KEEPAV=0 NUMSOC=4 EDIT=         TIMEOUT=0
HWSC0001I    CLIENTID USERID   TRANCODE DATASTORE STATUS    SECOND  CLNTPORT IP-ADDRESS       APSB-TOKEN
HWSC0001I    CLIENT63 USRT003  apol12   IMSY      RECV WFCM     55      4118 192.0.2.0
HWSC0001I    CLIENT62 USRT002           IMS1      CONN         210      4115 192.0.2.0
HWSC0001I    CLIENT61 USRT001  apol12   IMSN      RECV          13      4112 192.0.2.0
HWSC0001I    TOTAL CLIENTS=3  RECV=2 CONN=1 XMIT=0 OTHER=0
```

If a port has no active clients on it, these output fields are not displayed. Instead, the command output displays NO ACTIVE CLIENTS after the row that identifies the port.

The information returned for active clients includes, in alphabetical order:

APSB-TOKEN
Open Database Manager (ODBM) APSB token.

CLIENTID
The name of the client.

For connections from other IMS Connect instances, the other IMS Connect instance generates this value.

CLNTPORT
A random number that TCP/IP generates to represent a connection from a client.

DATASTORE
The data store associated with this client.

The IP address of the client for this connection.

If IPV6 is enabled, the IP address format consists of eight hexadecimal numbers divided by colons. If IPV6 is not enabled, the IP address format of IPV4 is used. The following example is for an IPV6 IP address that is displayed in IPV6 format:

```
FEDC:ABCD:2222:3333:FEDC:DB55:6666:3322
```

The following example is for an IPV4 IP address that is displayed in IPV6 format:

```
0:0:0:0:0:FFFF:945:33FF
```

ISCNAME
The name of the ISC statement in the IMS Connect configuration member.

LCLPLKID
The MSC physical link in the local IMS system to which IMS Connect routes the messages from the client. This field exists only for MSC communications.

SECOND
Number of seconds that the client has been in the specified status.

STATUS
State of the thread of the client. The client thread state can be one of the following values:

RECV In a receive state waiting for input from client.

READ In a partial read state. IMS Connect is reading a message from the client but has not yet received the entire length of the message.

CONN
Waiting for output from IMS.

XMIT Sending data to client.

CONV
In a conversational state.

WFCM
Waiting for confirmation (ACK, NAK, or DEALLOCATE) from client.

TRANCODE
The transaction code submitted by the client.

USERID
The USERID name passed to IMS Connect.

Example 1 for VIEWPORT command

In the following example, *nn* is the 2-digit number of z/OS system console outstanding display.

To view the information for a single port, 9999:

```
nnVIEWPORT 9999
```

To view the information for all ports defined to the IMS Connect:

```
nnVIEWPORT ALL
```

The command *nn*VIEWPORT ALL produces the following output:

```
HWSC0001I  PORT=9999     STATUS=ACTIVE     KEEPAV=0 NUMSOC=4 EDIT=         TIMEOUT=0
HWSC0001I    CLIENTID USERID   TRANCODE DATASTORE STATUS    SECOND  CLNTPORT IP-ADDRESS       APSB-TOKEN
HWSC0001I    CLIENT63 USRT003  apol12   IMSY      RECV WFCM     55      4118 192.0.2.0
HWSC0001I    CLIENT62 USRT002           IMS1      CONN         210      4115 192.0.2.1
HWSC0001I    CLIENT61 USRT001  apol12   IMSN      RECV          13      4112 192.0.2.2
HWSC0001I    TOTAL CLIENTS=3  RECV=2 CONN=1 XMIT=0 OTHER=0
HWSC0001I  PORT=LOCAL    STATUS=ACTIVE     KEEPAV=0 NUMSOC=0 EDIT=         TIMEOUT=0
HWSC0001I    NO ACTIVE CLIENTS
HWSC0001I  PORT=9998S    STATUS=NOT ACTIVE KEEPAV=0 NUMSOC=1 EDIT=         TIMEOUT=0
HWSC0001I  PORT=1111D    STATUS=ACTIVE     KEEPAV=5 NUMSOC=1 EDIT=         TIMEOUT=6000
HWSC0001I    NO ACTIVE CLIENTS
HWSC0001I  PORT=2222D    STATUS=ACTIVE     KEEPAV=0 NUMSOC=1 EDIT=         TIMEOUT=100
HWSC0001I    NO ACTIVE CLIENTS
```

Example 2: IMS Connect to IMS Connect connections for MSC

For IMS Connect to IMS Connect connections that are used for MSC, the information displayed by VIEWPORT and QUERY PORT commands relates only to the receiving end of the IMS Connect to IMS Connect connection. To view information about the sending side of an IMS Connect to IMS Connect connection, use the VIEWRMT or QUERY RMTIMSCON command.

In the following example output, the VIEWPORT or QUERY PORT command was executed on a local IMS Connect. The command output displays the information for port 9999, on which IMS Connect receives MSC messages. The client connections on the port, MSCC73E0 and MSC0EBB0, are shown under CLIENTID. The LCLPLKID label identifies this port as being used for MSC TCP/IP communication.

```
HWSC0001I    PORT=9999     STATUS=ACTIVE     KEEPAV=0 NUMSOC=3      EDIT=       TIMEOUT=0
HWSC0001I      CLIENTID LCLPLKID  STATUS      SECOND CLNTPORT IP-ADDRESS
HWSC0001I      MSCC73E0 MSC12     RECV             6 1026     2001:DB8:0:0:0:0:0:0/32
HWSC0001I      MSC0EBB0 MSC12     RECV            56 1025     2001:DB8:0:0:0:0:0:0/32
HWSC0001I      TOTAL CLIENTS=2  RECV=2 READ=0 CONN=0 XMIT=0 OTHER=0
```

Example 3: IMS Connect to IMS Connect connections for OTMA

If a socket connection between IMS Connect instances is used for OTMA messages, the VIEWPORT or QUERY PORT command output shows relevant port information only when issued at the receiving IMS Connect instance.

In the sending IMS Connect instance, the VIEWPORT or QUERY PORT command does not display port information that is relevant to the IMS Connect to IMS Connect connection. However, the VIEWRMT or QUERY RMTIMSCON command displays the listening port of the receiving IMS Connect, as well as the send client ID associated with the connection. The listening port and send client ID match the information displayed at the receiving installation by the VIEWPORT or QUERY PORT command.

In the receiving IMS Connect instance, the VIEWPORT or QUERY PORT command displays the port that the receiving IMS Connect is listening on and the sending client ID associated with the connection.

The following example shows the output from the VIEWPORT or QUERY PORT command on a receiving IMS Connect instance. The port that the receiving IMS Connect uses for the IMS Connect to IMS Connect connection is identified by PORT=5555. The connection from the sending IMS Connect is identified by the auto-generated client ID, OTM924FA, that is shown under CLIENTID. This CLIENTID value is the same value shown under SENDCLNT at the sending IMS installation. The CLNTPORT is the port that the sending IMS Connect uses to send messages on the connection.

```
HWSC0001I    PORT=5555     STATUS=ACTIVE     KEEPAV=0 NUMSOC=2       EDIT=        TIMEOUT=0
HWSC0001I      CLIENTID USERID TRANCODE DATASTORE STATUS  SECOND CLNTPORT IP-ADDRESS               APSB-TOKEN
HWSC0001I      OTM924FA APOL1  APOL11   IMS2      RECV        11 1026     2001:DB8:0:0:0:0:0:0/32
HWSC0001I      TOTAL CLIENTS=1  RECV=1 READ=0 CONN=0 XMIT=0 OTHER=0
```

Example 4: An ISC TCP/IP connection to CICS

Command input:

```
nnVIEWPORT 7777
```

Command output:

```
HWSC0001I  PORT=7777C  STATUS=ACTIVE   KEEPAV=0 NUMSOC=3 EDIT=       TIMEOUT=0
HWSC0001I     CLIENTID ISCNAME STATUS SECOND CLNTPORT  IP-ADDRESS
HWSC0001I     ISC33333 LU6TCP2   CONN     14     3579  2001:DB8:0:0:0:0:0:0/32
HWSC0001I     ISC44444 LU6TCP2   CONN      9     2468  2001:DB8:0:0:0:0:0:0/32
HWSC0001I     TOTAL CLIENTS=3  RECV=1 CONN=2 XMIT=0 OTHER=0
```

Related reference:

z/OS: IPv6 addressing

QUERY IMSCON TYPE(PORT) command (Commands)

"IMS Connect QUERY PORT command" on page 576

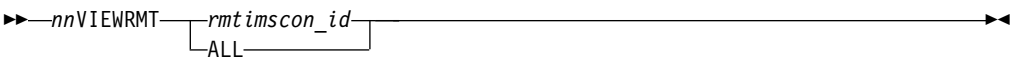
VIEWRMT command

Use the VIEWRMT command in a local instance of IMS Connect to display information about all connections to a remote instance of IMS Connect.

Subsections:

- "Syntax"
- "Keywords"
- "Usage notes" on page 527
- "Equivalent IMS Connect type-2 and z/OS commands" on page 527
- "Examples" on page 530

Syntax



Keywords

The following keywords are valid for the VIEWRMT command:

ALL
Displays information about all remote IMS Connect connections that are defined to this instance IMS Connect.

nn The two-digit number of the z/OS system console outstanding display.

rmtimscon_id
The ID of a connection to a remote instance of IMS Connect. The value specified here must match a value specified on the ID parameter of a RMTIMSCON statement that is defined to the local IMS Connect instance.

Usage notes

Use the VIEWRMT command to display information about the attributes and status of connections with remote instances of IMS Connect and about the sockets that are currently open on the connections.

Equivalent IMS Connect type-2 and z/OS commands

The following table lists IMS Connect type-2 commands and IMS Connect z/OS commands that perform similar functions as the VIEWRMT command.

VIEWRMT command	Equivalent IMS Connect type-2 command	Equivalent IMS Connect z/OS command
VIEWRMT ALL	QUERY IMSCON TYPE(RMTIMSCON) NAME(*) SHOW(ALL | *show_parm*)	QUERY RMTIMSCON NAME(*)
VIEWRMT *rmtimscon_name*	QUERY IMSCON TYPE(RMTIMSCON) NAME(*rmtimscon_name*) SHOW(ALL | *show_parm*)	QUERY RMTIMSCON NAME(*rmtimscon_name*)

Output fields

Output fields that return only a single value are displayed in the format `FIELD_LABEL=VALUE`, for example: `RMTIMSCON=ICON1`.

Output fields that return multiple values, such as the output fields for multiple sockets on a connection, are displayed in a column format, where the field label is a column heading and the values appear under the heading on separate rows in the column. For example:

```
HWSC0001I      SENDCLNT USERID   STATUS SECOND SENDPORT
HWSC0001I      OTM11111 USER01   CONN      100 1111
HWSC0001I      OTM22222 USER01   CONN       89 2222
```

The following list describes the output fields of the VIEWRMT command.

RMTIMSCON=
Identifies a connection between this local IMS Connect instance and a remote instance of IMS Connect. The following additional output fields describe the status, attributes, and sockets that belong to the connection identified in the RMTIMSCON output field:

STATUS
The state of the connection displayed in the RMTIMSCON output field. A connection can be in the following states:

ACTIVE
This connection has at least one send socket open to a remote IMS Connect instance.

NOT ACTIVE
This connection does not have any send sockets open to a remote IMS Connect instance.

RETRY CONN
IMS Connect is attempting to automatically reconnect to the remote IMS Connect instance after a previous attempt to connect failed. If the current reconnection attempt fails, IMS Connect attempts to reconnect again every 2 minutes.

While a connection has a RETRY CONN status, the number of open sockets fluctuates. IMS Connect opens a socket when actively trying to reconnect. IMS Connect closes the socket during the 2-minute interval between reconnection attempts.

STOPPED
A STOPRMT command stopped communication on this connection. Any messages sent to this connection are rejected and an error is sent back to the sender.

IP-ADDRESS
The IP address of the IMS Connect instance at the remote end of this connection.

PORT The port number used by the remote IMS Connect instance for this connection.

HOSTNAME
The host name of the remote IMS Connect instance, if available.

AUTOCONN
Indicates whether IMS Connect automatically establishes send sockets for OTMA communications on this connection during IMS Connect startup. Possible values are:

Y Send sockets are automatically established on this connection when IMS Connect starts up. The number of sockets that IMS Connect establishes during startup is displayed in the RESVSOC output field

N No sockets are established on this connection during IMS Connect startup. For OTMA messages, send sockets are established upon receipt of a message from OTMA for delivery to the remote IMS system.

PERSISTENT
Indicates whether the sockets on this connection are persistent.

Y The sockets established on this connection are persistent.

N The sockets established on this connection are not persistent. After messages are sent on this connection, the sockets that they are sent on are closed.

IDLETO
The time interval that open sockets on this connection can remain idle before they are closed by IMS Connect.

RESVSOC
The number of send sockets that are reserved for use by this connection.

NUMSOC
The number of sockets that are currently open on this connection.

SENDCLNT
The randomly generated send socket ID. The local instance of IMS Connect uses the ID to uniquely identify the socket. The send socket ID is used as the client ID when establishing a socket connection with the remote IMS Connect instance.

The send socket IDs that are used for OTMA begin with "OTM." The send socket IDs that are used for MSC begin with "MSC."

The values returned for this output field are displayed in tabular format.

If no sockets are currently open on this connection, this output field is not displayed.

LCLPLKID
The ID of a local MSC physical link that is using this connection. This value is specified on the LCLPLKID parameter of the MSC statement in the IMS Connect configuration member. The LCLPLKID output field is displayed only for connections that are used for MSC communications.

The values returned for this output field are displayed in tabular format.

If no sockets are currently open on this connection, this output field is not displayed.

USERID
The user ID provided by the local IMS system for transaction authorization by the remote IMS system. The USERID output field is displayed only for connections that are used for OTMA communications.

If no sockets are currently open on this connection, this output field is not displayed.

The values returned for this output field are displayed in tabular format.

STATUS
The current state of this socket connection. A socket connection can have the following states:

CONN
This socket connection can send messages to the remote IMS Connect.

RECV This socket connection is waiting to receive a response from the remote IMS Connect.

The values returned for this output field are displayed in tabular format.

If no sockets are currently open on this connection, this output field is not displayed.

SECOND
The number of seconds that this connection has been in the state shown on the same row under the STATUS output field heading.

The values returned for this output field are displayed in tabular format.

If no sockets are currently open on this connection, this output field is not displayed.

SENDPORT
The local port used for this connection.

The values returned for this output field are displayed in tabular format.

If no sockets are currently open on this connection, this output field is not displayed.

Examples

The following example shows a VIEWRMT command that returns information about all connections to remote IMS Connect instances:

```
nnVIEWRMT ALL
```

The following example shows the output produced by a VIEWRMT command:

```
HWSC0001I    RMTIMSCON=CONNECT2 STATUS=ACTIVE
HWSC0001I      IP-ADDRESS=127.0.0.1       PORT=5555
HWSC0001I      HOSTNAME=ICON2.EXAMPLE.COM
HWSC0001I      AUTOCONN=N PERSISTENT=Y
HWSC0001I      IDLETO=6000
HWSC0001I      RESVSOC=10        NUMSOC=2
HWSC0001I      SENDCLNT LCLPLKID STATUS SECOND SENDPORT
HWSC0001I      MSC11111 MSC12    CONN      100 1234
HWSC0001I      MSC22222 MSC12    CONN       89 5678
HWSC0001I    RMTIMSCON=CONNECT3 STATUS=ACTIVE
HWSC0001I      IP-ADDRESS=192.0.2.0       PORT=9999
HWSC0001I      HOSTNAME=ICON3.EXAMPLE.COM
HWSC0001I      AUTOCONN=Y PERSISTENT=Y
HWSC0001I      IDLETO=6000
HWSC0001I      RESVSOC=4         NUMSOC=4
HWSC0001I      SENDCLNT USERID   STATUS SECOND SENDPORT
HWSC0001I      OTM11111 USER01   CONN      100 1111
HWSC0001I      OTM22222 USER01   CONN       89 2222
HWSC0001I      OTM33333 USER02   CONN       81 3333
HWSC0001I      OTM44444 USER03   RECV       23 4444
```

Related reference:

"STOPSCLN command" on page 481

QUERY IMSCON TYPE(RMTIMSCON) command (Commands)

"IMS Connect QUERY RMTIMSCON command" on page 581

VIEWUOR command

Use the VIEWUOR command to display the current status of a specific unit of recovery identifier (URID) or all URIDs in IMS Connect.

Subsections:

- "Syntax" on page 531
- "Keywords" on page 531
- "Equivalent IMS Connect type-2 and z/OS commands" on page 531
- "Output fields" on page 531

Syntax

Keywords

The following keywords are valid for the VIEWUOR command:

nn 2-digit number of z/OS system console outstanding display.

***urid* | ALL**
Specifies the status of a specific unit of recovery identifier or all unit recovery identifiers to be displayed.

Equivalent IMS Connect type-2 and z/OS commands

The following table lists IMS Connect type-2 commands and IMS Connect z/OS commands that perform similar functions as the VIEWUOR command.

VIEWUOR command	Equivalent IMS Connect type-2 command	Equivalent IMS Connect z/OS command
VIEWUOR ALL	QUERY IMSCON TYPE(UOR) NAME(*) SHOW(ALL | *show_parm*)	QUERY UOR NAME(*) SHOW(ALL)
VIEWUOR *uor_id*	QUERY IMSCON TYPE(UOR) NAME(*uor_id*) SHOW(ALL | *show_parm*)	QUERY UOR NAME(*uor_id*) SHOW(ALL)

Output fields

Information displayed includes:

URID Identifies the 16-byte character string of a specific unit-of-recovery identifier.

STATE
State of the UOR. The UOR state can be one of the following values:

- IN_RESET - The UOR is starting and has not yet changed any resources.
- IN_FLIGHT - The UOR can access resources and has the potential to change resources, but the changes are not committed.
- IN_STATE_CHECK - The UOR issues a commit and waits for the resource manager's STATE_CHECK exit routine to check if the resources are in the correct state.
- IN_PREPARE - The UOR in the proper state issues a commit and z/OS Resource Recovery Services (RRS) invokes the PREPARE exit routine.
- IN_DOUBT - RRS is waiting for the resource manager to tell it whether to resolve the UOR by a commit or by a backout.
- IN_COMMIT - One of the following actions occurred:
  - The PREPARE exit routines replied YES.
  - The DSRM or SDSRM told RRS to commit an IN_DOUBT UOR.
  - The installation used the RRS panels to commit an IN_DOUBT UOR.
- IN_BACKOUT - One of the following actions occurred:
  - One or more PREPARE exit routines replied NO.

- The application issued a backout.
- The DSRM or SDSRM told RRS to back out an IN_DOUBT UOR.
- The installation used the RRS panels to back out an IN_DOUBT UOR.
- Before phase 2 of the two-phase-commit protocol, the system, application, RRS, or a resource manager failed.

- `IN_END` - The resources have been updated.
- `IN_ONLY_AGENT` - Only one resource manager expressed interest in the UOR.
- `IN_COMPLETION` - The resources have been updated and RRS has completed processing the UOR.
- `IN_FORGET` - During distributed processing, the UOR has completed but RRS is waiting for the SDSRM to indicate how long to process the log records for the UOR.
- `FORGOTTEN` - The UOR has completed, and RRS has deleted its log records.

XID (X/Open identifier) Identifies the distributed transaction used by the X/Open architecture. The XID is comprised of four parts:

- `FMID` - 4-byte fixed format id
- `GTRID` - 4-byte fixed GTRID length
- `BQUAL` - 4-byte fixed BQUAL length
- `XID` - 128-byte character XID

TOTAL UOR
The total number of all UORs in any state.

INDOUBT
The total number of UORs in IN_DOUBT state.

INBACKOUT
The total number of UORs in IN_BACKOUT state.

INCOMMIT
The total number of UORs in IN_COMMIT state.

OTHER
The total number of UORs in other states.

Related reference:

QUERY IMSCON TYPE(UOR) command (Commands)

"IMS Connect QUERY UOR command" on page 585

Chapter 24. IMS Connect z/OS commands

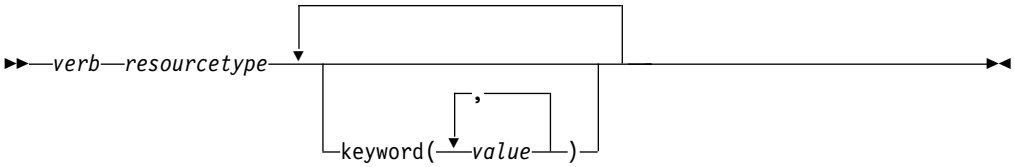
The IMS Connect z/OS commands can be used and issued through the z/OS (MVS) interface. The z/OS modify interface enables you to direct commands to IMS Connect using only the IMS Connect *jobname*.

Related reference:

Equivalent IMS Connect WTOR, z/OS, and type-2 commands (Commands)

IMS Connect z/OS command syntax

IMS Connect supports the verb-resourcetype syntax. The verb-resourcetype syntax consists of a verb, a resource type, and zero or more keyword value pairs, with the values enclosed in parentheses.



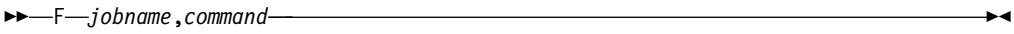
verb A command verb that represents an action. Verb examples include QUERY, SHUTDOWN, UPDATE, and DELETE.

resourcetype
The type of resource that is operated on by the verb. Some resource examples are DATASTORE, PORT, MEMBER, and UOR.

keyword(*value*)
One or more optional keywords and values that represent attributes, filters, or other modifiers that apply to the command. For example, NAME() to identify the specific resources or SET() to set an option.

IMS Connect z/OS invocation

Some IMS Connect commands can be issued through the z/OS MODIFY command interface. The following syntax diagram illustrates the general syntax for entering commands through the MODIFY command interface.



F The z/OS modify command.

jobname
The jobname of the address space to which the command is directed.

command
The command being issued.

© Copyright IBM Corp. 1974, 2017

IMS Connect wildcard character support

Some parameters on IMS Connect commands support wildcard characters for pattern matching. For such parameters, you can use the following wildcard parameters:

`*` Matches zero or more characters

`%` Matches exactly one character

The following examples illustrate some uses of wildcard characters.

`CO*`
Matches any string of any length that begins with "CO". For instance: CO, COO, COOP.

`%%S`
Matches any three-character string that ends with "S". For instance: IMS, CQS.

IMS Connect DELETE CLIENT command

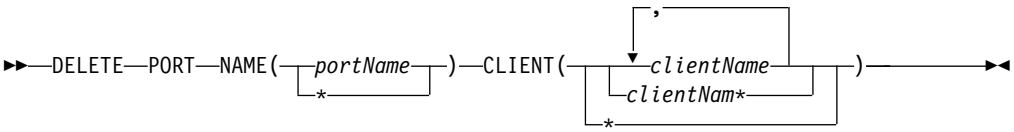
The IMS Connect DELETE CLIENT command deletes a client resource type.

This command performs functions similar to the STOPCLNT command.

Subsections:

- "Syntax"
- "Keywords"
- "Equivalent IMS Connect type-2 and WTOR commands" on page 535
- "Examples" on page 535

Syntax



Keywords

DELETE PORT is used to delete the client within the specified port name. DELETE or DEL is used to delete the client of the specified resource, which in this case is PORT. PORT is an IMS Connect managed resource.

`NAME ( )`
Specifies the name of the port where you want the client to be deleted. You can specify name(*) for the command to be processed for all ports.

`CLIENT ( )`
Specifies the name of the client, a wildcard, or a list of client names which you want to be deleted. You can specify client(*) for the command to be processed for all clients.

Equivalent IMS Connect type-2 and WTOR commands

The following table lists IMS Connect type-2 commands and IMS Connect WTOR commands that perform similar functions as the DELETE CLIENT command.

DELETE CLIENT command	Equivalent IMS Connect type-2 command	Equivalent WTOR command
DELETE PORT NAME(*portName*) CLIENT(*clientName*)	UPDATE IMSCON TYPE(CLIENT) NAME(*client_name*) PORT(*portid*) STOP(COMM)	STOPCLNT *portid clientid*

Examples

Command input:

```
F HWS01,DEL PORT NAME(9999) CLIENT(CLIENT01)
```

Command output:

```
HWSS0761I TCPIP COMMUNICATION WITH CLIENT=9999_CLIENT01 STOPPED; M=SCCM
```

Explanation: The client, `CLIENT01`, within port number 9999 is deleted.

Related reference:

"STOPCLNT command" on page 469

UPDATE IMSCON TYPE(CLIENT) command (Commands)

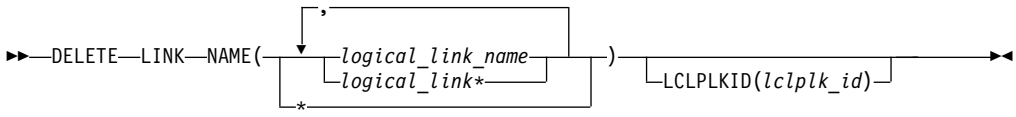
IMS Connect DELETE LINK command

Use the z/OS MODIFY command DELETE LINK to stop communications on an MSC logical link that is assigned to an MSC physical link in IMS Connect.

Subsections:

- "Syntax"
- "Keywords"
- "Usage" on page 536
- "Equivalent IMS Connect type-2 and WTOR commands" on page 537
- "Examples" on page 537

Syntax



Keywords

The command DELETE LINK is used to stop an MSC logical link. DELETE or DEL is used to delete or terminate the specified resource, LINK. LINK represents an MSC logical link that is managed by IMS Connect.

NAME( )

The name of the MSC logical link to be stopped. You can specify a single link name, a wildcard name, or a list of link names separated by commas. You can

specify NAME(*) for the command to be processed for all MSC logical links. If the DELETE LINK command is executed against two or more physical links, all logical links that use the specified logical link name are stopped.

logical_link_name
The MSC logical link to be stopped.

LCLPLKID
Restricts the processing of this command to the MSC physical link that has the specified LCLPLKID. This keyword is optional.

If an LCLPLKID value is not specified, the DELETE LINK command stops all matching logical link names on all MSC physical links that are defined to IMS Connect.

lclplk_id
The ID of an MSC physical link, as specified on the LCLPLKID parameter of the MSC statement in the IMS Connect configuration member.

Usage

Use the DELETE LINK command to clean up the resources associated with an MSC logical link when the link has already been terminated, but the IMS Connect resources associated with the link were not cleaned up correctly.

Recommendation: Use the IMS command /PSTOP to terminate MSC logical links. Use the DELETE LINK command only when the IMS Connect resources associated with an MSC logical link that has already been terminated have failed to clean up correctly.

When the DELETE LINK command is issued, IMS Connect:

- Stops communication on the MSC logical link
- Informs IMS that communication has stopped on the logical link
- Deletes the control blocks associated with the logical link and frees the associated storage
- Issues message HWSF3310I

To avoid accidentally stopping a logical link that uses the same logical link name on another physical link, limit the command processing to a specific physical link by specifying the name of the target physical link as the *lclplk_id* value.

If the DELETE LINK command is issued against two or more physical links, or the *lclplk_id* value is omitted, IMS Connect stops communication on all MSC logical links that match the *logical_link_name* specified on the DELETE LINK command.

To display information about the MSC logical links that are assigned to the MSC physical links that are defined to an IMS Connect instance, use any of the following commands:

- QUERY IMSCON TYPE(LINK) IMS type-2 command for IMS Connect
- QUERY IMSCON TYPE(MSC) IMS type-2 command for IMS Connect
- VIEWMSC WTOR command for IMS Connect
- QUERY MSC z/OS modify command for IMS Connect

Equivalent IMS Connect type-2 and WTOR commands

The following table lists IMS Connect type-2 commands and IMS Connect WTOR commands that perform similar functions as the DELETE LINK command.

DELETE LINK command	Equivalent IMS Connect type-2 command	Equivalent WTOR command
DELETE LINK NAME(*linkName*)	UPDATE IMSCON TYPE(LINK) NAME(*link_name*) STOP(COMM)	STOPLINK *linkName*
DELETE LINK NAME(*linkname*) LCLPLKID(*lclPlkid*)	UPDATE IMSCON TYPE(LINK) NAME(*linkname*) MSC(*lclPlkid*) STOP(COMM)	STOPLINK *linkname lclPlkid*

Examples

Command input:

```
F ICON1,DELETE LINK NAME(MSCLINK1)
```

Command output:

```
HWSF3310I LOGICAL LINK MSCLINK1 TERMINATED; MSC=MSC12, M=modc
```

Related reference:

"STOPLINK command" on page 474

UPDATE IMSCON TYPE(LINK) command (Commands)

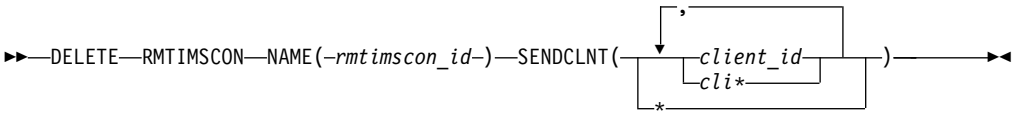
IMS Connect DELETE SENDCLNT command

When IMS Connect is used for sending OTMA messages to a remote IMS system, use the DELETE SENDCLNT command on a local IMS Connect instance to terminate the send sockets on the connection to the remote IMS Connect instance. DELETE SENDCLNT is a z/OS MODIFY command that performs similar functions as the WTOR command STOPSCLN.

Subsections:

- "Syntax"
- "Keywords"
- "Usage" on page 538
- "Equivalent IMS Connect type-2 and WTOR commands" on page 538
- "Examples" on page 539

Syntax



Keywords

The DELETE keyword is used to terminate a specified IMS Connect resource. The SENDCLNT keyword, which represents one or more specific send sockets on an IMS Connect to IMS Connect connection that is used for OTMA messages, is a

resource managed by IMS Connect. The command DELETE SENDCLNT is used to terminate one or more specified OTMA send sockets that are managed by IMS Connect.

NAME(*rmtimscon_id*)
The ID of the IMS Connect to IMS Connect connection on which the OTMA send sockets are open. The value specified for *rmtimscon_id* must match a value specified on the ID parameter of a RMTIMSCON statement of the IMS Connect instance specified on the MVS MODIFY command.

SENDCLNT(*sendclnt_id*)
The ID of the send socket to be terminated. You can specify a single send socket ID, the leading characters of a send socket ID with a wildcard character (an asterisk, *), or a list of send socket IDs separated by commas. You can also specify * by itself to terminate all send sockets on the specified IMS Connect to IMS Connect connection.

Usage

Use the DELETE SENDCLNT command to terminate send sockets on an IMS Connect to IMS Connect connection that is used to send messages to a remote IMS system.

To identify the IDs of the send sockets that you need to terminate, use any one of the following IMS Connect commands:

- In the IMS type-2 command format, QUERY IMSCON TYPE(SENDCLNT)
- In the WTOR command format, VIEWRMT
- In the z/OS MODIFY command format, QUERY RMTIMSCON

For MSC links, use the DELETE SENDCLNT command only when a send client does not clean up properly during MSC logical link termination. Otherwise, issuing the DELETE SENDCLNT command breaks MSC links. Under normal circumstances, stop MSC links by issuing the IMS command /PSTOP LINK from IMS, which automatically terminates the associated send and receive sockets in IMS Connect.

An alternative method of terminating send sockets that are used for MSC messages is by using the IMS Connect WTOR command STOPLINK or the IMS Connect z/OS MODIFY command DELETE LINK. These commands terminate the send sockets and receive sockets used by the MSC link. The IMS type-2 command equivalent is UPDATE IMSCON TYPE(LINK).

If a send socket connection is in RECV state when the DELETE SENDCLNT command is issued, IMS Connect issues a NAK to OTMA and then terminates the send socket connection. OTMA reroutes the message to the dead letter queue, HWS$DLQ.

Equivalent IMS Connect type-2 and WTOR commands

The following table lists IMS Connect type-2 commands and IMS Connect WTOR commands that perform similar functions as the DELETE SENDCLNT command.

DELETE SENDCLNT command	Equivalent IMS Connect type-2 command	Equivalent WTOR command
DELETE RMTIMSCON NAME(*rmtimsconname*) SENDCLNT(*clientid*)	UPDATE IMSCON TYPE(SENDCLNT) NAME(*sendclient_name*) RMTIMSCON(*rmtimscon_name*) STOP(COMM)	STOPSCLN *rmtimscon sendclient*

Examples

In the following example, the DELETE SENDCLNT command terminates the send socket OTM11111 on connection CONNECT2:

```
F ICON1,DELETE RMTIMSCON NAME(CONNECT2) SENDCLNT(OTM11111)
```

Command output:

```
HWST3525I THE SEND CLIENT OTM11111 TERMINATED FOR REMOTE IMS CONNECT CONNECT2; M=TCVC
```

Related reference:

"STOPSCLN command" on page 481

"VIEWRMT command" on page 526

"IMS Connect QUERY RMTIMSCON command" on page 581

UPDATE IMSCON TYPE(SENDCLNT) command (Commands)

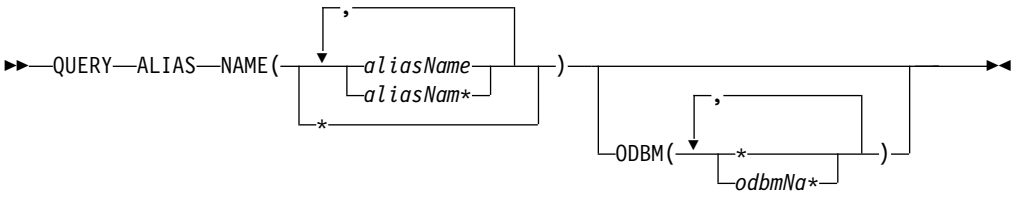
IMS Connect QUERY ALIAS command

Use the QUERY ALIAS command to display the requested IMS alias. The QUERY ALIAS command performs similar functions as the VIEWIA command.

Subsections:

- "Syntax"
- "Keywords"
- "Equivalent IMS Connect type-2 and WTOR commands" on page 540
- "Examples" on page 540

Syntax



Keywords

QUERY or QRY is used to display the status or attributes of a specified resource. ALIAS is an IMS Connect managed specified resource. The command QUERY ALIAS is used to display the current status of the requested alias.

NAME( )

Specifies the name of the alias to be displayed. You can specify a single alias, a

wildcard name, or a list of aliases separated by commas. You can specify NAME(*) for the command to be processed for all aliases.

aliasName
Specifies the alias to be displayed.

ODBM( )
Specifies the ODBM name with which the named alias is associated.

Equivalent IMS Connect type-2 and WTOR commands

The following table lists IMS Connect type-2 commands and IMS Connect WTOR commands that perform similar functions as the QUERY ALIAS command.

QUERY ALIAS command	Equivalent IMS Connect type-2 command	Equivalent WTOR command
QUERY ALIAS NAME(*)	QUERY IMSCON TYPE(ALIAS) NAME(*) SHOW(ALL | *show_parm*)	VIEWIA ALL
QUERY ALIAS NAME(*aliasName*)	QUERY IMSCON TYPE(ALIAS) NAME(*alias_name*) SHOW(ALL | *show_parm*)	VIEWIA *alias_name*
QUERY ALIAS NAME(*aliasName*) ODBM(*odbmName*)	QUERY IMSCON TYPE(ALIAS) NAME(*alias_name*) ODBM(*odbm_name*)	VIEWIA *alias_name odbm_name*

Examples

Command input:

```
F HWS1,QRY ALIAS NAME(ALI1)
```

Command output:

```
HWSC0001I    ALIAS=IMS1  ODBM=ODBM1OD  STATUS=ACTIVE
```

Explanation: The alias IMS1 associated with ODBM ODBM1OD is active.

Related reference:

"VIEWIA command" on page 512

QUERY IMSCON TYPE(ALIAS) command (Commands)

IMS Connect QUERY DATASTORE command

Use the QUERY DATASTORE command to display the data store status.

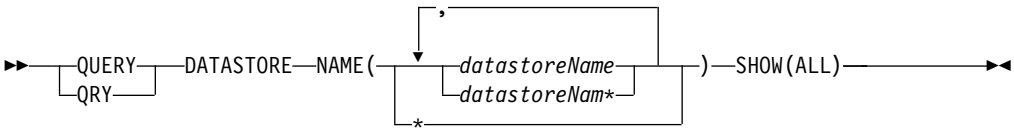
QUERY or QRY is used to query the status or attributes of a specified resource. For example, the specified resource is DATASTORE which is an IMS Connect managed resource.

The data store resource type refers to the destination that is accessing IMS OTMA. With the data store resource type, IMS Connect can communicate with IMS through IMS OTMA. If a data store name is used, this name must match the ID parameter of a DATASTORE configuration statement of the HWSCFGxx configuration member and only the information for this data store is displayed or updated. QUERY DATASTORE performs similar functions as the VIEWDS command.

Subsections:

- "Syntax"
- "Keywords"
- "Equivalent IMS Connect type-2 and WTOR commands"
- "Output fields"
- "Examples" on page 544

Syntax



Keywords

NAME(*datastoreName***)**
Specifies the data store name to be displayed. You can specify a single data store name, a wildcard name, or a list of data store names separated by commas. You can specify name(*) for the command to be processed for all data stores.

SHOW( )
Specifies the output fields to be returned.

ALL
Returns all output fields. This is the default.

Equivalent IMS Connect type-2 and WTOR commands

The following table lists IMS Connect type-2 commands and IMS Connect WTOR commands that perform similar functions as the QUERY DATASTORE command.

QUERY DATASTORE command	Equivalent IMS Connect type-2 command	Equivalent WTOR command
QUERY DATASTORE NAME(*) SHOW(ALL)	QUERY IMSCON TYPE(DATASTORE) NAME(*) SHOW(ALL|*show_parm*)	VIEWDS ALL
QUERY DATASTORE NAME(*datastore_name*)	QUERY IMSCON TYPE(DATASTORE) NAME(*datastore_name*) SHOW(ALL|*show_parm*)	VIEWDS *datastore_name*

Output fields

The following list shows the output fields of the QUERY DATASTORE command.

DATASTORE=
Name of the data store, as defined in the ID substatement of the DATASTORE configuration statement in the HWSCFGxx configuration member, or No active Datastores. The data store is an IMS system or, more precisely, a connection to an IMS system.

STATUS
State of the data store: ACTIVE, NOT ACTIVE or DISCONNECT.

If the data store goes down, IMS Connect is notified (from IMS OTMA through z/OS cross-system coupling facility (XCF)) of the status of the data store. When the data store is brought back up and restarted, IMS Connect is notified and automatically reconnects to the data store.

GROUP
XCF group name for the group to which the IMS Connect and IMS OTMA belong.

MEMBER
IMS Connect member name in the XCF group listed.

TARGET MEMBER
IMS OTMA member name in the XCF group listed.

STATE
Indicates the state of the OTMA server. The STATE value can have one of the following values:

AVAIL
Indicates that the OTMA server is available.

WARN
Indicates that the OTMA server has one or more resources in warning state.

SEVERE
Indicates that the OTMA server is experiencing some severe resource issues.

N/A Indicates that OTMA has not reported status for this data store.

DEFAULT REROUTE NAME
The name of an alternative destination specified in a client reroute request.

RACF APPL NAME
The RACF APPL name for PassTicket and data store access control.

MULTIRTP
Shows the current specification of the MULTIRTP parameter in the DATASTORE statement for the currently displayed DATASTORE connection. Possible values are:

Y The client bid requests for this DATASTORE connection request a TPIPE that supports multiple active resume TPIPE requests.

N The client bid requests for this DATASTORE connection request a TPIPE that supports only a single active resume TPIPE request.

The client bid requests for this DATASTORE connection do not contain a MULTIRTP specification. OTMA determines whether the TPIPEs for this DATASTORE connection support multiple active resume TPIPE requests.

b A MULTIRTP value is not specified in the DATASTORE statement. TPIPE support for multiple resume tpipe

requests is determined by either the MULTIRTP specification in the HWS statement or in the OTMA client descriptor.

CASCADE
Shows the specification in the data store definition for support for cascading global RRS transactions to IMS TM data stores that are located on different LPARs than IMS Connect.

This specification was set either by the CASCADE= parameter of the DATASTORE= statement in the HWSCFGxx member of the IMS PROCLIB data set or by the SET(CASCADE() keyword on type-2 command CREATE IMSCON TYPE(DATASTORE) or UPDATE IMSCON TYPE(DATASTORE).

The value shown here overrides the IMS Connect default value for CASCADE.

One of the following values is displayed:

blank A blank value indicates that cascaded transaction support was not specified for this data store. Support for this data store is determined by the IMS Connect default specification.

Y Indicates that when IMS Connect and the IMS data store are on different z/OS images (LPARs), this data store connection supports cascading global IMS TM transactions.

N Indicates that by default, this data store connection does not support cascading global IMS TM transactions.

OTMA ACEE AGING VALUE
The current OTMA accessor environment element (ACEE) aging value specified in the IMS Connect DATASTORE configuration statement. The OTMA ACEE aging value determines how frequently OTMA refreshes the security definitions that are cached in an ACEE for IMS Connect.

OTMA ACK TIMEOUT VALUE
The OTMA ACK timeout value in the IMS Connect configuration member.

OTMA MAX INPUT MESSAGE
The OTMA input message flood control value in the IMS Connect configuration file.

SUPER MEMBER NAME
The name of the OTMA super member to which this data store belongs.

IMSPLEX=
Name of the IMSplex as defined in the TMEMBER parameter of the IMSplex statement in the IMS Connect configuration member HWSCFGxx.

STATUS
The state of the IMSplex.

MEMBER
IMS Connect member name as defined in the IMS Connect configuration file IMSplex statement for MEMBER= parameter.

TARGET
IMS Connect target member name as defined in the IMS Connect configuration file IMSplex statement for TMEMBER= parameter.

ODBM=
Identifies an instance of Open Database Manager (ODBM) present in the IMSplex.

STATUS
Shows whether this instance of IMS Connect is registered with the identified instance of ODBM. STATUS can have one of the following values:

- `REGISTERED` - IMS Connect is registered with the identified instance of ODBM.
- `NOT ACTIVE` - The identified instance of ODBM is not available.

ODBMRRS
Indicates the RRS specification in an ODBM. When IMS Connect registers with ODBM, ODBM provides IMS Connect with its RRS setting in addition to the aliases.

ALIAS
The alias name of an IMS data store defined to the instance of ODBM identified in the preceding ODBM output field.

STATUS
Shows whether the IMS system associated with the preceding alias name is active or inactive. The status can have one of the following values:

- `ACTIVE` - IMS Connect can route messages to the identified alias.
- `NOT ACTIVE` - IMS Connect cannot route messages to the identified alias.

Examples

Example 1 for QUERY DATASTORE command

Command input:

```
F HWS1, QRY DATASTORE NAME(*) SHOW(ALL)
```

Command output:

```
HWSC0001I  DATASTORE=IMS1     STATUS=ACTIVE
HWSC0001I    GROUP=XCFGRP1  MEMBER=HWS1
HWSC0001I    TARGET MEMBER=IMS1           STATE=AVAIL
HWSC0001I    DEFAULT REROUTE NAME=CFG$DEF
HWSC0001I    RACF APPL NAME=APPLID1       MULTIRTP=Y   CASCADE=Y
HWSC0001I    OTMA ACEE AGING VALUE=2147483647
HWSC0001I    OTMA ACK TIMEOUT VALUE=120
HWSC0001I    OTMA MAX INPUT MESSAGE=5000
HWSC0001I    SUPER MEMBER NAME=SM01  CM0 ACK TOQ=
HWSC0001I  ODBM=ODBMAOD  STATUS=REGISTERED   ODBMRRS=Y
HWSC0001I    ALIAS=IMS1  STATUS=ACTIVE, TRACE
HWSC0001I    ALIAS=IMS2  STATUS=NOT ACTIVE(IMSCON)
HWSC0001I  ODBM=ODBMBOD  STATUS=REGISTERED ODBMRRS=N
HWSC0001I    ALIAS=IMS1  STATUS=ACTIVE
HWSC0001I    ALIAS=IMS2  STATUS=ACTIVE
HWSC0001I  ODBM=ODBMCOD  STATUS=NOT ACTIVE
HWSC0001I  ODBM=ODBMDOD  STATUS=NOT ACTIVE
```

Explanation: The status of all the data stores defined to IMS Connect is displayed.

Example 2 for QUERY DATASTORE command

Command input:

```
F HWS1, QRY DATASTORE NAME(IMS1) SHOW(ALL)
```

Command output:

```
HWSC0001I  DATASTORE=IMS1     STATUS=ACTIVE
HWSC0001I    GROUP=XCFGRP1  MEMBER=HWS1
HWSC0001I    TARGET MEMBER=IMS1           STATE=AVAIL
HWSC0001I    DEFAULT REROUTE NAME=CFG$DEF
HWSC0001I    RACF APPL NAME=APPLID1       MULTIRTP=Y   CASCADE=Y
HWSC0001I    OTMA ACEE AGING VALUE=2147483647
HWSC0001I    OTMA ACK TIMEOUT VALUE=120
HWSC0001I    OTMA MAX INPUT MESSAGE=5000
HWSC0001I    SUPER MEMBER NAME=SM01  CM0 ACK TOQ=
```

Explanation: The status of a single IMS Connect data store, IMS1, is displayed.

Related reference:

"VIEWDS command" on page 483

QUERY IMSCON TYPE(DATASTORE) command (Commands)

IMS Connect QUERY IMSPLEX command

Use the QUERY IMSPLEX command to display the IMSplex status.

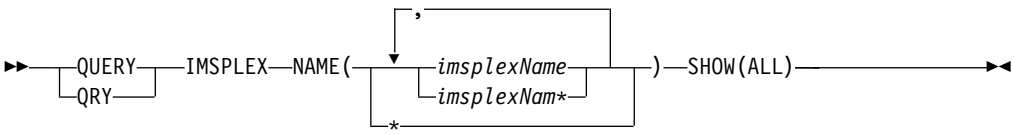
QUERY or QRY is used to query the status or attributes of a specified resource. For example, the specified resource is IMSplex, which is an IMS Connect managed resource.

The IMSplex resource type refers to the name of the SCI to which IMS Connect communicates. If an IMSplex name is used, this name must match the TMEMBER parameter of an IMSPLEX configuration statement of the HWSCFGxx configuration file, and only the information for this IMSplex is displayed or updated. QUERY IMSPLEX performs similar functions as the VIEWIP command.

Subsections:

- "Syntax"
- "Keywords" on page 546
- "Equivalent IMS Connect type-2 and WTOR commands" on page 546
- "Output fields" on page 546
- "Examples" on page 546

Syntax



Keywords

NAME(*imsplexName***)**
Specifies the IMSplex name to be displayed. You can specify a single IMSplex name, a wildcard name, or a list of IMSplex names separated by commas. You can specify NAME(*) for the command to be processed for all IMSplexes.

SHOW( )
Specifies the output fields to be returned.

ALL
Returns all output fields. This is the default.

Equivalent IMS Connect type-2 and WTOR commands

The following table lists IMS Connect type-2 commands and IMS Connect WTOR commands that perform similar functions as the QUERY IMSPLEX command.

QUERY IMSPLEX command	Equivalent IMS Connect type-2 command	Equivalent WTOR command
QUERY IMSPLEX NAME(*) SHOW(ALL)	QUERY IMSCON TYPE(IMSPLEX) NAME(*) SHOW(ALL|*show_parm*)	VIEWIP ALL
QUERY IMSPLEX NAME(*imsplexName*) SHOW(ALL)	QUERY IMSCON TYPE(IMSPLEX) NAME(*IMSplex_name*) SHOW(ALL|*show_parm*)	VIEWIP *IMSplex_name*

Output fields

The following list shows the output fields of the QUERY IMSPLEX command.

IMSPLEX=
Name of the IMSplex as defined in the TMEMBER parameter of the IMSplex statement in the IMS Connect configuration member HWSCFGxx.

STATUS
The state of the IMSplex.

MEMBER
IMS Connect member name as defined in the IMS Connect configuration file IMSplex statement for MEMBER= parameter.

TARGET
IMS Connect target member name as defined in the IMS Connect configuration file IMSplex statement for TMEMBER= parameter.

Examples

Example 1 for QUERY IMSPLEX command

Command input:

```
F HWS1, QRY IMSPLEX NAME(PLEX1) SHOW(ALL)
```

Command output:

```
HWSC0001I    IMSPLEX=PLEX1    STATUS=ACTIVE
HWSC0001I      MEMBER=HWS1     TARGET=PLEX1
```

Explanation: The status of the IMS Connect IMSplex, PLEX1, is displayed.

Related reference:

"VIEWIP command" on page 515

QUERY IMSCON TYPE(IMSPLEX) command (Commands)

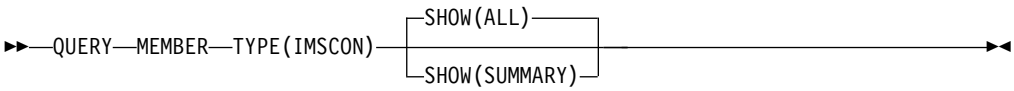
IMS Connect QUERY MEMBER command

This command is used to display the status of IMS Connect. QUERY MEMBER performs similar functions as the VIEWHWS command.

Subsections:

- "Syntax"
- "Keywords"
- "Equivalent IMS Connect type-2 and WTOR commands"
- "Output fields" on page 548
- "Example 1: QUERY MEMBER" on page 563

Syntax



Keywords

QUERY or QRY is used to query the status or attributes of a specified resource and MEMBER is an IMS Connect managed specified resource. The QUERY MEMBER command is used to display the status of IMS Connect.

TYPE ( )
Specifies the target type for action.

IMSCON
Specifies that IMS Connect is the target type.

SHOW ( )
Specifies the target type to be returned.

ALL
Returns all output fields. This is the default.

SUMMARY
Excludes detailed client information.

Usage notes

There are two commands named QUERY MEMBER. QUERY MEMBER TYPE(IMS) is a type-2 command that gets information about IMS members from TSO SPOC or the OM API. QUERY MEMBER TYPE(IMSCON) is a z/OS modify command that gets information about IMS Connect members. See QUERY MEMBER command (Commands) for more information.

Equivalent IMS Connect type-2 and WTOR commands

The following table lists IMS Connect type-2 commands and IMS Connect WTOR commands that perform similar functions as the QUERY MEMBER command.

QUERY MEMBER command	Equivalent IMS Connect type-2 command	Equivalent WTOR command
QUERY MEMBER TYPE(IMSCON) SHOW(ALL)	QUERY IMSCON TYPE(CONFIG) SHOW(ALL|*show_parm*)	VIEWHWS

Output fields

The following list shows the output fields of the QUERY MEMBER command.

HWS ID=
Name of the IMS Connect, as defined in the ID substatement of the HWS configuration statement in the `HWSCFGxx` configuration member.

RACF Indicates whether RACF support is currently enabled for the instance of IMS Connect identified in the HWS ID= field.

UIDCACHE
Indicates whether the RACF user ID cache is enabled for the instance of IMS Connect identified in the HWS ID= field.

UIDAGE
The length of time in seconds that a RACF user ID is used before IMS Connect refreshes it.

PSWDMC
Indicates whether mixed-case password support is currently enabled for the instance of IMS Connect identified in the HWS ID= field. “R” indicates that the IMS Connect support for mixed-case passwords depends on the mixed-case password specification in RACF.

MAXSOC
The maximum total number of sockets that this instance of IMS Connect can open. The maximum number of physical connections that can be made is the MAXSOC= value less the number of ports, because IMS Connect uses one socket on each port for listening.

TCPIPQ
The maximum number of queued requests that IMS Connect can maintain without assigned sockets.

TIMEOUT
The interval of time IMS Connect waits before disconnecting a client application program after either waiting for a response for the client from IMS or waiting for data from the client after the client opens a socket connection.

NUMSOC
The sum of all active sockets on all defined ports, RMTIMSCON connections, and RMTCICS connections.

WARNSOC
A decimal value between 1 and 50 that indicates the warning level as a percentage of the max sockets limit (MAXSOC= value in the TCPIP configuration statement). When the number of sockets increases to this warning level, IMS Connect issues an HWSS0772W message.

WARNINC
A decimal value between 1 and 50 that indicates the warning incremental percentage. After the warning level (WARNSOC value in the TCPIP configuration statement) is reached, each time the number of sockets increases by the warning incremental percentage, IMS Connect issues an HWSS0772W message again.

RRS Shows if RRS is set to Y or N in the HWS configuration file.

STATUS
State of RRS. The RRS state can be one of the following:

- `ACTIVE` - IMS Connect restart with RRS is complete.
- `NOT ACTIVE` - IMS Connect is not registered with RRS.
- `REGISTERED` - IMS Connect is registered with RRS.

MULTIRTP
Shows the current specification of the MULTIRTP parameter in the HWS statement. Possible values are:

Y MULTIRTP=Y is the default value for all DATASTORE connections from this IMS Connect instance. Unless this value is overridden by the MULTIRTP parameter in the DATASTORE statement, the client bid requests for DATASTORE connections made from this IMS Connect instance will request TPIPEs that support multiple active resume TPIPE requests.

N MULTIRTP=N is the default value for all DATASTORE connections from this IMS Connect instance. Unless this value is overridden by the MULTIRTP parameter in the DATASTORE statement, the client bid requests for DATASTORE connections made from this IMS Connect instance will request TPIPEs that support only a single active resume TPIPE request.

b A MULTIRTP value is not specified in the HWS statement. TPIPE support for multiple resume tpipe requests is determined by either the MULTIRTP specification in the DATASTORE statement or in the OTMA client descriptor.

CASCADE

Shows the IMS Connect default specification for support for cascading global RRS transactions to IMS TM data stores that are located on different LPARs than IMS Connect. This specification was set either by the CASCADE= parameter of the HWS= statement in the HWSCFGxx member of the IMS PROCLIB data set or by the type-2 UPDATE IMSCON TYPE(CONFIG) SET(CASCADE()) command.

The value shown here can be overridden in the definition of an individual data store connection.

One of the following values is displayed:

Y Indicates that when IMS Connect and the IMS data store are on different z/OS images (LPARs), IMS Connect enables cascaded transaction support for all data store connections that do not have a different CASCADE value in their definitions.

N Indicates that by default, cascaded transaction support is disabled for data store connections that do not have a different CASCADE value in their definitions.

VERSION
The version of this instance of IMS Connect.

IP-ADDRESS
The IP address for this instance of IMS Connect. Client application programs use this IP address to connect to IMS Connect.

SUPER MEMBER NAME
1-4 character OTMA super member name.

ADAPTER
Shows whether XML adapter support is enabled or disabled.

MAXCVRT
The maximum number of XML converters that this instance of IMS Connect can load concurrently. The actual number of converters that IMS Connect can load depends on the size of the converters and the IMS Connect region size.

NUMCVRT
The current number of XML converters loaded in this instance of IMS Connect.

MAXLSSSZ
Displays the maximum language structure segment size that is passed to the XML converter.

ODBM AUTO CONNECTION
Indicates whether the instance of IMS Connect identified in the ID field is configured to automatically connect to the instances of ODBM in the IMSplex in which IMS Connect is a member.

ODBM TIMEOUT
The time interval that IMS Connect waits before disconnecting a client application program that uses the Open Database architecture.

ODBM IMSPLEX
The names of IMS Connect and SCI that are used for communication with ODBM within an IMSplex.

MEMBER
The name of IMS Connect that SCI uses to manage communications between IMS Connect and ODBM.

TARGET MEMBER
The name of the SCI that manages the communication between IMS Connect and ODBM.

DATASTORE=
Name of the data store, as defined in the ID substatement of the DATASTORE configuration statement in the `HWSCFGxx` configuration member, or `No active Datastores`. The data store is an IMS system or, more precisely, a connection to an IMS system.

STATUS
State of the data store: ACTIVE, NOT ACTIVE or DISCONNECT.

If the data store goes down, IMS Connect is notified (from IMS OTMA through z/OS cross-system coupling facility (XCF)) of the

status of the data store. When the data store is brought back up and restarted, IMS Connect is notified and automatically reconnects to the data store.

GROUP
XCF group name for the group to which the IMS Connect and IMS OTMA belong.

MEMBER
IMS Connect member name in the XCF group listed.

TARGET MEMBER
IMS OTMA member name in the XCF group listed.

STATE
Indicates the state of the OTMA server. The STATE value can have one of the following values:

AVAIL
Indicates that the OTMA server is available.

WARN
Indicates that the OTMA server has one or more resources in warning state.

SEVERE
Indicates that the OTMA server is experiencing some severe resource issues.

N/A Indicates that OTMA has not reported status for this data store.

DEFAULT REROUTE NAME
The name of an alternative destination specified in a client reroute request.

RACF APPL NAME
The RACF APPL name for PassTicket and data store access control.

MULTIRTP
Shows the current specification of the MULTIRTP parameter in the DATASTORE statement for the currently displayed DATASTORE connection. Possible values are:

Y The client bid requests for this DATASTORE connection request a TPIPE that supports multiple active resume TPIPE requests.

N The client bid requests for this DATASTORE connection request a TPIPE that supports only a single active resume TPIPE request.

The client bid requests for this DATASTORE connection do not contain a MULTIRTP specification. OTMA determines whether the TPIPEs for this DATASTORE connection support multiple active resume TPIPE requests.

b A MULTIRTP value is not specified in the DATASTORE statement. TPIPE support for multiple resume tpipe requests is determined by either the MULTIRTP specification in the HWS statement or in the OTMA client descriptor.

CASCADE
Shows the specification in the data store definition for support for cascading global RRS transactions to IMS TM data stores that are located on different LPARs than IMS Connect.

This specification was set either by the CASCADE= parameter of the DATASTORE= statement in the HWSCFGxx member of the IMS PROCLIB data set or by the SET(CASCADE() keyword on type-2 command CREATE IMSCON TYPE(DATASTORE) or UPDATE IMSCON TYPE(DATASTORE).

The value shown here overrides the IMS Connect default value for CASCADE.

One of the following values is displayed:

blank A blank value indicates that cascaded transaction support was not specified for this data store. Support for this data store is determined by the IMS Connect default specification.

Y Indicates that when IMS Connect and the IMS data store are on different z/OS images (LPARs), this data store connection supports cascading global IMS TM transactions.

N Indicates that by default, this data store connection does not support cascading global IMS TM transactions.

OTMA ACEE AGING VALUE
The current OTMA accessor environment element (ACEE) aging value specified in the IMS Connect DATASTORE configuration statement. The OTMA ACEE aging value determines how frequently OTMA refreshes the security definitions that are cached in an ACEE for IMS Connect.

OTMA ACK TIMEOUT VALUE
The OTMA ACK timeout value in the IMS Connect configuration member.

OTMA MAX INPUT MESSAGE
The OTMA input message flood control value in the IMS Connect configuration file.

SUPER MEMBER NAME
The name of the OTMA super member to which this data store belongs.

IMSPLEX=
Name of the IMSplex as defined in the TMEMBER parameter of the IMSplex statement in the IMS Connect configuration member HWSCFGxx.

STATUS
The state of the IMSplex.

MEMBER
IMS Connect member name as defined in the IMS Connect configuration file IMSplex statement for MEMBER= parameter.

TARGET
IMS Connect target member name as defined in the IMS Connect configuration file IMSplex statement for TMEMBER= parameter.

ODBM=
Identifies an instance of Open Database Manager (ODBM) present in the IMSplex.

STATUS
Shows whether this instance of IMS Connect is registered with the identified instance of ODBM. STATUS can have one of the following values:

- `REGISTERED` - IMS Connect is registered with the identified instance of ODBM.
- `NOT ACTIVE` - The identified instance of ODBM is not available.

ODBMRRS
Indicates the RRS specification in an ODBM. When IMS Connect registers with ODBM, ODBM provides IMS Connect with its RRS setting in addition to the aliases.

ALIAS
The alias name of an IMS data store defined to the instance of ODBM identified in the preceding ODBM output field.

STATUS
Shows whether the IMS system associated with the preceding alias name is active or inactive. The status can have one of the following values:

- `ACTIVE` - IMS Connect can route messages to the identified alias.
- `NOT ACTIVE` - IMS Connect cannot route messages to the identified alias.

MSC= Name of the displayed MSC physical link, as specified on the LCLPLKID parameter of the MSC statement in the IMS Connect configuration member. The following additional output fields describe the status, attributes, and logical links that belong to the physical link identified in the MSC= output field:

STATUS
The state of the MSC physical link displayed in the MSC output field. A physical link can be in the following states:

ACTIVE
The MSC physical link is enabled for communications between IMS Connect and the local IMS system.

NOT ACTIVE
The MSC physical link is disabled for communications between IMS Connect and the local IMS system.

RMTPLKID
The ID of the MSC physical link at the remote IMS site, as defined in the RMTPLKID parameter of the MSC statement in the IMS Connect configuration member.

LCLIMSID
The ID of the local MSC-enabled IMS system, as defined to IMS Connect on the LCLIMS parameter of the MSC statement in the IMS Connect configuration member. This LCLIMS value must match the IMS ID that the local IMS system registers with SCI in

the IMSplex. You can determine the IMS ID that is registered with SCI by issuing the QUERY IMSPLEX command in the local IMSplex.

If the second IMS ID is shown in the output, it identifies the other IMS system in an XRF couple.

RMTIMSID
The name of the remote MSC-enabled IMS system, as defined on the RMTIMS parameter of the MSC statement in the IMS Connect configuration member.

GENIMSID
The generic IMS ID that is being used for TCP/IP generic resources for MSC, as defined in the GENIMSID parameter of the MSC statement in the IMS Connect configuration member.

AFFINITY
The name of the IMS system with which the MSC logical link currently has affinity for TCP/IP generic resources.

IMSPLEX
Name of the IMSplex that manages communications between IMS Connect and the local, MSC-enabled IMS system. This name is the IMSplex name that is specified on the IMSPLEX parameter of MSC statement in the IMS Connect configuration member.

This IMSPLEX name must match the name specified on the IMSPLEX(NAME=) statement of the CSLSIxxx PROCLIB member of the SCI instance that is managing communications between IMS Connect and the IMSplex.

MEMBER
The name by which this IMS Connect is registered in the IMSplex, as defined on the MEMBER parameter in the IMSPLEX substatement of the MSC statement in the IMS Connect configuration member.

This MEMBER name must also be specified on the LCLICON parameter in the MSPLINK macro that defines this MSC physical link in the local IMS system.

TARGET MEMBER
Name of the IMSplex as defined in the TMEMBER subparameter of the IMSPLEX parameter of the MSC statement in the IMS Connect configuration member.

RMTIMSCON
The name of the IMS Connect connection that is used by this MSC physical link, as defined in the RMTIMSCON parameter of the MSC statement in the IMS Connect configuration member.

IP-ADDRESS
The IP address of the remote IMS Connect instance.

PORT The port number used by the remote IMS Connect instance.

HOSTNAME
The host name of the remote IMS Connect instance, if one is available.

LINK The ID of an MSC logical link that is assigned to this physical link.

The values returned for this output field are displayed in column format.

If no logical links are currently assigned to this physical link, this output field is not displayed.

PARTNERID

The ID of the remote partner for this logical link, as specified on the PARTNER parameter in the MSLINK macro that defines this logical link.

The values returned for this output field are displayed in column format.

If no logical links are currently assigned to this physical link, this output field is not displayed.

STATUS

The current state of this logical link. A logical link can have the following state:

ACTIVE

This logical link can send messages to the remote IMS Connect.

The values returned for this output field are displayed in column format.

If no logical links are currently assigned to this physical link, this output field is not displayed.

SENDCLNT

The send socket ID generated by this instance of IMS Connect to identify a send socket opened on this connection. The SENDCLNT ID is used as the client ID when establishing a socket connection with the remote IMS Connect instance.

The send socket IDs generated for sockets on connections that are used for OTMA begin with "OTM." The send socket IDs that are generated for sockets on connections that are used for MSC begin with "MSC."

The values returned for this output field are displayed in column format.

If no logical links are currently assigned to this physical link, this output field is not displayed.

RECVCLNT

The client ID that identifies the receive socket for this logical link. The client ID is generated by the remote IMS Connect instance.

The values returned for this output field are displayed in column format.

If no logical links are currently assigned to this physical link, this output field is not displayed.

ISC= ID of the ISC configuration statement in the IMS Connect configuration member that defines the ISC link to which the following output fields apply.

STATUS
Status of this ISC link. Possible values are:

ACTIVE
This ISC link is enabled for communications between IMS Connect and the local IMS system.

NOT ACTIVE
This ISC link is disabled for communications between IMS Connect and the local IMS system.

NODE
The name of this ISC node as defined to the local IMS.

LCLIMSID
Name of the local IMS as defined in the LCLIMS parameter of the ISC statement in the IMS Connect configuration member.

IMSPLEX=

Name of the IMSplex as defined in the IMSPLEX parameter of ISC statement in the IMS Connect configuration member. If IMSPLEX is not specified in the ISC statement, the name from the first IMSPLEX statement in the configuration is displayed.

MEMBER=
Name of the member as defined in the MEMBER subparameter of the IMSPLEX parameter of the ISC statement in the IMS Connect configuration member. This is the IMS Connect member name registered to the IMSplex.

TARGET MEMBER=
Name of the IMSplex as defined in the TMEMBER subparameter of the IMSPLEX parameter of the ISC statement in the IMS Connect configuration member.

RMTCICS
Name of the remote CICS connection that this ISC link is using. The value shown here is the value specified on the RMTCICS parameter of the ISC statement in the IMS Connect configuration member.

IP-ADDRESS
The IP address of this remote CICS.

PORT The port of this remote CICS.

HOSTNAME
The hostname of this remote CICS. This field is displayed if HOSTNAME parameter is specified in the RMTCICS statement.

CICSAPPL
The value specified on the APPLID= parameter of the DFHSIT macro definition in the remote CICS subsystem.

CICSPORT
The ID of the port that IMS Connect uses to receive messages from CICS on this ISC link. The CICSPORT is defined on the TCPIP configuration statement.

CICSNETID
The network ID of the remote CICS subsystem.

ISCUSER STATUS RECVCLNT SENDCLNT

If one or more ISC parallel sessions (ISCUSERs) are active on this ISC link, the command output contains information about each ISCUSER. The information for multiple ISCUSERs is displayed in a column format under a heading that identifies the information type. The information for each ISCUSER is displayed on its own row.

ISCUSER
Name of the ISC parallel session.

STATUS
Status of this ISCUSER.

SENDCLNT
Name of the client ID that IMS Connect uses to send messages to the remote CICS subsystem for this ISCUSER.

RECVCLNT
Name of the client ID of the remote CICS subsystem that this IMS Connect receives messages from for this ISCUSER.

RMTCICS=
Name of the remote CICS connection as defined in the ID parameter of the RMTCICS configuration statement in the IMS Connect configuration member.

STATUS=
State of the connection to the remote CICS subsystem that is identified in the RMTCICS field. Possible values include:

ACTIVE
This IMS Connect has at least one socket connection to the remote CICS subsystem identified in the RMTCICS field.

NOT ACTIVE
This IMS Connect does not have any socket connections to the remote CICS identified in the RMTCICS field.

STOPPED
Communications between this IMS Connect and the remote CICS subsystem is stopped. Any messages sent to the remote subsystem while the connection is stopped are rejected and an error message is returned to the sender.

IP-ADDRESS
The IP address of the remote CICS.

PORT The port number of the remote CICS that this IMS Connect instance communicates on.

HOSTNAME
The host name of the remote CICS or remote IMS Connect, if available.

NUMSOC
The number of sockets that are currently open.

SENDCLNT ISCNAME STATUS SECOND SENDPORT
If this instance of IMS Connect has one or more active connections

to this remote CICS subsystem, the command output includes information about each connection. The information for an active connection is displayed immediately under the heading that describes the value.

The headings are described in the next list of fields.

SENDCLNT
The client ID that this instance of IMS Connect used to connect to the remote CICS.

ISCNAME
Name of the ISC configuration statement in the IMS Connect configuration member.

STATUS
State of this remote IMS Connect connection. Possible values include:

CONN
This connection is in connect state. It can send messages to the remote CICS.

RECV This connection is in receive state. It is waiting to receive a response from the remote CICS.

SECOND
Number of seconds that this connection is in the specified state.

SENDPORT
The port that this instance of IMS Connect used to connect to the remote CICS.

RMTIMSCON=
Identifies a connection between this local IMS Connect instance and a remote instance of IMS Connect. The following additional output fields describe the status, attributes, and sockets that belong to the connection identified in the RMTIMSCON output field:

STATUS
The state of the connection displayed in the RMTIMSCON output field. A connection can be in the following states:

ACTIVE
This connection has at least one send socket open to a remote IMS Connect instance.

NOT ACTIVE
This connection does not have any send sockets open to a remote IMS Connect instance.

RETRY CONN
IMS Connect is attempting to automatically reconnect to the remote IMS Connect instance after a previous attempt to connect failed. If the current reconnection attempt fails, IMS Connect attempts to reconnect again every 2 minutes.

While a connection has a RETRY CONN status, the number of open sockets fluctuates. IMS Connect opens a socket when actively trying to reconnect. IMS Connect closes the socket during the 2-minute interval between reconnection attempts.

STOPPED
A STOPRMT command stopped communication on this connection. Any messages sent to this connection are rejected and an error is sent back to the sender.

IP-ADDRESS
The IP address of the IMS Connect instance at the remote end of this connection.

PORT The port number used by the remote IMS Connect instance for this connection.

HOSTNAME
The host name of the remote IMS Connect instance, if available.

AUTOCONN
Indicates whether IMS Connect automatically establishes send sockets for OTMA communications on this connection during IMS Connect startup. Possible values are:

Y Send sockets are automatically established on this connection when IMS Connect starts up. The number of sockets that IMS Connect establishes during startup is displayed in the RESVSOC output field

N No sockets are established on this connection during IMS Connect startup. For OTMA messages, send sockets are established upon receipt of a message from OTMA for delivery to the remote IMS system.

PERSISTENT
Indicates whether the sockets on this connection are persistent.

Y The sockets established on this connection are persistent.

N The sockets established on this connection are not persistent. After messages are sent on this connection, the sockets that they are sent on are closed.

IDLETO
The time interval that open sockets on this connection can remain idle before they are closed by IMS Connect.

RESVSOC
The number of send sockets that are reserved for use by this connection.

NUMSOC
The number of sockets that are currently open on this connection.

SENDCLNT
The randomly generated send socket ID. The local instance of IMS Connect uses the ID to uniquely identify the socket. The send socket ID is used as the client ID when establishing a socket connection with the remote IMS Connect instance.

The send socket IDs that are used for OTMA begin with "OTM." The send socket IDs that are used for MSC begin with "MSC."

The values returned for this output field are displayed in tabular format.

If no sockets are currently open on this connection, this output field is not displayed.

LCLPLKID
The ID of a local MSC physical link that is using this connection. This value is specified on the LCLPLKID parameter of the MSC statement in the IMS Connect configuration member. The LCLPLKID output field is displayed only for connections that are used for MSC communications.

The values returned for this output field are displayed in tabular format.

If no sockets are currently open on this connection, this output field is not displayed.

USERID
The user ID provided by the local IMS system for transaction authorization by the remote IMS system. The USERID output field is displayed only for connections that are used for OTMA communications.

If no sockets are currently open on this connection, this output field is not displayed.

The values returned for this output field are displayed in tabular format.

STATUS
The current state of this socket connection. A socket connection can have the following states:

CONN
This socket connection can send messages to the remote IMS Connect.

RECV This socket connection is waiting to receive a response from the remote IMS Connect.

The values returned for this output field are displayed in tabular format.

If no sockets are currently open on this connection, this output field is not displayed.

SECOND
The number of seconds that this connection has been in the state shown on the same row under the STATUS output field heading.

The values returned for this output field are displayed in tabular format.

If no sockets are currently open on this connection, this output field is not displayed.

SENDPORT
The local port used for this connection.

The values returned for this output field are displayed in tabular format.

If no sockets are currently open on this connection, this output field is not displayed.

PORT=
Identifies the port or ports to which the following output fields and

statuses apply. The port numbers are defined in the IMS Connect configuration member in the IMS.PROCLIB data set.

You can determine the type of a port by the port-type indicator that is appended to the end of the port number. IMS Connect uses the following port-type indicators:

No type indicator
Indicates a port that can be used for IMS TM OTMA clients and IMS-to-IMS MSC or OTMA communications. Ports that do not have a type indicator are defined on either the PORT keyword or the PORTID keyword of the TCPIP statement in the IMS Connect configuration member.

C Indicates an IBM CICS Transaction Server for z/OS port for ISC communications that is defined to IMS Connect on the CICSPORT parameter of the TCPIP and ISC statements. CICS ports are used exclusively by CICS subsystems that access IMS TM via an ISC TCP/IP link.

D Indicates a Distributed Relational Database Architecture (DRDA) port that is defined to IMS Connect on the DRDAPORT parameter. DRDA ports are used exclusively by client application programs that access IMS DB through the Open Database Manager (ODBM).

S Indicates an SSL port that is defined to IMS Connect on the SSLPORT parameter. SSL ports are used only when IMS Connect support for SSL is enabled.

The following output fields apply to the port identified in the preceding PORT output field.

STATUS
State of the port, whether `ACTIVE` or `INACTIVE`.

KEEPAV
The amount of time a connection remains idle before the z/OS TCP/IP layer sends a packet to maintain the connection.

NUMSOC
The number of sockets used on each port.

TIMEOUT
The amount of time IMS Connect waits before terminating a client connection if no messages are received from the client.

CLIENTID ISCNAME STATUS SECOND CLNTPORT IP-ADDRESS
or
CLIENTID LCLPLKID STATUS SECOND CLNTPORT IP-ADDRESS
or
CLIENTID USERID TRANCODE DATASTORE STATUS SECOND CLNTPORT IP-ADDRESS APSB-TOKEN
The ID and information for an active client on a receive socket.

The information for each active client is displayed on a single row. Clients are grouped together by client type under a row of headings that identify the information that is returned for that client type.

For ISC clients, the following information is displayed: `CLIENTID ISCNAME STATUS SECOND CLNTPORT IP-ADDRESS`.

For MSC clients, the following information is displayed: CLIENTID LCLPLKID STATUS SECOND CLNTPORT IP-ADDRESS.

For OTMA and ODBM clients, the following information is displayed: CLIENTID USERID TRANCODE DATASTORE STATUS SECOND CLNTPORT IP-ADDRESS APSB-TOKEN.

The following example shows the information returned for three OTMA clients:

```
HWSC0001I  PORT=9999     STATUS=ACTIVE     KEEPAV=0 NUMSOC=4 EDIT=         TIMEOUT=0
HWSC0001I    CLIENTID USERID   TRANCODE DATASTORE STATUS    SECOND  CLNTPORT IP-ADDRESS       APSB-TOKEN
HWSC0001I    CLIENT63 USRT003  apol12   IMSY      RECV WFCM     55      4118 192.0.2.0
HWSC0001I    CLIENT62 USRT002           IMS1      CONN         210      4115 192.0.2.0
HWSC0001I    CLIENT61 USRT001  apol12   IMSN      RECV          13      4112 192.0.2.0
HWSC0001I    TOTAL CLIENTS=3  RECV=2 CONN=1 XMIT=0 OTHER=0
```

If a port has no active clients on it, these output fields are not displayed. Instead, the command output displays NO ACTIVE CLIENTS after the row that identifies the port.

The information returned for active clients includes, in alphabetical order:

APSB-TOKEN
Open Database Manager (ODBM) APSB token.

CLIENTID
The name of the client.

For connections from other IMS Connect instances, the other IMS Connect instance generates this value.

CLNTPORT
A random number that TCP/IP generates to represent a connection from a client.

DATASTORE
The data store associated with this client.

The IP address of the client for this connection.

If IPV6 is enabled, the IP address format consists of eight hexadecimal numbers divided by colons. If IPV6 is not enabled, the IP address format of IPV4 is used. The following example is for an IPV6 IP address that is displayed in IPV6 format:

```
FEDC:ABCD:2222:3333:FEDC:DB55:6666:3322
```

The following example is for an IPV4 IP address that is displayed in IPV6 format:

```
0:0:0:0:0:FFFF:945:33FF
```

ISCNAME
The name of the ISC statement in the IMS Connect configuration member.

LCLPLKID
The MSC physical link in the local IMS system to which IMS Connect routes the messages from the client. This field exists only for MSC communications.

SECOND
Number of seconds that the client has been in the specified status.

STATUS
State of the thread of the client. The client thread state can be one of the following values:

RECV In a receive state waiting for input from client.

READ In a partial read state. IMS Connect is reading a message from the client but has not yet received the entire length of the message.

CONN
Waiting for output from IMS.

XMIT Sending data to client.

CONV
In a conversational state.

WFCM
Waiting for confirmation (ACK, NAK, or DEALLOCATE) from client.

TRANCODE
The transaction code submitted by the client.

USERID
The USERID name passed to IMS Connect.

Example 1: QUERY MEMBER

The following command queries the status of IMS Connect.

Command input:

```
F HWS1,QRY MEMBER TYPE(IMSCON)
```

Command output:

```
HWSC0001I HWS ID=HWS1     RACF=N  PSWDMC=N
HWSC0001I    UIDCACHE=Y   UIDAGE=2147483647
HWSC0001I    MAXSOC=2000  TIMEOUT=5000 TCPIPQ=50
HWSC0001I    NUMSOC=7     WARNSOC=80%  WARNINC=5%
HWSC0001I    RRS=Y  STATUS=REGISTERED  MULTIRTP=  CASCADE=N
HWSC0001I    VERSION=V14 IP-ADDRESS=127.0.0.1
HWSC0001I    SUPER MEMBER NAME=SM01  CM0 ACK TOQ=

HWSC0001I   ADAPTER=N    MAXCVRT=100  NUMCVRT=0
HWSC0001I    MAXLSSSZ=32767
HWSC0001I  ODBM AUTO CONNECTION=Y
HWSC0001I  ODBM TIMEOUT=18000
HWSC0001I  ODBM IMSPLEX MEMBER=IMSPLEX1 TARGET MEMBER=PLEX1
HWSC0001I  DATASTORE=IMS1     STATUS=ACTIVE
HWSC0001I    GROUP=XCFGRP1  MEMBER=HWS1
HWSC0001I    TARGET MEMBER=IMS1           STATE=AVAIL
HWSC0001I    DEFAULT REROUTE NAME=CFG$DEF
HWSC0001I    RACF APPL NAME=APPLID1       MULTIRTP=  CASCADE=Y
HWSC0001I    OTMA ACEE AGING VALUE=2147483647
HWSC0001I    OTMA ACK TIMEOUT VALUE=120
HWSC0001I    OTMA MAX INPUT MESSAGE=5000
HWSC0001I    SUPER MEMBER NAME=SM01  CM0 ACK TOQ=
HWSC0001I      STATUS=ACTIVE
HWSC0001I    MEMBER=IMSPLEX1 TARGET=PLEX1
HWSC0001I  ODBM=ODBMAOD  STATUS=REGISTERED   ODBMRRS=Y
HWSC0001I    ALIAS=IMS1  STATUS=ACTIVE, TRACE
HWSC0001I    ALIAS=IMS2  STATUS=NOT ACTIVE
HWSC0001I  ODBM=ODBMBOD  STATUS=REGISTERED ODBMRRS=N
HWSC0001I    ALIAS=IMS1  STATUS=ACTIVE
HWSC0001I    ALIAS=IMS2  STATUS=ACTIVE
HWSC0001I  ODBM=ODBMCOD  STATUS=NOT ACTIVE
HWSC0001I  ODBM=ODBMDOD  STATUS=NOT ACTIVE
HWSC0001I  PORT=9999     STATUS=ACTIVE     KEEPAV=0 NUMSOC=4 EDIT=         TIMEOUT=0
HWSC0001I    CLIENTID USERID   TRANCODE DATASTORE STATUS    SECOND  CLNTPORT IP-ADDRESS       APSB-TOKEN
HWSC0001I    CLIENT63 USRT003  apol12   IMSY      RECV WFCM     55      4118 192.0.2.0
HWSC0001I    CLIENT62 USRT002           IMS1      CONN         210      4115 192.0.2.0
HWSC0001I    CLIENT61 USRT001  apol12   IMSN      RECV          13      4112 192.0.2.0
```

```
HWSC0001I    TOTAL CLIENTS=3  RECV=2 CONN=1 XMIT=0 OTHER=0
HWSC0001I  PORT=LOCAL    STATUS=ACTIVE     KEEPAV=0 NUMSOC=0 EDIT=         TIMEOUT=0
HWSC0001I    NO ACTIVE CLIENTS
HWSC0001I  PORT=9998S    STATUS=NOT ACTIVE KEEPAV=0 NUMSOC=1 EDIT=         TIMEOUT=0
HWSC0001I  PORT=1111D    STATUS=ACTIVE     KEEPAV=5 NUMSOC=1 EDIT=         TIMEOUT=6000
HWSC0001I    NO ACTIVE CLIENTS
HWSC0001I  PORT=2222D    STATUS=ACTIVE     KEEPAV=0 NUMSOC=1 EDIT=         TIMEOUT=100
HWSC0001I    NO ACTIVE CLIENTS
```

Explanation: The status of IMS Connect is displayed. The status of each data store and port number is listed.

Example 2: SHOW(SUMMARY)

The following command produces the same output as the QUERY MEMBER command without the SHOW(SUMMARY) option except for the PORT= sections, which give only the total counts for the ports, eliminating individual client displays.

Command input:

```
F HWS1,QRY MEMBER TYPE(IMSCON) SHOW(SUMMARY)
```

Command output:

```
HWSC0001I HWS ID=HWS1     RACF=N  PSWDMC=N
HWSC0001I    UIDCACHE=Y   UIDAGE=2147483647
HWSC0001I    MAXSOC=50    TIMEOUT=5000 TCPIPQ=50
HWSC0001I    NUMSOC=7     WARNSOC=80%  WARNINC=5%
HWSC0001I    RRS=Y   STATUS=REGISTERED MULTIRTP=  CASCADE=Y
HWSC0001I    VERSION=V14 IP-ADDRESS=127.0.0.1
HWSC0001I    SUPER MEMBER NAME=SM01  CM0 ACK TOQ=

HWSC0001I   ADAPTER=N    MAXCVRT=100  NUMCVRT=0
HWSC0001I    MAXLSSSZ=32767
HWSC0001I  ODBM AUTO CONNECTION=Y
HWSC0001I  ODBM TIMEOUT=18000
HWSC0001I  ODBM IMSPLEX MEMBER=IMSPLEX1 TARGET MEMBER=PLEX1
HWSC0001I  DATASTORE=IMS1     STATUS=ACTIVE
HWSC0001I    GROUP=XCFGRP1  MEMBER=HWS1
HWSC0001I    TARGET MEMBER=IMS1           STATE=AVAIL
HWSC0001I    DEFAULT REROUTE NAME=CFG$DEF
HWSC0001I    RACF APPL NAME=APPLID1       MULTIRTP=  CASCADE=
HWSC0001I    OTMA ACEE AGING VALUE=2147483647
HWSC0001I    OTMA ACK TIMEOUT VALUE=120
HWSC0001I    OTMA MAX INPUT MESSAGE=5000
HWSC0001I    SUPER MEMBER NAME=SM01  CM0 ACK TOQ=
HWSC0001I  IMSPLEX=PLEX1    STATUS=ACTIVE
HWSC0001I    MEMBER=IMSPLEX1 TARGET=PLEX1
HWSC0001I  ODBM=ODBMAOD  STATUS=REGISTERED   ODBMRRS=Y
HWSC0001I    ALIAS=IMS1  STATUS=ACTIVE, TRACE
HWSC0001I    ALIAS=IMS2  STATUS=NOT ACTIVE(IMSCON,ODBM)
HWSC0001I  ODBM=ODBMBOD  STATUS=REGISTERED ODBMRRS=N
HWSC0001I    ALIAS=IMS1  STATUS=ACTIVE
HWSC0001I    ALIAS=IMS2  STATUS=ACTIVE
HWSC0001I  ODBM=ODBMCOD  STATUS=NOT ACTIVE
HWSC0001I  ODBM=ODBMDOD  STATUS=NOT ACTIVE
HWSC0001I  PORT=9999     STATUS=ACTIVE     KEEPAV=0 NUMSOC=4 EDIT=         TIMEOUT=0
HWSC0001I    TOTAL CLIENTS=3  RECV=2 CONN=1 XMIT=0 OTHER=0
HWSC0001I  PORT=LOCAL    STATUS=ACTIVE     KEEPAV=0 NUMSOC=0 EDIT=         TIMEOUT=0
HWSC0001I    NO ACTIVE CLIENTS
HWSC0001I  PORT=9998S    STATUS=NOT ACTIVE KEEPAV=0 NUMSOC=1 EDIT=         TIMEOUT=0
HWSC0001I  PORT=1111D    STATUS=ACTIVE     KEEPAV=5 NUMSOC=1 EDIT=         TIMEOUT=6000
HWSC0001I    NO ACTIVE CLIENTS
HWSC0001I  PORT=2222D    STATUS=ACTIVE     KEEPAV=0 NUMSOC=1 EDIT=         TIMEOUT=100
HWSC0001I    NO ACTIVE CLIENTS
```

Explanation: The status of IMS Connection is displayed. Detailed client information for ports is not included.

Example 3: IMS Connect to IMS Connect connections for MSC

If connections between IMS Connect instances are used for MSC, the command output presents different perspectives of the connections, depending on which instance of IMS Connect the command is issued against.

For MSC communications, one instance of IMS Connect has a connection defined to send MSC messages to the other instance of IMS Connect. The other instance of IMS Connect has a connection defined to send the reply messages.

In the following example output, the command was issued for an IMS Connect instance, HWS1, at the IMS installation where the MSC transactions are input to IMS. HWS1 is connected to another IMS Connect instance, HWS2. HWS1 knows HWS2 only by the RMTIMSCON name ICON2. HWS1 has two active client connections for sending MSC transactions to ICON2 for processing, MSCBB435 and MSC84CF7, shown under SENDCLNT in the RMTIMSCON information. HWS1 receives the response messages to the MSC transactions on the client connections MSCC73E0 and MSC0EBB0, shown under CLIENTID in the PORT information.

A DATASTORE statement is defined for this instance of IMS Connect; however, DATASTORE statements are not used for MSC communications and can be omitted if your instance of IMS Connect does not communicate with OTMA.

```
HWSC0001I   HWS ID=HWS1     RACF=N  PSWDMC=R
HWSC0001I      UIDCACHE=N   UIDAGE=2147483647
HWSC0001I      MAXSOC=2000  TIMEOUT=5000 TCPIPQ=50
HWSC0001I      NUMSOC=5     WARNSOC=80%  WARNINC=5%
HWSC0001I      RRS=N   STATUS=REGISTERED MULTIRTP=  CASCADE=N
HWSC0001I      VERSION=V14 IP-ADDRESS=192.0.2.0
HWSC0001I      SUPER MEMBER NAME=      CM0 ACK TOQ=

HWSC0001I     ADAPTER=N   MAXCVRT=100  NUMCVRT=0
HWSC0001I      MAXLSSSZ=32767
HWSC0001I    DATASTORE=IMS1     STATUS=DISCONNECT
HWSC0001I      GROUP=XCFGRP1  MEMBER=HWS1
HWSC0001I      TARGET MEMBER=IMS1             STATE=N/A
HWSC0001I      DEFAULT REROUTE NAME=HWS$DEF
HWSC0001I      RACF APPL NAME=APPLID1         MULTIRTP=  CASCADE=
HWSC0001I      OTMA ACEE AGING VALUE=2147483647
HWSC0001I      OTMA ACK TIMEOUT VALUE=120
HWSC0001I      OTMA MAX INPUT MESSAGE=5000
HWSC0001I      SUPER MEMBER NAME=      CM0 ACK TOQ=
HWSC0001I    IMSPLEX=PLEX1    STATUS=ACTIVE
HWSC0001I      MEMBER=HWS1     TARGET=PLEX1
HWSC0001I    MSC=MSC12    STATUS=ACTIVE
HWSC0001I      RMTPLKID=MSC21
HWSC0001I      LCLIMSID=IMS1      RMTIMSID=IMS2
HWSC0001I      IMSPLEX=PLEX1
HWSC0001I        MEMBER=HWS1              TARGET MEMBER=PLEX1
HWSC0001I      RMTIMSCON=ICON2
HWSC0001I        IP-ADDRESS=127.0.0.1        PORT=5555
HWSC0001I        HOSTNAME=WWW.EXAMPLE.COM
HWSC0001I      LINK       PARTNERID   STATUS       SENDCLNT   RECVCLNT
HWSC0001I      LNK12T01   TA          ACTIVE       MSC84CF7   MSC0EBB0
HWSC0001I      LNK12T02   TB          ACTIVE       MSCBB435   MSCC73E0
HWSC0001I    NO ACTIVE ODBM
HWSC0001I    PORT=9999     STATUS=ACTIVE     KEEPAV=0 NUMSOC=3      EDIT=     TIMEOUT=0
HWSC0001I      CLIENTID LCLPLKID  STATUS      SECOND CLNTPORT IP-ADDRESS
HWSC0001I      MSCC73E0 MSC12     RECV             6 1026     0:0:0:0:0:FFFF:7F00:1
HWSC0001I      MSC0EBB0 MSC12     RECV            56 1025     0:0:0:0:0:FFFF:7F00:1
HWSC0001I      TOTAL CLIENTS=2  RECV=2 READ=0 CONN=0 XMIT=0 OTHER=0
```

```
HWSC0001I    RMTIMSCON=ICON2     STATUS=ACTIVE
HWSC0001I      IP-ADDRESS=127.0.0.1        PORT=5555
HWSC0001I      HOSTNAME=WWW.EXAMPLE.COM
HWSC0001I      AUTOCONN=N  PERSISTENT=Y
HWSC0001I      IDLETO=0
HWSC0001I      RESVSOC=2   NUMSOC=2
HWSC0001I      SENDCLNT LCLPLKID STATUS       SECOND SENDPORT
HWSC0001I      MSCBB435 MSC12    CONN              6 1028
HWSC0001I      MSC84CF7 MSC12    CONN             56 1027
HWSC0001I      TOTAL SENDCLNTS=2 RECV=0 CONN=2 XMIT=0 OTHER=0
```

The following example output corresponds to the preceding example output, and was produced by executing the command on IMS Connect instance HWS2, at the remote IMS installation where the MSC transactions are processed. HWS2 knows HWS1 only by the RMTIMSCON name ICON1. HWS2 receives MSC transactions on two active client connections, MSCBB435 and MSC84CF7, shown in the PORT information under CLIENTID. These IDs are the same IDs shown in the HWS1 example under SENDCLNT. After processing the MSC transactions, HWS2 sends the response messages back to ICON1 on the two active client connections shown under SENDCLNT in the RMTIMSCON information: MSCC73E0 and MSC0EBB0, the same IDs shown in the HWS1 example under CLIENTID.

```
HWSC0001I   HWS ID=HWS2     RACF=N  PSWDMC=R
HWSC0001I      UIDCACHE=N   UIDAGE=2147483647
HWSC0001I      MAXSOC=50  TIMEOUT=5000    TCPIPQ=50
HWSC0001I      NUMSOC=5      WARNSOC=80%  WARNINC=5%
HWSC0001I      RRS=N   STATUS=REGISTERED  MULTIRTP=  CASCADE=N
HWSC0001I      VERSION=V14 IP-ADDRESS=127.0.0.1
HWSC0001I      SUPER MEMBER NAME=      CM0 ACK TOQ=

HWSC0001I    ADAPTER=N     MAXCVRT=100   NUMCVRT=0
HWSC0001I      MAXLSSSZ=32767
HWSC0001I    DATASTORE=IMS2     STATUS=DISCONNECT
HWSC0001I      GROUP=XCFGRP1  MEMBER=HWS2
HWSC0001I      TARGET MEMBER=IMS2             STATE=N/A
HWSC0001I      DEFAULT REROUTE NAME=HWS$DEF
HWSC0001I      RACF APPL NAME=APPLID1         MULTIRTP=  CASCADE=
HWSC0001I      OTMA ACEE AGING VALUE=2147483647
HWSC0001I      OTMA ACK TIMEOUT VALUE=120
HWSC0001I      OTMA MAX INPUT MESSAGE=5000
HWSC0001I      SUPER MEMBER NAME=      CM0 ACK TOQ=
HWSC0001I    IMSPLEX=PLEX2    STATUS=ACTIVE
HWSC0001I      MEMBER=HWS2     TARGET=PLEX2
HWSC0001I    MSC=MSC21    STATUS=ACTIVE
HWSC0001I      RMTPLKID=MSC12
HWSC0001I      LCLIMSID=IMS2      RMTIMSID=IMS1
HWSC0001I      IMSPLEX=PLEX2
HWSC0001I        MEMBER=HWS2              TARGET MEMBER=PLEX2
HWSC0001I      RMTIMSCON=ICON1
HWSC0001I        IP-ADDRESS=192.0.2.0        PORT=9999
HWSC0001I        HOSTNAME=WWW.EXAMPLE.COM
HWSC0001I      LINK       PARTNERID   STATUS       SENDCLNT   RECVCLNT
HWSC0001I      LNK21T01   TA          ACTIVE       MSC0EBB0   MSC84CF7
HWSC0001I      LNK21T02   TB          ACTIVE       MSCC73E0   MSCBB435
HWSC0001I    NO ACTIVE ODBM
HWSC0001I    PORT=5555     STATUS=ACTIVE     KEEPAV=0 NUMSOC=3     EDIT=      TIMEOUT=0
HWSC0001I      CLIENTID LCLPLKID  STATUS      SECOND CLNTPORT IP-ADDRESS
HWSC0001I      MSCBB435 MSC21     RECV            18 1028     0:0:0:0:0:FFFF:C000:200
HWSC0001I      MSC84CF7 MSC21     RECV            69 1027     0:0:0:0:0:FFFF:C000:200
HWSC0001I      TOTAL CLIENTS=2  RECV=2 READ=0 CONN=0 XMIT=0 OTHER=0
HWSC0001I    RMTIMSCON=ICON1     STATUS=ACTIVE
HWSC0001I      IP-ADDRESS=192.0.2.0        PORT=9999
HWSC0001I      HOSTNAME=WWW.EXAMPLE.COM
HWSC0001I      AUTOCONN=N  PERSISTENT=Y
HWSC0001I      IDLETO=0
HWSC0001I      RESVSOC=2   NUMSOC=2
HWSC0001I      SENDCLNT LCLPLKID STATUS       SECOND SENDPORT
```

```
HWSC0001I      MSCC73E0 MSC21    CONN             18 1026
HWSC0001I      MSC0EBB0 MSC21    CONN             69 1025
HWSC0001I      TOTAL SENDCLNTS=2 RECV=0 CONN=2 XMIT=0 OTHER=0
```

Example 4: IMS Connect to IMS Connect connections for OTMA

If a connection between IMS Connect instances is used for OTMA messages, the command output shows different perspectives of the connection, depending on which instance of IMS Connect the command is executed on.

For OTMA, because the connections between IMS Connect instances are one-way connections, the connection is defined only in the sending IMS Connect instance. The connection is not defined in the remote IMS Connect instance.

In the following example output, the command is executed on HWS1, the IMS Connect instance at the IMS installation that sends the OTMA output messages. HWS1 is connected to a receiving IMS Connect instance, HWS2; however, HWS1 knows HWS2 only by the RMTIMSCON name ICON2B shown in the example. HWS1 sends the OTMA output messages to ICON2 for processing on the active client connection, OTM924FA, shown under SENDCLNT in the RMTIMSCON information. The SENDCLNT ID is auto-generated by HWS1 when establishing the connection with HWS2.

```
HWSC0001I   HWS ID=HWS1     RACF=N  PSWDMC=R
HWSC0001I      UIDCACHE=N   UIDAGE=2147483647
HWSC0001I      MAXSOC=2000  TIMEOUT=5000  TCPIPQ=50
HWSC0001I      NUMSOC=2      WARNSOC=80%  WARNINC=5%
HWSC0001I      RRS=N   STATUS=REGISTERED  MULTIRTP=  CASCADE=N
HWSC0001I      VERSION=V14 IP-ADDRESS=192.0.2.0
HWSC0001I      SUPER MEMBER NAME=      CM0 ACK TOQ=
HWSC0001I    ADAPTER=N  MAXCVRT=100  NUMCVRT=0
HWSC0001I      MAXLSSSZ=32767
HWSC0001I    DATASTORE=IMS1     STATUS=ACTIVE
HWSC0001I      GROUP=XCFGRP1  MEMBER=HWS1
HWSC0001I      TARGET MEMBER=IMS1             STATE=AVAIL
HWSC0001I      DEFAULT REROUTE NAME=HWS$DEF
HWSC0001I      RACF APPL NAME=APPLID1         MULTIRTP=  CASCADE=
HWSC0001I      OTMA ACEE AGING VALUE=2147483647
HWSC0001I      OTMA ACK TIMEOUT VALUE=120
HWSC0001I      OTMA MAX INPUT MESSAGE=5000
HWSC0001I      SUPER MEMBER NAME=      CM0 ACK TOQ=
HWSC0001I    NO ACTIVE IMSPLEX
HWSC0001I    NO ACTIVE ODBM
HWSC0001I    NO ACTIVE MSC
HWSC0001I    PORT=9999     STATUS=ACTIVE     KEEPAV=0 NUMSOC=1     EDIT=     TIMEOUT=0
HWSC0001I      NO ACTIVE CLIENTS
HWSC0001I    RMTIMSCON=ICON2B    STATUS=ACTIVE
HWSC0001I      IP-ADDRESS=127.0.0.1        PORT=5555
HWSC0001I      HOSTNAME=WWW.EXAMPLE.COM
HWSC0001I      AUTOCONN=N  PERSISTENT=Y
HWSC0001I      IDLETO=3000
HWSC0001I      RESVSOC=4   NUMSOC=1
HWSC0001I      SENDCLNT USERID   STATUS       SECOND SENDPORT
HWSC0001I      OTM924FA APOL1    CONN           5941 1026
HWSC0001I      TOTAL SENDCLNTS=1 RECV=0 CONN=1 XMIT=0 OTHER=0
```

The following example output corresponds to the preceding example output, and was produced by executing the command on HWS2, the IMS Connect instance at the IMS installation that receives the OTMA output messages. HWS2 knows HWS1 only by the auto-generated client ID, OTM924FA, shown under CLIENTID in the PORT information. The CLIENTID value is the same value shown under SENDCLNT at the local IMS installation. HWS2 receives the OTMA transactions on the connection identified by the client ID OTM924FA. After processing the OTMA

message, the remote IMS system queues any response messages to the tpipe hold queue identified by the client ID OTM924FA for later asynchronous retrieval.

```
HWSC0001I   HWS ID=HWS2     RACF=N  PSWDMC=R
HWSC0001I     UIDCACHE=N   UIDAGE=2147483647
HWSC0001I     MAXSOC=50  TIMEOUT=5000  TCPIPQ=50
HWSC0001I     NUMSOC=5      WARNSOC=80%  WARNINC=5%
HWSC0001I     RRS=N   STATUS=REGISTERED  MULTIRTP=  CASCADE=N
HWSC0001I     VERSION=V14 IP-ADDRESS=127.0.0.1
HWSC0001I     SUPER MEMBER NAME=      CM0 ACK TOQ=
HWSC0001I    ADAPTER=N  MAXCVRT=100  NUMCVRT=0
HWSC0001I     MAXLSSSZ=32767
HWSC0001I    DATASTORE=IMS2     STATUS=ACTIVE
HWSC0001I     GROUP=XCFGRP2  MEMBER=HWS2
HWSC0001I     TARGET MEMBER=IMS2             STATE=AVAIL
HWSC0001I     DEFAULT REROUTE NAME=HWS$DEF
HWSC0001I     RACF APPL NAME=APPLID1         MULTIRTP=  CASCADE=
HWSC0001I     OTMA ACEE AGING VALUE=2147483647
HWSC0001I     OTMA ACK TIMEOUT VALUE=120
HWSC0001I     OTMA MAX INPUT MESSAGE=5000
HWSC0001I     SUPER MEMBER NAME=      CM0 ACK TOQ=
HWSC0001I    NO ACTIVE IMSPLEX
HWSC0001I    NO ACTIVE ODBM
HWSC0001I    NO ACTIVE MSC
HWSC0001I    PORT=9999     STATUS=ACTIVE     KEEPAV=0 NUMSOC=1         EDIT=         TIMEOUT=0
HWSC0001I     NO ACTIVE CLIENTS
HWSC0001I    PORT=8888     STATUS=ACTIVE     KEEPAV=0 NUMSOC=1         EDIT=         TIMEOUT=0
HWSC0001I     NO ACTIVE CLIENTS
HWSC0001I    PORT=7777     STATUS=ACTIVE     KEEPAV=0 NUMSOC=1         EDIT=         TIMEOUT=0
HWSC0001I     NO ACTIVE CLIENTS
HWSC0001I    PORT=5555     STATUS=ACTIVE     KEEPAV=0 NUMSOC=2         EDIT=         TIMEOUT=0
HWSC0001I     CLIENTID USERID TRANCODE DATASTORE STATUS SECOND CLNTPORT IP-ADDRESS              APSB-TOKEN
HWSC0001I     OTM924FA APOL1  APOL11   IMS2      RECV       11 1026     0:0:0:0:0:FFFF:C000:200
HWSC0001I     TOTAL CLIENTS=1  RECV=1 READ=0 CONN=0 XMIT=0 OTHER=0
HWSC0001I    RMTIMSCON=ICON1B    STATUS=NOT ACTIVE
HWSC0001I     IP-ADDRESS=192.0.2.0        PORT=9999
HWSC0001I     HOSTNAME=WWW.EXAMPLE.COM
HWSC0001I     AUTOCONN=N  PERSISTENT=Y
HWSC0001I     IDLETO=0
HWSC0001I     RESVSOC=4   NUMSOC=0
HWSC0001I     NO ACTIVE CLIENTS
```

Example 5: ISC connections between IMS Connect and CICS

In the following example output, an ISC link, IMS1CIC4, links to a remote CICS subsystem by way of the RMTCICS connection, CICS4.

Currently, three parallel sessions, or *ISCUSERs*, are active: CIC4U01, CIC4U02, and CIC4U03. The status, the send client ID, and the receive client ID are shown on the row for each parallel session.

IMS Connect receives input messages from CICS on port 9876C, as defined by the CICSPORT keyword in the IMS Connect configuration member. Each receive socket for each parallel session on the CICSPORT is identified by a receive client ID that is generated by IMS Connect.

IMS Connect sends output messages to CICS through send sockets on ports that are automatically selected by IMS Connect. Each send socket for each parallel session is identified by a send client ID that is automatically generated by IMS Connect.

```
HWSC0001I HWS ID=HWS1     RACF=N  PSWDMC=R
HWSC0001I    UIDCACHE=N   UIDAGE=2147483647
HWSC0001I    MAXSOC=50  TIMEOUT=5000  TCPIPQ=50
HWSC0001I    NUMSOC=7      WARNSOC=80%  WARNINC=5%
HWSC0001I    RRS=N   STATUS=NOT ACTIVE  MULTIRTP=  CASCADE=N
```

```
HWSC0001I    VERSION=V14 IP-ADDRESS=127.0.0.1
HWSC0001I    SUPER MEMBER NAME=      CM0 ACK TOQ=

HWSC0001I ADAPTER=N MAXCVRT=100 NUMCVRT=0
HWSC0001I    MAXLSSSZ=32767
HWSC0001I  IMSPLEX=PLEX1    STATUS=ACTIVE
HWSC0001I    MEMBER=HWS1     TARGET=PLEX1
HWSC0001I  ISC=IMS1CIC4 STATUS=ACTIVE
HWSC0001I    NODE=CIC4
HWSC0001I    LCLIMSID=IMS1
HWSC0001I    IMSPLEX=PLEX1
HWSC0001I      MEMBER=HWS1              TARGET MEMBER=PLEX1
HWSC0001I    RMTCICS=CICS4
HWSC0001I      IP-ADDRESS=192.0.2.0        PORT=7777
HWSC0001I      HOSTNAME=WWW.EXAMPLE.COM
HWSC0001I    CICSAPPL=CICS1     CICSPORT=9876
HWSC0001I    CICSNETID=IMSNET
HWSC0001I    ISCUSER    STATUS       SENDCLNT   RECVCLNT
HWSC0001I    CIC4U01    ACTIVE       ISCE7811   ISCF4816
HWSC0001I    CIC4U02    ACTIVE       ISC3A819   ISC7121D
HWSC0001I    CIC4U03    ACTIVE       ISCECD99   ISC5C09F
HWSC0001I    TOTAL ISCUSERS=3 ACTIVE=3 OTHER=0
HWSC0001I  NO ACTIVE ODBM
HWSC0001I  NO ACTIVE MSC
HWSC0001I  PORT=9876C    STATUS=ACTIVE     KEEPAV=0 NUMSOC=4 EDIT=         TIMEOUT=0
HWSC0001I    CLIENTID ISCNAME   STATUS      SECOND CLNTPORT IP-ADDRESS
HWSC0001I    ISC5C09F IMS1CIC4  RECV             4 1037     0:0:0:0:0:FFFF:C000:200
HWSC0001I    ISC7121D IMS1CIC4  RECV            13 1035     0:0:0:0:0:FFFF:C000:200
HWSC0001I    ISCF4816 IMS1CIC4  RECV           200 1033     0:0:0:0:0:FFFF:C000:200
HWSC0001I    TOTAL CLIENTS=3  RECV=3 READ=0 CONN=0 XMIT=0 OTHER=0
HWSC0001I  NO ACTIVE RMTIMSCON
HWSC0001I  RMTCICS=CICS4     STATUS=ACTIVE
HWSC0001I    IP-ADDRESS=192.0.2.0      PORT=7777
HWSC0001I    HOSTNAME=WWW.EXAMPLE.COM
HWSC0001I    SENDCLNT ISCNAME  STATUS       SECOND SENDPORT
HWSC0001I    ISCECD99 IMS1CIC4 CONN             4 1036
HWSC0001I    ISC3A819 IMS1CIC4 CONN            13 1034
HWSC0001I    ISCE7811 IMS1CIC4 CONN           200 1032
HWSC0001I    TOTAL SENDCLNTS=3 RECV=0 CONN=3 XMIT=0 OTHER=0
```

Example 6: Multiple client types

In the following example output, IMS Connect supports many different client types, including:

- Clients that connect to IMS DB via a DRDA port and the Open Database Manager (ODBM) component of the IMS Common Service Layer (CSL).
- Clients that connect to IMS TM via a DATASTORE connection and the IMS Open Transaction Manager Access (OTMA) component.
- A CICS system that connects to IMS TM via ISC and the CICS IPIC protocol
- An IMS system that connects to another IMS system via OTMA
- An IMS system that connects in another IMS system via MSC

Certain output fields are always displayed while other output fields are displayed only if IMS Connect is actively supporting a specific client type.

```
HWSC0001I   HWS ID=ICON1    RACF=N  PSWDMC=R
HWSC0001I      MAXSOC=50  TIMEOUT=5000  TCPIPQ=50
HWSC0001I      NUMSOC=18     WARNSOC=80%  WARNINC=5%
HWSC0001I      RRS=N   STATUS=ACTIVE      MULTIRTP=  CASCADE=N
HWSC0001I      VERSION=V14 IP-ADDRESS=127.0.0.1
HWSC0001I      SUPER MEMBER NAME=      CM0 ACK TOQ=

HWSC0001I    ADAPTER=Y  MAXCVRT=100  NUMCVRT=0
HWSC0001I      MAXLSSSZ=32767
HWSC0001I    ODBM AUTO CONNECTION=Y
```

```
HWSC0001I    ODBM TIMEOUT=1200000
HWSC0001I    ODBM IMSPLEX MEMBER=                TARGET MEMBER=
HWSC0001I    DATASTORE=IMS1     STATUS=ACTIVE
HWSC0001I      GROUP=XCFGRP1  MEMBER=ICON1
HWSC0001I      TARGET MEMBER=IMS1             STATE=N/A
HWSC0001I      DEFAULT REROUTE NAME=THATWAY
HWSC0001I      RACF APPL NAME=APPLID1         MULTIRTP=  CASCADE=
HWSC0001I      OTMA ACEE AGING VALUE=2147483647
HWSC0001I      OTMA ACK TIMEOUT VALUE=120
HWSC0001I      OTMA MAX INPUT MESSAGE=5000
HWSC0001I      SUPER MEMBER NAME=      CM0 ACK TOQ=
HWSC0001I    IMSPLEX=PLEX1    STATUS=ACTIVE
HWSC0001I      MEMBER=ICON1    TARGET=PLEX1
HWSC0001I    ODBM=ODBM1OD  STATUS=REGISTERED   ODBMRRS=Y
HWSC0001I      ALIAS=ALI1     STATUS=ACTIVE
HWSC0001I      ALIAS=IMSA     STATUS=ACTIVE
HWSC0001I      ALIAS=IMS1     STATUS=ACTIVE
HWSC0001I      ALIAS=ALI2     STATUS=ACTIVE
HWSC0001I      ALIAS=IMS2     STATUS=ACTIVE
HWSC0001I    MSC=MSC12    STATUS=ACTIVE
HWSC0001I      RMTPLKID=MSC21
HWSC0001I      LCLIMS=IMS1 RMTIMS=IMS2 GENIMSID=IMS AFFINTY=IMS1
HWSC0001I      IMSPLEX=PLEX1
HWSC0001I        MEMBER=ICON1    TARGET MEMBER=PLEX1
HWSC0001I      RMTIMSCON=CONNECT2
HWSC0001I        IP-ADDRESS=192.0.2.0      PORT=5555
HWSC0001I        HOSTNAME=WWW.EXAMPLE.COM
HWSC0001I      LINK     PARTNERID STATUS     SENDCLNT RECVCLNT
HWSC0001I      LINK12A  AA        ACTIVE     MSC11111 MSC33333
HWSC0001I      LINK12B  BB        ACTIVE     MSC22222 MSC44444
HWSC0001I    ISC=CICS    STATUS=ACTIVE
HWSC0001I      NODE=CICSTRM1
HWSC0001I      LCLIMSID=IMS1
HWSC0001I      IMSPLEX=PLEX1
HWSC0001I        MEMBER=HWS1               TARGET MEMBER=PLEX1
HWSC0001I      RMTCICS=ICONNT1
HWSC0001I        IP-ADDRESS=192.0.2.1        PORT=9999
HWSC0001I        HOSTNAME=WWW.EXAMPLE.COM
HWSC0001I        CICSAPPL=CICSA1   CICSPORT=6666
HWSC0001I        CICSNETID=CICSNET
HWSC0001I      NO ACTIVE ISCUSER
HWSC0001I    ISC=LU6TCP2    STATUS=ACTIVE
HWSC0001I      NODE=CICSTRM2
HWSC0001I      LCLIMSID=IMS1
HWSC0001I      IMSPLEX=PLEX1
HWSC0001I        MEMBER=HWS1               TARGET MEMBER=PLEX1
HWSC0001I      RMTCICS=ICONNT2
HWSC0001I        IP-ADDRESS=192.0.2.2        PORT=9998
HWSC0001I        HOSTNAME=WWW.EXAMPLE.COM
HWSC0001I        CICSAPPL=CICSA2   CICSPORT=7777
HWSC0001I        CICSNETID=CICSNET
HWSC0001I      ISCUSER  STATUS    SENDCLNT RECVCLNT
HWSC0001I      IMSUS01  ACTIVE    ISC11111 ISC33333
HWSC0001I      IMSUS02  ACTIVE    ISC22222 ISC44444
HWSC0001I        TOTAL ISCUSERS=2 ACTIVE=2 OTHER=0
HWSC0001I    PORT=9996     STATUS=ACTIVE     KEEPAV=0 NUMSOC=4 EDIT=         TIMEOUT=0
HWSC0001I      CLIENTID USERID   TRANCODE DATASTORE STATUS  SECOND CLNTPORT IP-ADDRESS            APSB-TOKEN
HWSC0001I      CLIENT01 USRT003  ITOC04   IMS1       RECV        27 3464     0:0:0:0:0:FFFF:C000:204
HWSC0001I      CLIENTID LCLPLKID  STATUS SECOND CLNTPORT IP-ADDRESS
HWSC0001I      MSC33333 MSC12     CONN       14 1739     0:0:0:0:0:FFFF:C000:200
HWSC0001I      MSC44444 MSC12     CONN        9 2684     0:0:0:0:0:FFFF:C000:200
HWSC0001I      TOTAL CLIENTS=3  RECV=1 CONN=2 XMIT=0 OTHER=0
HWSC0001I    PORT=6666C    STATUS=NOT ACTIVE KEEPAV=0 NUMSOC=1 EDIT=         TIMEOUT=0
HWSC0001I      NO ACTIVE CLIENTS
HWSC0001I    PORT=7777C    STATUS=ACTIVE     KEEPAV=0 NUMSOC=3 EDIT=         TIMEOUT=0
HWSC0001I
HWSC0001I      CLIENTID ISCNAME STATUS SECOND CLNTPORT  IP-ADDRESS
HWSC0001I      ISC33333 LU6TCP2   CONN     14     3579  0:0:0:0:0:FFFF:C000:202
HWSC0001I      ISC44444 LU6TCP2   CONN      9     2468  0:0:0:0:0:FFFF:C000:202
HWSC0001I      TOTAL CLIENTS=3  RECV=1 CONN=2 XMIT=0 OTHER=0
```

```
HWSC0001I    PORT=LOCAL    STATUS=ACTIVE     KEEPAV=0 NUMSOC=0 EDIT=         TIMEOUT=0
HWSC0001I      NO ACTIVE CLIENTS
HWSC0001I    PORT=8899S    STATUS=NOT ACTIVE KEEPAV=0 NUMSOC=1 EDIT=         TIMEOUT=0
HWSC0001I      NO ACTIVE CLIENTS
HWSC0001I    PORT=8888D    STATUS=ACTIVE     KEEPAV=0 NUMSOC=1 EDIT=         TIMEOUT=600000
HWSC0001I      NO ACTIVE CLIENTS
HWSC0001I    RMTIMSCON=CONNECT2 STATUS=ACTIVE
HWSC0001I      IP-ADDRESS=192.0.2.0       PORT=5555
HWSC0001I      HOSTNAME=WWW.EXAMPLE.COM
HWSC0001I      AUTOCONN=N PERSISTENT=Y
HWSC0001I      IDLETO=6000
HWSC0001I      RESVSOC=10        NUMSOC=2
HWSC0001I      SENDCLNT LCLPLKID STATUS SECOND SENDPORT
HWSC0001I      MSC11111 MSC12    CONN      100 1234
HWSC0001I      MSC22222 MSC12    CONN       89 5678
HWSC0001I    RMTIMSCON=CONNECT3 STATUS=ACTIVE
HWSC0001I      IP-ADDRESS=192.0.2.4       PORT=9999
HWSC0001I      HOSTNAME=WWW.EXAMPLE.COM
HWSC0001I      AUTOCONN=Y PERSISTENT=Y
HWSC0001I      IDLETO=6000
HWSC0001I      RESVSOC=4         NUMSOC=4
HWSC0001I      SENDCLNT USERID   STATUS SECOND SENDPORT
HWSC0001I      OTM11111 USER01   CONN      100 1111
HWSC0001I      OTM22222 USER01   CONN       89 2222
HWSC0001I      OTM33333 USER02   CONN       81 3333
HWSC0001I      OTM44444 USER03   RECV       23 4444
HWSC0001I      TOTAL SENDCLNTS=4 RECV=1 CONN=3 XMIT=0 OTHER=0
HWSC0001I    RMTCICS=ICONNT1   STATUS=NOT ACTIVE
HWSC0001I      IP-ADDRESS=192.0.2.1        PORT=9997
HWSC0001I      HOSTNAME=WWW.EXAMPLE.COM
HWSC0001I      NO ACTIVE CLIENTS
HWSC0001I    RMTCICS=ICONNT2   STATUS=ACTIVE
HWSC0001I      IP-ADDRESS=192.0.2.2        PORT=9998
HWSC0001I      HOSTNAME=WWW.EXAMPLE.COM
HWSC0001I      SENDCLNT ISCNAME     STATUS SECOND SENDPORT
HWSC0001I      ISC11111 LU6TCP2     CONN      100 1111
HWSC0001I      ISC22222 LU6TCP2     CONN       89 2222
HWSC0001I      TOTAL SENDCLNTS=2 RECV=0 CONN=2 XMIT=0 OTHER=0
```

Related reference:

"VIEWHWS command" on page 488

QUERY IMSCON TYPE(CONFIG) command (Commands)

IMS Connect QUERY MSC command

Use the IMS Connect z/OS MODIFY command QUERY MSC to display the attributes and status of the MSC links that IMS Connect supports.

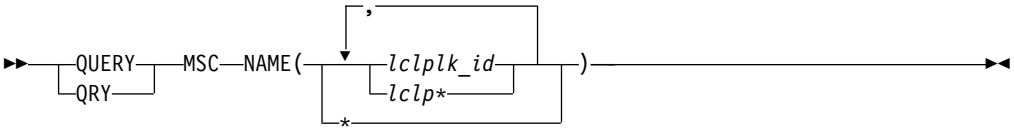
The QUERY MSC command displays the same information as the IMS Connect WTOR command VIEWMSC.

To display information about the sockets that are currently open for MSC on IMS Connect, use the IMS Connect z/OS MODIFY command QUERY RMTIMSCON or the IMS Connect WTOR command VIEWRMT.

Subsections:

- "Syntax" on page 572
- "Keywords" on page 572
- "Usage" on page 572
- "Equivalent IMS Connect type-2 and WTOR commands" on page 572
- "Output fields" on page 572
- "Examples" on page 575

Syntax



Keywords

QUERY or QRY keyword is used to query the status or attributes of a specified resource, such as MSC links, which are an IMS Connect managed resource. The MSC links are represented in the command syntax by the keyword MSC.

NAME(*lclplk_id***)**
The ID of the MSC physical link, as specified LCLPLKID parameter of the MSC statement in the IMS Connect PROCLIB configuration member.

You can specify a single link ID, a wildcard name, or a list of link IDs that are separated by commas. Specify NAME(*) to display information about all the MSC physical links that are defined to the IMS Connect instance.

Usage

Use the QUERY MSC command to display information of the MSC physical link defined in the LCLPLKID parameter of the MSC statement in the IMS Connect configuration member.

Equivalent IMS Connect type-2 and WTOR commands

The following table lists IMS Connect type-2 commands and IMS Connect WTOR commands that perform similar functions as the QUERY MSC command.

QUERY MSC command	Equivalent IMS Connect type-2 command	Equivalent WTOR command
QUERY MSC NAME(*)	QUERY IMSCON TYPE(MSC) NAME(*) SHOW(ALL | *show_parm*)	VIEWMSC ALL
QUERY MSC NAME(*msc_id*)	QUERY IMSCON TYPE(MSC) NAME(*msc_id*) SHOW(ALL | *show_parm*)	VIEWMSC *msc_id*

Output fields

Output fields that return only a single value are displayed in the format `FIELD_LABEL=VALUE`, for example: `RMTIMSCON=ICON1`.

Output fields that return multiple values, such as the output fields for multiple sockets on a connection, are displayed in a column format, where the field label is a column heading and the values appear under the heading on separate rows in the column. For example:

```
HWSC0001I      LINK     PARTNERID STATUS     SENDCLNT RECVCLNT
HWSC0001I      LINK12A  AA        ACTIVE     MSC11111 MSC33333
HWSC0001I      LINK12B  BB        ACTIVE     MSC22222 MSC44444
```

The following list shows the output fields of the QUERY MSC command.

MSC= Name of the displayed MSC physical link, as specified on the LCLPLKID parameter of the MSC statement in the IMS Connect configuration member. The following additional output fields describe the status, attributes, and logical links that belong to the physical link identified in the MSC= output field:

STATUS
The state of the MSC physical link displayed in the MSC output field. A physical link can be in the following states:

ACTIVE
The MSC physical link is enabled for communications between IMS Connect and the local IMS system.

NOT ACTIVE
The MSC physical link is disabled for communications between IMS Connect and the local IMS system.

RMTPLKID
The ID of the MSC physical link at the remote IMS site, as defined in the RMTPLKID parameter of the MSC statement in the IMS Connect configuration member.

LCLIMSID
The ID of the local MSC-enabled IMS system, as defined to IMS Connect on the LCLIMS parameter of the MSC statement in the IMS Connect configuration member. This LCLIMS value must match the IMS ID that the local IMS system registers with SCI in the IMSplex. You can determine the IMS ID that is registered with SCI by issuing the QUERY IMSPLEX command in the local IMSplex.

If the second IMS ID is shown in the output, it identifies the other IMS system in an XRF couple.

RMTIMSID
The name of the remote MSC-enabled IMS system, as defined on the RMTIMS parameter of the MSC statement in the IMS Connect configuration member.

GENIMSID
The generic IMS ID that is being used for TCP/IP generic resources for MSC, as defined in the GENIMSID parameter of the MSC statement in the IMS Connect configuration member.

AFFINITY
The name of the IMS system with which the MSC logical link currently has affinity for TCP/IP generic resources.

IMSPLEX
Name of the IMSplex that manages communications between IMS Connect and the local, MSC-enabled IMS system. This name is the IMSplex name that is specified on the IMSPLEX parameter of MSC statement in the IMS Connect configuration member.

This IMSPLEX name must match the name specified on the IMSPLEX(NAME=) statement of the CSLSIxxx PROCLIB member of the SCI instance that is managing communications between IMS Connect and the IMSplex.

MEMBER
The name by which this IMS Connect is registered in the

IMSplex, as defined on the MEMBER parameter in the IMSPLEX substatement of the MSC statement in the IMS Connect configuration member.

This MEMBER name must also be specified on the LCLICON parameter in the MSPLINK macro that defines this MSC physical link in the local IMS system.

TARGET MEMBER
Name of the IMSplex as defined in the TMEMBER subparameter of the IMSPLEX parameter of the MSC statement in the IMS Connect configuration member.

RMTIMSCON
The name of the IMS Connect connection that is used by this MSC physical link, as defined in the RMTIMSCON parameter of the MSC statement in the IMS Connect configuration member.

IP-ADDRESS
The IP address of the remote IMS Connect instance.

PORT The port number used by the remote IMS Connect instance.

HOSTNAME
The host name of the remote IMS Connect instance, if one is available.

LINK The ID of an MSC logical link that is assigned to this physical link.

The values returned for this output field are displayed in column format.

If no logical links are currently assigned to this physical link, this output field is not displayed.

PARTNERID

The ID of the remote partner for this logical link, as specified on the PARTNER parameter in the MSLINK macro that defines this logical link.

The values returned for this output field are displayed in column format.

If no logical links are currently assigned to this physical link, this output field is not displayed.

STATUS
The current state of this logical link. A logical link can have the following state:

ACTIVE
This logical link can send messages to the remote IMS Connect.

The values returned for this output field are displayed in column format.

If no logical links are currently assigned to this physical link, this output field is not displayed.

SENDCLNT
The send socket ID generated by this instance of IMS Connect to identify a send socket opened on this

connection. The SENDCLNT ID is used as the client ID when establishing a socket connection with the remote IMS Connect instance.

The send socket IDs generated for sockets on connections that are used for OTMA begin with "OTM." The send socket IDs that are generated for sockets on connections that are used for MSC begin with "MSC."

The values returned for this output field are displayed in column format.

If no logical links are currently assigned to this physical link, this output field is not displayed.

RECVCLNT
The client ID that identifies the receive socket for this logical link. The client ID is generated by the remote IMS Connect instance.

The values returned for this output field are displayed in column format.

If no logical links are currently assigned to this physical link, this output field is not displayed.

Examples

In the following example, the QUERY MSC command returns information about the MSC physical link MSC13, which is defined to the local IMS Connect instance HWS1:

```
F ICON1,QUERY MSC NAME(MSC13)

HWSC0001I    MSC=MSC13    STATUS=ACTIVE
HWSC0001I      RMTPLKID=MSC31
HWSC0001I      LCLIMSID=IMS1      RMTIMSID=IMS3
HWSC0001I      GENIMSID=          AFFINITY=
HWSC0001I      IMSPLEX=PLEX1
HWSC0001I        MEMBER=HWS1              TARGET MEMBER=PLEX1
HWSC0001I      RMTIMSCON=ICONNT2
HWSC0001I        IP-ADDRESS=127.0.0.1        PORT=9992
HWSC0001I        HOSTNAME=
HWSC0001I      NO ACTIVE LINK
```

In the following example, the QUERY MSC command returns information about the MSC physical link MSC12:

```
F ICON1,QUERY MSC NAME(MSC12)

HWSC0001I    MSC=MSC12    STATUS=ACTIVE
HWSC0001I      RMTPLKID=MSC21
HWSC0001I      LCLIMSID=IMS1    ,IMS3      RMTIMSID=IMS3
HWSC0001I      GENIMSID=IMS       AFFINITY=
HWSC0001I      IMSPLEX=PLEX1
HWSC0001I        MEMBER=HWS1              TARGET MEMBER=PLEX1
HWSC0001I      RMTIMSCON=ICONNT1
HWSC0001I        IP-ADDRESS=192.0.2.2        PORT=9999
HWSC0001I        HOSTNAME=WWW.EXAMPLE.COM
HWSC0001I      NO ACTIVE LINK
```

Related reference:

"VIEWMSC command" on page 516

QUERY IMSCON TYPE(MSC) command (Commands)

IMS Connect QUERY PORT command

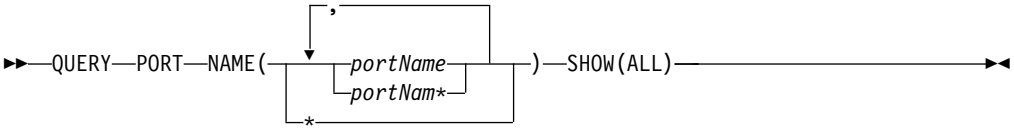
Use the QUERY PORT command to display the current status of a requested port.

The PORT resource type refers to the port number that binds a socket with TCP/IP. If a port name is specified, the name must match a port name already defined in the PORTID, CICSPORT, DRDAPORT, or SSLPORT substatements of the TCP/IP configuration statement in the IMS Connect configuration member. The QUERY PORT command performs similar functions as the VIEWPORT command.

Subsections:

- "Syntax"
- "Keywords"
- "Equivalent IMS Connect type-2 and WTOR commands"
- "Output fields" on page 577
- Examples

Syntax



Keywords

QUERY or QRY is used to query the status or attributes of a specified resource and PORT is an IMS Connect managed specified resource. The QUERY PORT command is used to display the requested port status.

NAME (*portName*)
Specifies the port name to be displayed. You can specify a single port name, a wildcard name, or a list of port names separated by commas. You can specify name(*) for the command to be processed for all ports.

SHOW ( )
Specifies the output fields to be returned.

ALL
Returns all output fields. This is the default.

Usage notes

Use the QUERY PORT command to display the current status, client, and activity of one port or all ports.

The information displayed by QUERY PORT is the same as the information returned for all ports by the QUERY MEMBER command.

Equivalent IMS Connect type-2 and WTOR commands

The following table lists IMS Connect type-2 commands and IMS Connect WTOR commands that perform similar functions as the QUERY PORT command.

QUERY PORT command	Equivalent IMS Connect type-2 command	Equivalent WTOR command
QUERY PORT NAME(*) SHOW(ALL)	QUERY IMSCON TYPE(PORT) NAME(*) SHOW(ALL | *show_parm*)	VIEWPORT ALL
QUERY PORT NAME(*port_id*) SHOW(ALL)	QUERY IMSCON TYPE(PORT) NAME(*portid*) SHOW(ALL | *show_parm*)	VIEWPORT *port_id*
QUERY PORT NAME(LOCAL) SHOW(ALL)	QUERY IMSCON TYPE(PORT) NAME(LOCAL) SHOW(ALL | *show_parm*)	VIEWPORT LOCAL

Output fields

The following list shows the output fields of the QUERY PORT command.

PORT=
Identifies the port or ports to which the following output fields and statuses apply. The port numbers are defined in the IMS Connect configuration member in the IMS.PROCLIB data set.

You can determine the type of a port by the port-type indicator that is appended to the end of the port number. IMS Connect uses the following port-type indicators:

No type indicator
Indicates a port that can be used for IMS TM OTMA clients and IMS-to-IMS MSC or OTMA communications. Ports that do not have a type indicator are defined on either the PORT keyword or the PORTID keyword of the TCPIP statement in the IMS Connect configuration member.

C Indicates an IBM CICS Transaction Server for z/OS port for ISC communications that is defined to IMS Connect on the CICSPORT parameter of the TCPIP and ISC statements. CICS ports are used exclusively by CICS subsystems that access IMS TM via an ISC TCP/IP link.

D Indicates a Distributed Relational Database Architecture (DRDA) port that is defined to IMS Connect on the DRDAPORT parameter. DRDA ports are used exclusively by client application programs that access IMS DB through the Open Database Manager (ODBM).

S Indicates an SSL port that is defined to IMS Connect on the SSLPORT parameter. SSL ports are used only when IMS Connect support for SSL is enabled.

The following output fields apply to the port identified in the preceding PORT output field.

STATUS
State of the port, whether `ACTIVE` or `INACTIVE`.

KEEPAV
The amount of time a connection remains idle before the z/OS TCP/IP layer sends a packet to maintain the connection.

NUMSOC
The number of sockets used on each port.

TIMEOUT
The amount of time IMS Connect waits before terminating a client connection if no messages are received from the client.

CLIENTID ISCNAME STATUS SECOND CLNTPORT IP-ADDRESS
or
CLIENTID LCLPLKID STATUS SECOND CLNTPORT IP-ADDRESS
or
CLIENTID USERID TRANCODE DATASTORE STATUS SECOND CLNTPORT IP-ADDRESS APSB-TOKEN
The ID and information for an active client on a receive socket.

The information for each active client is displayed on a single row. Clients are grouped together by client type under a row of headings that identify the information that is returned for that client type.

For ISC clients, the following information is displayed: `CLIENTID ISCNAME STATUS SECOND CLNTPORT IP-ADDRESS`.

For MSC clients, the following information is displayed: `CLIENTID LCLPLKID STATUS SECOND CLNTPORT IP-ADDRESS`.

For OTMA and ODBM clients, the following information is displayed: `CLIENTID USERID TRANCODE DATASTORE STATUS SECOND CLNTPORT IP-ADDRESS APSB-TOKEN`.

The following example shows the information returned for three OTMA clients:

```
HWSC0001I  PORT=9999     STATUS=ACTIVE     KEEPAV=0 NUMSOC=4 EDIT=         TIMEOUT=0
HWSC0001I    CLIENTID USERID   TRANCODE DATASTORE STATUS    SECOND  CLNTPORT IP-ADDRESS       APSB-TOKEN
HWSC0001I    CLIENT63 USRT003  apol12   IMSY      RECV WFCM     55      4118 192.0.2.0
HWSC0001I    CLIENT62 USRT002           IMS1      CONN         210      4115 192.0.2.0
HWSC0001I    CLIENT61 USRT001  apol12   IMSN      RECV          13      4112 192.0.2.0
HWSC0001I    TOTAL CLIENTS=3  RECV=2 CONN=1 XMIT=0 OTHER=0
```

If a port has no active clients on it, these output fields are not displayed. Instead, the command output displays `NO ACTIVE CLIENTS` after the row that identifies the port.

The information returned for active clients includes, in alphabetical order:

APSB-TOKEN
Open Database Manager (ODBM) APSB token.

CLIENTID
The name of the client.

For connections from other IMS Connect instances, the other IMS Connect instance generates this value.

CLNTPORT
A random number that TCP/IP generates to represent a connection from a client.

DATASTORE
The data store associated with this client.

The IP address of the client for this connection.

If IPV6 is enabled, the IP address format consists of eight hexadecimal numbers divided by colons. If IPV6 is not

enabled, the IP address format of IPV4 is used. The following example is for an IPV6 IP address that is displayed in IPV6 format:

```
FEDC:ABCD:2222:3333:FEDC:DB55:6666:3322
```

The following example is for an IPV4 IP address that is displayed in IPV6 format:

```
0:0:0:0:0:FFFF:945:33FF
```

ISCNAME
The name of the ISC statement in the IMS Connect configuration member.

LCLPLKID
The MSC physical link in the local IMS system to which IMS Connect routes the messages from the client. This field exists only for MSC communications.

SECOND
Number of seconds that the client has been in the specified status.

STATUS
State of the thread of the client. The client thread state can be one of the following values:

RECV In a receive state waiting for input from client.

READ In a partial read state. IMS Connect is reading a message from the client but has not yet received the entire length of the message.

CONN
Waiting for output from IMS.

XMIT Sending data to client.

CONV
In a conversational state.

WFCM
Waiting for confirmation (ACK, NAK, or DEALLOCATE) from client.

TRANCODE
The transaction code submitted by the client.

USERID
The USERID name passed to IMS Connect.

Example 1: QUERY PORT

Command input:

```
F HWS1,QUERY PORT NAME(*) SHOW(ALL)
```

Command output:

```
HWSC0001I  PORT=9999     STATUS=ACTIVE     KEEPAV=0 NUMSOC=4 EDIT=         TIMEOUT=0
HWSC0001I    CLIENTID USERID   TRANCODE DATASTORE STATUS    SECOND  CLNTPORT IP-ADDRESS       APSB-TOKEN
HWSC0001I    CLIENT63 USRT003  apol12   IMSY      RECV WFCM     55      4118 192.0.2.0
HWSC0001I    CLIENT62 USRT002           IMS1      CONN         210      4115 192.0.2.1
HWSC0001I    CLIENT61 USRT001  apol12   IMSN      RECV          13      4112 192.0.2.2
HWSC0001I    TOTAL CLIENTS=3  RECV=2 CONN=1 XMIT=0 OTHER=0
HWSC0001I  PORT=LOCAL    STATUS=ACTIVE     KEEPAV=0 NUMSOC=0 EDIT=         TIMEOUT=0
HWSC0001I    NO ACTIVE CLIENTS
HWSC0001I  PORT=9998S    STATUS=NOT ACTIVE KEEPAV=0 NUMSOC=1 EDIT=         TIMEOUT=0
```

```
HWSC0001I  PORT=1111D    STATUS=ACTIVE     KEEPAV=5 NUMSOC=1 EDIT=         TIMEOUT=6000
HWSC0001I    NO ACTIVE CLIENTS
HWSC0001I  PORT=2222D    STATUS=ACTIVE     KEEPAV=0 NUMSOC=1 EDIT=         TIMEOUT=100
HWSC0001I    NO ACTIVE CLIENTS
```

Explanation: The status of all IMS Connect ports is displayed and shows no active clients.

Example 2: IMS Connect to IMS Connect connections for MSC

For IMS Connect to IMS Connect connections that are used for MSC, the information displayed by VIEWPORT and QUERY PORT commands relates only to the receiving end of the IMS Connect to IMS Connect connection. To view information about the sending side of an IMS Connect to IMS Connect connection, use the VIEWRMT or QUERY RMTIMSCON command.

In the following example output, the VIEWPORT or QUERY PORT command was executed on a local IMS Connect. The command output displays the information for port 9999, on which IMS Connect receives MSC messages. The client connections on the port, MSCC73E0 and MSC0EBB0, are shown under CLIENTID. The LCLPLKID label identifies this port as being used for MSC TCP/IP communication.

```
HWSC0001I    PORT=9999     STATUS=ACTIVE     KEEPAV=0 NUMSOC=3      EDIT=       TIMEOUT=0
HWSC0001I      CLIENTID LCLPLKID  STATUS      SECOND CLNTPORT IP-ADDRESS
HWSC0001I      MSCC73E0 MSC12     RECV             6 1026     2001:DB8:0:0:0:0:0:0/32
HWSC0001I      MSC0EBB0 MSC12     RECV            56 1025     2001:DB8:0:0:0:0:0:0/32
HWSC0001I      TOTAL CLIENTS=2  RECV=2 READ=0 CONN=0 XMIT=0 OTHER=0
```

Example 3: IMS Connect to IMS Connect connections for OTMA

If a socket connection between IMS Connect instances is used for OTMA messages, the VIEWPORT or QUERY PORT command output shows relevant port information only when issued at the receiving IMS Connect instance.

In the sending IMS Connect instance, the VIEWPORT or QUERY PORT command does not display port information that is relevant to the IMS Connect to IMS Connect connection. However, the VIEWRMT or QUERY RMTIMSCON command displays the listening port of the receiving IMS Connect, as well as the send client ID associated with the connection. The listening port and send client ID match the information displayed at the receiving installation by the VIEWPORT or QUERY PORT command.

In the receiving IMS Connect instance, the VIEWPORT or QUERY PORT command displays the port that the receiving IMS Connect is listening on and the sending client ID associated with the connection.

The following example shows the output from the VIEWPORT or QUERY PORT command on a receiving IMS Connect instance. The port that the receiving IMS Connect uses for the IMS Connect to IMS Connect connection is identified by `PORT=5555`. The connection from the sending IMS Connect is identified by the auto-generated client ID, OTM924FA, that is shown under CLIENTID. This CLIENTID value is the same value shown under SENDCLNT at the sending IMS installation. The CLNTPORT is the port that the sending IMS Connect uses to send messages on the connection.

```
HWSC0001I    PORT=5555     STATUS=ACTIVE     KEEPAV=0 NUMSOC=2       EDIT=        TIMEOUT=0
HWSC0001I      CLIENTID USERID TRANCODE DATASTORE STATUS  SECOND CLNTPORT IP-ADDRESS              APSB-TOKEN
HWSC0001I      OTM924FA APOL1  APOL11   IMS2      RECV        11 1026     2001:DB8:0:0:0:0:0:0/32
HWSC0001I      TOTAL CLIENTS=1  RECV=1 READ=0 CONN=0 XMIT=0 OTHER=0
```

Example 4: An ISC TCP/IP connection to CICS

Command input:

```
F ICON1,QUERY PORT NAME(7777) SHOW(ALL)
```

Command output:

```
HWSC0001I  PORT=7777C  STATUS=ACTIVE   KEEPAV=0 NUMSOC=3 EDIT=       TIMEOUT=0
HWSC0001I     CLIENTID ISCNAME STATUS SECOND CLNTPORT  IP-ADDRESS
HWSC0001I     ISC33333 LU6TCP2   CONN     14     3579  2001:DB8:0:0:0:0:0:0/32
HWSC0001I     ISC44444 LU6TCP2   CONN      9     2468  2001:DB8:0:0:0:0:0:0/32
HWSC0001I     TOTAL CLIENTS=3  RECV=1 CONN=2 XMIT=0 OTHER=0
```

Related reference:

"VIEWPORT command" on page 521

QUERY IMSCON TYPE(PORT) command (Commands)

IMS Connect QUERY RMTIMSCON command

Use the IMS Connect z/OS MODIFY command QUERY RMTIMSCON in a local instance of IMS Connect to display information about all connections to a remote instance of IMS Connect.

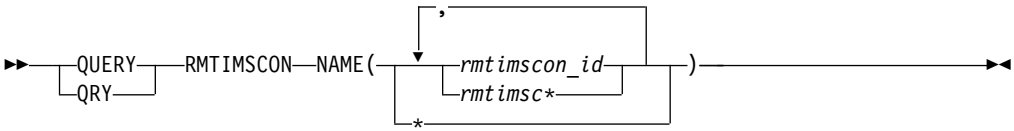
QUERY or QRY is used to query the status or attributes of a specified resource that is managed by IMS Connect. RMTIMSCON, which represents a connection between a local instance of IMS Connect and a remote instance of IMS Connect, is an IMS Connect-managed resource.

The QUERY RMTIMSCON command performs similar functions as the IMS Connect WTOR command VIEWRMT.

Subsections:

- "Syntax"
- "Keywords"
- "Usage" on page 582
- "Equivalent IMS Connect type-2 and WTOR commands" on page 582
- "Output fields" on page 582
- "Examples" on page 585

Syntax



Keywords

NAME(*rmtimscon_id*)

The ID of a connection to a remote instance of IMS Connect. The value specified here must match a value specified on the ID parameter of a RMTIMSCON statement that is defined to the local IMS Connect instance. Wildcards (* and %) are supported in the name.

Usage

Use the QUERY RMTIMSCON command to display information about the attributes and status of connections with remote instances of IMS Connect and about the sockets that are currently open on the connections.

Equivalent IMS Connect type-2 and WTOR commands

The following table lists IMS Connect type-2 commands and IMS Connect WTOR commands that perform similar functions as the QUERY RMTIMSCON command.

QUERY RMTIMSCON command	Equivalent IMS Connect type-2 command	Equivalent WTOR command
QUERY RMTIMSCON NAME(*)	QUERY IMSCON TYPE(RMTIMSCON) NAME(*) SHOW(ALL | *show_parm*)	VIEWRMT ALL
QUERY RMTIMSCON NAME(*rmtimscon_name*)	QUERY IMSCON TYPE(RMTIMSCON) NAME(*rmtimscon_name*) SHOW(ALL | *show_parm*)	VIEWRMT *rmtimscon_name*

Output fields

Output fields that return only a single value are displayed in the format `FIELD_LABEL=VALUE`, for example: `RMTIMSCON=ICON1`.

Output fields that return multiple values, such as the output fields for multiple sockets on a connection, are displayed in a column format, where the field label is a column heading and the values appear under the heading on separate rows in the column. For example:

```
HWSC0001I      SENDCLNT USERID   STATUS SECOND SENDPORT
HWSC0001I      OTM11111 USER01   CONN      100 1111
HWSC0001I      OTM22222 USER01   CONN       89 2222
```

The following list describes the output fields of the QUERY RMTIMSCON command.

RMTIMSCON=
Identifies a connection between this local IMS Connect instance and a remote instance of IMS Connect. The following additional output fields describe the status, attributes, and sockets that belong to the connection identified in the RMTIMSCON output field:

STATUS
The state of the connection displayed in the RMTIMSCON output field. A connection can be in the following states:

ACTIVE
This connection has at least one send socket open to a remote IMS Connect instance.

NOT ACTIVE
This connection does not have any send sockets open to a remote IMS Connect instance.

RETRY CONN
IMS Connect is attempting to automatically reconnect to the remote IMS Connect instance after a previous attempt

to connect failed. If the current reconnection attempt fails, IMS Connect attempts to reconnect again every 2 minutes.

While a connection has a RETRY CONN status, the number of open sockets fluctuates. IMS Connect opens a socket when actively trying to reconnect. IMS Connect closes the socket during the 2-minute interval between reconnection attempts.

STOPPED
A STOPRMT command stopped communication on this connection. Any messages sent to this connection are rejected and an error is sent back to the sender.

IP-ADDRESS
The IP address of the IMS Connect instance at the remote end of this connection.

PORT The port number used by the remote IMS Connect instance for this connection.

HOSTNAME
The host name of the remote IMS Connect instance, if available.

AUTOCONN
Indicates whether IMS Connect automatically establishes send sockets for OTMA communications on this connection during IMS Connect startup. Possible values are:

Y Send sockets are automatically established on this connection when IMS Connect starts up. The number of sockets that IMS Connect establishes during startup is displayed in the RESVSOC output field

N No sockets are established on this connection during IMS Connect startup. For OTMA messages, send sockets are established upon receipt of a message from OTMA for delivery to the remote IMS system.

PERSISTENT
Indicates whether the sockets on this connection are persistent.

Y The sockets established on this connection are persistent.

N The sockets established on this connection are not persistent. After messages are sent on this connection, the sockets that they are sent on are closed.

IDLETO
The time interval that open sockets on this connection can remain idle before they are closed by IMS Connect.

RESVSOC
The number of send sockets that are reserved for use by this connection.

NUMSOC
The number of sockets that are currently open on this connection.

SENDCLNT
The randomly generated send socket ID. The local instance of IMS Connect uses the ID to uniquely identify the socket. The send

socket ID is used as the client ID when establishing a socket connection with the remote IMS Connect instance.

The send socket IDs that are used for OTMA begin with "OTM." The send socket IDs that are used for MSC begin with "MSC."

The values returned for this output field are displayed in tabular format.

If no sockets are currently open on this connection, this output field is not displayed.

LCLPLKID
The ID of a local MSC physical link that is using this connection. This value is specified on the LCLPLKID parameter of the MSC statement in the IMS Connect configuration member. The LCLPLKID output field is displayed only for connections that are used for MSC communications.

The values returned for this output field are displayed in tabular format.

If no sockets are currently open on this connection, this output field is not displayed.

USERID
The user ID provided by the local IMS system for transaction authorization by the remote IMS system. The USERID output field is displayed only for connections that are used for OTMA communications.

If no sockets are currently open on this connection, this output field is not displayed.

The values returned for this output field are displayed in tabular format.

STATUS
The current state of this socket connection. A socket connection can have the following states:

CONN
This socket connection can send messages to the remote IMS Connect.

RECV This socket connection is waiting to receive a response from the remote IMS Connect.

The values returned for this output field are displayed in tabular format.

If no sockets are currently open on this connection, this output field is not displayed.

SECOND
The number of seconds that this connection has been in the state shown on the same row under the STATUS output field heading.

The values returned for this output field are displayed in tabular format.

If no sockets are currently open on this connection, this output field is not displayed.

SENDPORT
The local port used for this connection.

The values returned for this output field are displayed in tabular format.

If no sockets are currently open on this connection, this output field is not displayed.

Examples

Example 1 for QUERY RMTIMSCON command

Command input:

```
F ICON1,QUERY RMTIMSCON NAME(*)
```

Command output:

```
HWSC0001I    RMTIMSCON=CONNECT2 STATUS=ACTIVE
HWSC0001I      IP-ADDRESS=127.0.0.1       PORT=5555
HWSC0001I      HOSTNAME=ICON2.EXAMPLE.COM
HWSC0001I      AUTOCONN=N PERSISTENT=Y
HWSC0001I      IDLETO=6000
HWSC0001I      RESVSOC=10        NUMSOC=2
HWSC0001I      SENDCLNT LCLPLKID STATUS SECOND SENDPORT
HWSC0001I      MSC11111 MSC12    CONN      100 1234
HWSC0001I      MSC22222 MSC12    CONN       89 5678
HWSC0001I    RMTIMSCON=CONNECT3 STATUS=ACTIVE
HWSC0001I      IP-ADDRESS=192.0.2.0       PORT=9999
HWSC0001I      HOSTNAME=ICON3.EXAMPLE.COM
HWSC0001I      AUTOCONN=Y PERSISTENT=Y
HWSC0001I      IDLETO=6000
HWSC0001I      RESVSOC=4         NUMSOC=4
HWSC0001I      SENDCLNT USERID   STATUS SECOND SENDPORT
HWSC0001I      OTM11111 USER01   CONN      100 1111
HWSC0001I      OTM22222 USER01   CONN       89 2222
HWSC0001I      OTM33333 USER02   CONN       81 3333
HWSC0001I      OTM44444 USER03   RECV       23 4444
```

Related reference:

"STOPSCLN command" on page 481

"IMS Connect DELETE SENDCLNT command" on page 537

"VIEWRMT command" on page 526

QUERY IMSCON TYPE(RMTIMSCON) command (Commands)

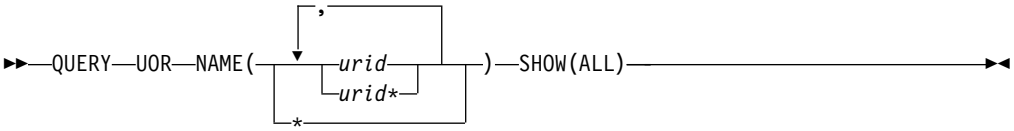
IMS Connect QUERY UOR command

This command is used to display the current status of the request unit of recovery (UOR) identifier. QUERY UOR performs similar function as the VIEWUOR command.

Subsections:

- "Syntax" on page 586
- "Keywords" on page 586
- "Equivalent IMS Connect type-2 and WTOR commands" on page 586
- "Examples" on page 586

Syntax



Keywords

QUERY or QRY is used to query the status or attributes of the specified resource. UOR is an IMS Connect managed resource. QUERY MEMBER is used to display the status of IMS Connect. This command can be issued through the Operations Manager API or through the z/OS modify interface.

NAME ( )
Specifies the name of the unit of recovery identifier (URID) to be displayed. You can specify a single URID, a wildcard, or a list of URIDs separated by commas. You can specify NAME(*) for the command to be processed for all UORs.

urid
Specifies the URID to be displayed.

SHOW ( )
Specifies the target type to be returned.

ALL
Returns all output fields. This is the default.

Equivalent IMS Connect type-2 and WTOR commands

The following table lists IMS Connect type-2 commands and IMS Connect WTOR commands that perform similar functions as the QUERY UOR command.

QUERY UOR command	Equivalent IMS Connect type-2 command	Equivalent WTOR command
QUERY UOR NAME(*) SHOW(ALL)	QUERY IMSCON TYPE(UOR) NAME(*) SHOW(ALL | *show_parm*)	VIEWUOR ALL
QUERY UOR NAME(*uor_id*) SHOW(ALL)	QUERY IMSCON TYPE(UOR) NAME(*uor_id*) SHOW(ALL | *show_parm*)	VIEWUOR *uor_id*

Examples

Query the status of IMS Connect URIDs.

Command input:

```
F HWS1,QRY UOR NAME(*)
```

Command output:

```
HWSC0050I NO ACTIVE UOR
```

Related reference:

"VIEWUOR command" on page 530

QUERY IMSCON TYPE(UOR) command (Commands)

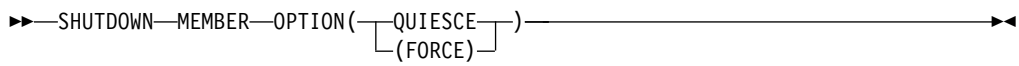
IMS Connect SHUTDOWN MEMBER command

This command is used to shut down IMS Connect. SHUTDOWN MEMBER performs similar functions as the CLOSEHWS command.

Subsections:

- "Syntax"
- "Keywords"
- "Equivalent IMS Connect type-2 and WTOR commands"
- "Examples" on page 588

Syntax



Keywords

`SHUTDOWN` or `SHUT` is used to shutdown a specified resource. `MEMBER` is an IMS Connect specified managed resource. The command, `SHUTDOWN MEMBER`, is used to shutdown IMS Connect.

`OPTION ( )`
Specifies the attributes to be stopped.

`QUIESCE`
Specifies that termination is to end all client and data store connections in a controlled manner.

`FORCE ( )`
Specifies that termination is to end all client and data store connections immediately. Immediate termination forces any IMS application that is running with connected clients to abnormally terminate.

Equivalent IMS Connect type-2 and WTOR commands

The following table lists IMS Connect type-2 commands and IMS Connect WTOR commands that perform similar functions as the SHUTDOWN MEMBER command.

SHUTDOWN MEMBER command	Equivalent IMS Connect type-2 command	Equivalent WTOR command
SHUTDOWN MEMBER	UPDATE IMSCON TYPE(CONFIG) SHUTDOWN(COMM)	CLOSEHWS
SHUTDOWN MEMBER OPTION(FORCE)	UPDATE IMSCON TYPE(CONFIG) SHUTDOWN(COMM) OPTION(FORCE)	CLOSEHWS FORCE
SHUTDOWN MEMBER OPTION(QUIESCE)	UPDATE IMSCON TYPE(CONFIG) SHUTDOWN(COMM) OPTION(QUIESCE)	CLOSEHWS QUIESCE

Examples

Shutdown HWS with force option.

Command input:

```
F HWS07,SHUTDOWN MEMBER OPTION(FORCE)
```

Command output:

```
HWS07 PURGED
```

Explanation: The HWS member is shut down.

Related reference:

UPDATE IMSCON TYPE(CONFIG) command (Commands)

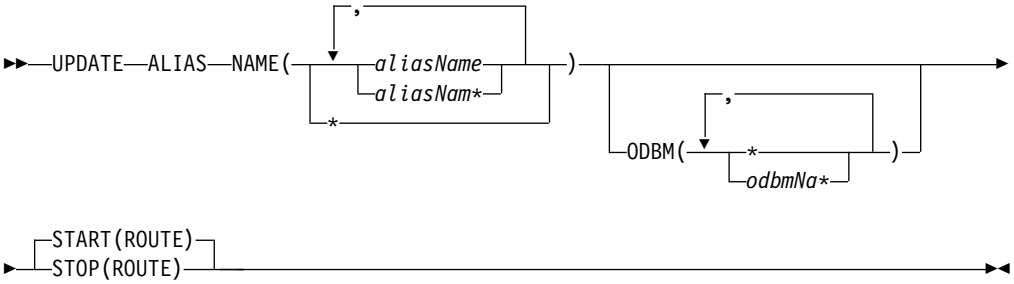
IMS Connect UPDATE ALIAS command

Use the UPDATE ALIAS command to update the requested IMS alias. The UPDATE ALIAS command performs similar functions as the STARTIA and STOPIA commands.

Subsections:

- "Syntax"
- "Keywords"
- "Equivalent IMS Connect type-2 and WTOR commands" on page 589
- "Examples" on page 589

Syntax



Keywords

UPDATE or UPD is used to update the status or attributes of a specified resource. ALIAS is a resource managed by IMS Connect. The command UPDATE ALIAS is used to update the current status of the specified alias.

NAME( )

Specifies the name of the alias to be updated. You can specify a single alias, a wildcard name, or a list of aliases separated by commas. You can specify NAME(*) for the command to be processed for all aliases.

aliasName

Specifies the alias to be updated.

ODBM( )
Specifies the Open Database Manager (ODBM) name with which the named alias is associated.

START( )
Specifies the attributes to be started.

ROUTE
Enables message routing to this alias.

STOP( )
Specifies the attributes to be stopped.

ROUTE
Disables message routing to this alias.

Equivalent IMS Connect type-2 and WTOR commands

The following table lists IMS Connect type-2 commands and IMS Connect WTOR commands that perform similar functions as the UPDATE ALIAS command.

UPDATE ALIAS command	Equivalent IMS Connect type-2 command	Equivalent WTOR command
UPDATE ALIAS NAME(*aliasName*) ODBM(*odbmName*) START(ROUTE)	UDPATE IMSCON TYPE(ALIAS) NAME(*alias_name*) ODBM(*odbm_name*) START(COMM)	STARTIA *alias_name odbm_name*
UPDATE ALIAS NAME(*aliasName*) ODBM(*odbmName*) STOP(ROUTE)	UDPATE IMSCON TYPE(ALIAS) NAME(*alias_name*) ODBM(*odbm_name*) STOP(COMM)	STOPIA *alias_name odbm_name*

Examples

Command input:

```
F HWS1,UPD ALIAS NAME(ALI1) STOP(ROUTE)
```

Explanation: The message routing to this alias, ALI1, is disabled.

Related reference:

"STARTIA command" on page 461

"STOPIA command" on page 471

UPDATE IMSCON TYPE(ALIAS) command (Commands)

IMS Connect UPDATE CONVERTER command

Use the UPDATE CONVERTER command to refresh one or more XML converters without restarting IMS Connect.

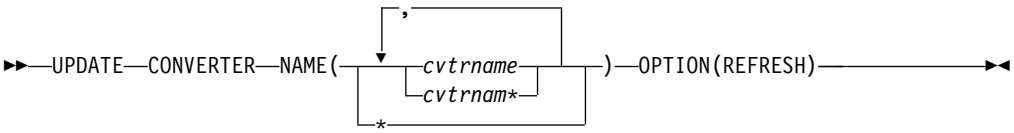
This command is equivalent to the WTOR command REFRESH CONVERTER.

Subsections:

- "Syntax" on page 590
- "Keywords" on page 590
- "Usage notes" on page 590
- "Equivalent IMS Connect type-2 and WTOR commands" on page 590

- "Examples"

Syntax



Keywords

The following keywords are valid for the UPDATE CONVERTER command:

cvtrname
Specifies the XML converter for IMS Connect to refresh. The wildcard symbol (*) is supported for pattern matching. Specifying only a wildcard refreshes all XML converters.

OPTION(REFRESH)
Required keyword.

Usage notes

Issue the UPDATE CONVERTER command to IMS Connect with a z/OS modify identifier.

Equivalent IMS Connect type-2 and WTOR commands

The following table lists IMS Connect type-2 commands and IMS Connect WTOR commands that perform similar functions as the UPDATE CONVERTER command.

UPDATE CONVERTER command	Equivalent IMS Connect type-2 command	Equivalent WTOR command
UPDATE CONVERTER NAME(*converter_name*) OPTION(REFRESH)	UPDATE IMSCON TYPE(CONVERTER) NAME(*converter_name*) OPTION(REFRESH)	REFRESH CONVERTER NAME(*converter_name*)

Examples

Command input:

```
F hws1,UPDATE CONVERTER NAME(IMSPHBKD)
```

Command output:

```
HWSA0375I XML CONVERTER IMSPHBKD WAS SUCCESSFULLY REFRESHED,M=XMLA
```

Related tasks:

Configuring XML conversion support for IMS Connect clients (System Definition)

Related reference:

"REFRESH CONVERTER command" on page 451

UPDATE IMSCON TYPE(CONVERTER) command (Commands)

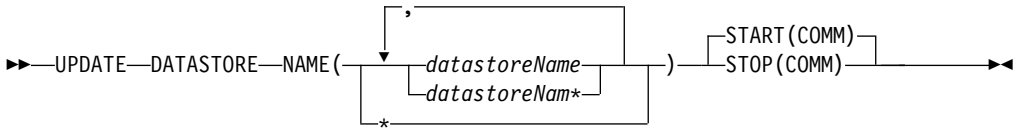
IMS Connect UPDATE DATASTORE command

The UPDATE DATASTORE command is used to update the requested data store. The UPDATE DATASTORE command performs similar functions as the OPENDS and STOPDS commands.

Subsections:

- "Syntax"
- "Keywords"
- "Equivalent IMS Connect type-2 and WTOR commands"
- "Examples" on page 592

Syntax



Keywords

UPDATE or UPD is used to update the status or attributes of a specified resource. DATASTORE is an IMS Connect managed specified resource. The command UPDATE DATASTORE is used to update the current status of the requested data store.

NAME( )
Specifies the name of the data store to be updated. You can specify a single data store, a wildcard name, or a list of data stores separated by commas. You can specify NAME(*) for the command to be processed for all data stores.

datastoreName
Specifies the data store to be updated.

START( )
Specifies the attributes to be started.

COMM
Starts the communication with the data store.

STOP( )
Specifies the attributes to be stopped.

COMM
Stops the communication with the data store.

Equivalent IMS Connect type-2 and WTOR commands

The following table lists IMS Connect type-2 commands and IMS Connect WTOR commands that perform similar functions as the UPDATE DATASTORE command.

UPDATE DATASTORE command	Equivalent IMS Connect type-2 command	Equivalent WTOR command
UPDATE DATASTORE NAME(*datastoreName*) START(COMM)	UPDATE IMSCON TYPE(DATASTORE) NAME(*datastore_name*) START(COMM)	OPENDS *datastore_id* STARTDS *datastore_id*
UPDATE DATASTORE NAME(*datastoreName*) STOP(COMM)	UPDATE IMSCON TYPE(DATASTORE) NAME(*datastore_name*) STOP(COMM)	STOPDS *datastore_id*

Examples

Command input:

```
F HWS1,UPD DATASTORE NAME(SOCKEYE) STOP(COMM)
```

Command output:

```
HWSD028I COMMUNICATION WITH DS=SOCKEYE    STOPPED;
M=DSCM
```

Explanation: The communication with the data store, SOCKEYE, is stopped.

Related reference:

"OPENDS command" on page 447

"STARTDS command" on page 460

"STOPDS command" on page 470

UPDATE IMSCON TYPE(DATASTORE) command (Commands)

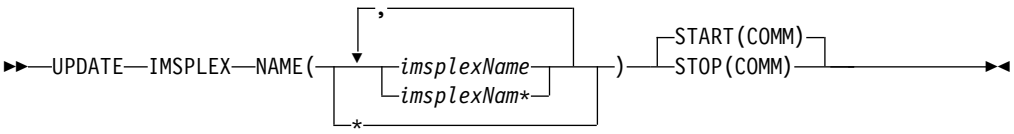
IMS Connect UPDATE IMSPLEX command

Use the UPDATE IMSPLEX command to stop or start communications between IMS Connect and the specified IMSplex.

Subsections:

- "Syntax"
- "Keywords" on page 593
- "Usage notes" on page 593
- "Equivalent IMS Connect type-2 and WTOR commands" on page 593
- "Examples" on page 594

Syntax



Keywords

UPDATE or UPD is used to update the status or attributes of a specified resource. IMSPLEX is a resource managed by IMS Connect. The command UPDATE IMSPLEX is used to update the current status of the specified IMSplex.

NAME( )
Specifies the name of the IMSplex to be updated. You can specify a single IMSplex, a wildcard name, or a list of IMSplexes separated by commas. You can specify NAME(*) for the command to be processed for all IMSplexes.

imsplexName
Specifies the IMSplex to be updated.

START( )
Specifies the attributes to be started.

COMM
Starts the communication with the IMSplex.

STOP( )
Specifies the attributes to be stopped.

COMM
Stops the communication with the IMSplex.

Usage notes

When the UPDATE IMSPLEX NAME(*imsplexName*) is issued with the STOP(COMM) keyword, work currently in progress for the IMSplex ends and communication with the IMSplex and its threads are terminated. Any messages in progress are rejected and an error message is returned to the requester. STOPIP can also be used for any error situation that requires immediate termination of communication with the IMSplex.

If IMS Connect supports ISC TCP/IP links and the STOP(COMM) keyword is used, IMS Connect sends a notification to IMS for each ISC link to inform IMS that all of the ISC parallel sessions are terminated on the ISC link.

If IMS Connect supports MSC IMS-to-IMS TCP/IP connections and the STOP(COMM) keyword is used, IMS Connect sends a notification to IMS for each MSC physical link to inform IMS that all of the MSC logical links are terminated on the physical link.

Use the START(COMM) keyword to start communication with the IMSplex at a later time.

Equivalent IMS Connect type-2 and WTOR commands

The following table lists IMS Connect type-2 commands and IMS Connect WTOR commands that perform similar functions as the UPDATE IMSPLEX command.

UPDATE IMSPLEX command	Equivalent IMS Connect type-2 command	Equivalent WTOR command
UPDATE IMSPLEX NAME(*imsplex_name*) START(COMM)	UPDATE IMSCON TYPE(IMSPLEX) NAME(*imsplex_name*) START(COMM)	OPENIP *imsplex_id* STARTIP *imsplex_id*

UPDATE IMSPLEX command	Equivalent IMS Connect type-2 command	Equivalent WTOR command
UPDATE IMSPLEX NAME(*imsplex_name*) STOP(COMM)	UPDATE IMSCON TYPE(IMSPLEX) NAME(*imsplex_name*) STOP(COMM)	STOPIP *imsplex_id*

Examples

Command input:

```
F HWS1,UPD IMSPLEX NAME(PLEX1) STOP(COMM)
```

Command output:

```
HWSM0584I COMMUNICATION WITH IMSPLEX=PLEX1    STOPPED; M=DSCM
```

Explanation: The communication with the IMSplex, PLEX1, is stopped.

Related reference:

"OPENIP command" on page 448

"STARTIP command" on page 462

"STOPIP command" on page 473

UPDATE IMSCON TYPE(IMSPLEX) command (Commands)

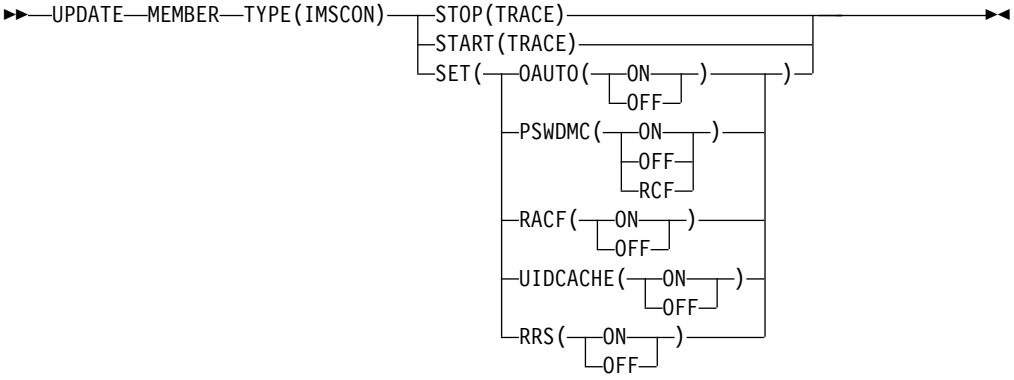
IMS Connect UPDATE MEMBER command

The UPDATE MEMBER command is used to update the attributes of IMS Connect.

Subsections:

- "Syntax"
- "Keywords" on page 595
- "Equivalent IMS Connect type-2 and WTOR commands" on page 596
- "Examples" on page 597

Syntax



Keywords

UPDATE or UPD is used to update the status or attributes of a specified resource. MEMBER is an IMS Connect managed resource. The command, UPDATE MEMBER, is used to update the status of IMS Connect.

TYPE( )
Specifies the target type for action.

IMSCON
Specifies IMS Connect as the target type.

START( )
Specifies the attributes to be started.

TRACE
Specifies that the line trace will be started.

STOP( )
Specifies the attributes to be stopped.

TRACE
Specifies that the line trace will be stopped.

SET
Specifies the attribute values to be changed.

OAUTO( )
Specifies the attribute to be set.

ON Enables IMS Connect to automatically register with all current ODBMs and future ODBMs that enter the IMSplex.

OFF Disables IMS Connect from automatically registering with any future ODBMs that enter the IMSplex.

PSWDMC( )
Specifies the IMS Connect support for mixed-case passwords.

ON Enables IMS Connect support for mixed-case passwords.

OFF Disables IMS Connect support for mixed-case passwords. IMS Connect converts any lowercase characters in passwords to uppercase characters.

RCF IMS Connect support for mixed-case passwords is determined by the mixed-case password specification in RACF. If RACF does not support mixed-case passwords, IMS Connect converts any lowercase characters in passwords to uppercase characters.

RACF( )
Specifies whether IMS Connect will use RACF security.

ON Enables the RACF user identification and verification.

OFF Disables the RACF user identification and verification.

UIDCACHE( )
Specifies whether IMS Connect stores verified RACF user IDs in a cache. This keyword performs the same function as the WTOR command SETUIDC.

Restriction: This keyword is valid only if RACF security support is enabled.

ON Enables the RACF user ID cache.

OFF Disables the RACF user ID cache.

RRS( )
Specifies the attributes to be set.

ON Enables communication between IMS Connect and z/OS Resource Recovery Services (RRS). RRS is required for two-phase-commit support.

OFF Disables communication between IMS Connect and RRS.

Equivalent IMS Connect type-2 and WTOR commands

The following table lists IMS Connect type-2 commands and IMS Connect WTOR commands that perform similar functions as the UPDATE MEMBER command.

UPDATE MEMBER command	Equivalent IMS Connect type-2 command	Equivalent WTOR command
UPDATE MEMBER TYPE(IMSCON) SET(OAUTO(ON))	UPDATE IMSCON TYPE(CONFIG) SET(OAUTO(ON))	SETOAUTO YES
UPDATE MEMBER TYPE(IMSCON) SET(OAUTO(OFF))	UPDATE IMSCON TYPE(CONFIG) SET(OAUTO(OFF)	SETOAUTO NO
UPDATE MEMBER TYPE(IMSCON) SET(PSWDMC(ON))	UPDATE IMSCON TYPE(CONFIG) SET(PSWDMC(ON))	SETPWMC ON
UPDATE MEMBER TYPE(IMSCON) SET(PSWDMC(OFF))	UPDATE IMSCON TYPE(CONFIG) SET(PSWDMC(OFF))	SETPWMC OFF
UPDATE MEMBER TYPE(IMSCON) SET(PSWDMC(RCF))	UPDATE IMSCON TYPE(CONFIG) SET(PSWDMC(RCF))	SETPWMC RCF
UPDATE MEMBER TYPE(IMSCON) SET(RACF(ON))	UPDATE IMSCON TYPE(CONFIG) SET(RACF(ON))	SETRACF ON
UPDATE MEMBER TYPE(IMSCON) SET(RACF(OFF))	UPDATE IMSCON TYPE(CONFIG) SET(RACF(OFF))	SETRACF OFF
UPDATE MEMBER TYPE(IMSCON) SET(RRS(ON))	UPDATE IMSCON TYPE(CONFIG) SET(RRS(ON))	SETRRS ON
UPDATE MEMBER TYPE(IMSCON) SET(RRS(OFF))	UPDATE IMSCON TYPE(CONFIG) SET(RRS(OFF))	SETRRS OFF
UPDATE MEMBER TYPE(IMSCON) SET(UIDCACHE(ON))	UPDATE IMSCON TYPE(CONFIG) SET(UIDCACHE(ON))	SETUIDC ON
UPDATE MEMBER TYPE(IMSCON) SET(UIDCACHE(OFF))	UPDATE IMSCON TYPE(CONFIG) SET(UIDCACHE(OFF))	SETUIDC OFF
UPDATE MEMBER TYPE(IMSCON) START(TRACE)	UPDATE IMSCON TYPE(CONFIG) START(RECORDER)	RECORDER OPEN

UPDATE MEMBER command	Equivalent IMS Connect type-2 command	Equivalent WTOR command
UPDATE MEMBER TYPE(IMSCON) STOP(TRACE)	UPDATE IMSCON TYPE(CONFIG) STOP(RECORDER)	RECORDER CLOSE

Examples

Command input:

```
F HWS1,UPD MEMBER TYPE(IMSCON) SET(RACF(OFF))
```

Command output: None.

Command input:

```
F HWS1,QUERY MEMBER TYPE(IMSCON) SHOW(ALL)
```

Command output:

```
HWSC0001I   HWS  ID=HWS1     RACF=N
```

Explanation: RACF security check is turned off.

Command input:

```
F HWS1,UPDATE MEMBER TYPE(IMSCON) SET(UIDCACHE(ON))
```

Command output:

```
HWSP1501I RACF USERID CACHING ENABLED,M=SDRC
```

Explanation: The RACF user ID cache is now enabled.

Related reference:

"RECORDER command" on page 450
"SETOAUTO command" on page 454
"SETPWMC command" on page 455
"SETRACF command" on page 457
"SETRRS command" on page 458
"SETUIDC command" on page 459

UPDATE IMSCON TYPE(CONFIG) command (Commands)

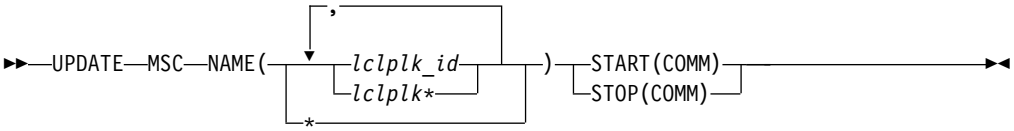
IMS Connect UPDATE MSC command

Use the UPDATE MSC command in a local IMS Connect instance to start or stop the communications on an MSC physical link that is defined to the IMS Connect instance. The command performs similar functions as the STARTMSC and STOPMSC commands.

Subsections:

- "Syntax" on page 598
- "Keywords" on page 598
- "Usage" on page 598
- "Equivalent IMS Connect type-2 and WTOR commands" on page 599
- "Examples" on page 599

Syntax



Keywords

UPDATE or UPD is used to update the status or attributes of a specified resource. MSC represents an MSC physical link definition that is a resource managed by IMS Connect. The command UPDATE MSC is used to update the current status of the specified MSC physical link.

NAME( )
The name of the MSC physical link to be updated, as defined to the local IMS Connect instance on the LCLPLKID parameter of the MSC configuration statement. You can specify a single MSC physical link name, a list of MSC physical link names separated by commas, or a partial name with a wildcard character. You can specify NAME(*) for the command to be executed against all MSC physical links defined to an IMS Connect instance.

START( )
Specifies the attributes to be started.

COMM
Starts communications on the MSC physical link.

STOP( )
Specifies the attributes to be stopped.

COMM
Stops communication on the MSC physical link.

Usage

When the UPDATE MSC command is issued with the STOP(COMM) keyword, IMS Connect:

- Stops communication on the specified MSC physical link, including communications on all the MSC logical links that are assigned to the physical link
- Informs IMS that communication has stopped on the physical link so that IMS can also terminate the physical link and any logical links that are assigned to the physical link
- Changes the status of the MSC physical link and its assigned logical links to NOT ACTIVE
- For TCP/IP generic resources, clears affinity of a physical link to the IMS system
- Issues message HWSF3305I

When the UPDATE MSC command is issued with the START(COMM) keyword, IMS Connect:

- Resumes communications on the specified MSC physical link
- Changes the status of the MSC physical link to ACTIVE
- Issues message HWSF3300I

Use the QUERY MSC command or the VIEWMSC command to display information about MSC physical links that are defined to IMS Connect.

Equivalent IMS Connect type-2 and WTOR commands

The following table lists IMS Connect type-2 commands and IMS Connect WTOR commands that perform similar functions as the UPDATE MSC command.

UPDATE MSC command	Equivalent IMS Connect type-2 command	Equivalent WTOR command
UPDATE MSC NAME(*lclPlkid*) START(COMM)	UPDATE IMSCON TYPE(MSC) NAME(*lclPlkid*) START(COMM)	STARTMSC *lclPlkid*
UPDATE MSC NAME(*lclPlkid*) STOP(COMM)	UPDATE IMSCON TYPE(MSC) NAME(*lclPlkid*) STOP(COMM)	STOPMSC *lclPlkid*

Examples

UPDATE MSC command input when the STOP(COMM) keyword is specified:

```
F ICON1,UPDATE MSC NAME(MSC12) STOP(COMM)
```

UPDATE MSC command output when the STOP(COMM) keyword is specified:

```
HWSF3305I COMMUNICATIONS ON MSC PHYSICAL LINK MSC12 STOPPED; M=mc
```

UPDATE MSC command input when the START(COMM) keyword is specified:

```
F ICON1,UPDATE MSC NAME(MSC12) START(COMM)
```

UPDATE MSC command output when the START(COMM) keyword is specified:

```
HWSF3300I COMMUNICATIONS ON MSC PHYSICAL LINK MSC12 STARTED; M=mc
```

Related reference:

"STARTMSC command" on page 464

"STOPMSC command" on page 476

UPDATE IMSCON TYPE(MSC) command (Commands)

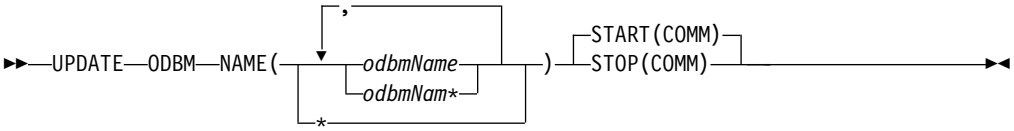
IMS Connect UPDATE ODBM command

Use the UPDATE ODBM command to update the requested ODBM. The UPDATE ODBM command performs similar functions as the STARTOD and STOPOD commands.

Subsections:

- "Syntax" on page 600
- "Keywords" on page 600
- "Examples" on page 600

Syntax



Keywords

UPDATE or UPD is used to update the status or attributes of a specified resource. ODBM is a resource managed by IMS Connect. The command UPDATE ODBM is used to update the current status of the specified ODBM.

NAME( )
Specifies the name of the ODBM to be updated. You can specify a single ODBM, a wildcard name, or a list of ODBMs separated by commas. You can specify NAME(*) for the command to be processed for all ODBMs.

odbmName
Specifies the ODBM to be updated.

START( )
Specifies the attributes to be started.

COMM
Starts the communication with the ODBM.

STOP( )
Specifies the attributes to be stopped.

COMM
Stops the communication with the ODBM.

Equivalent IMS Connect type-2 and WTOR commands

The following table lists IMS Connect type-2 commands and IMS Connect WTOR commands that perform similar functions as the UPDATE ODBM command.

UPDATE ODBM command	Equivalent IMS Connect type-2 command	Equivalent WTOR command
UPDATE ODBM NAME(*odbmName*) START(COMM)	UDPATE IMSCON TYPE(ODBM) NAME(*odbm_name*) START(COMM)	STARTOD *odbm_name*

Examples

Command input:

```
F HWS1,UPD ODBM NAME(ODBM1OD) STOP(COMM)
```

Command output:

```
HWSN1985I COMMUNICATION WITH ODBM=ODBM1OD  CLOSED; M=DSCM
```

Explanation: The communication with the ODBM, ODBM1OD, is stopped.

Related reference:

"STARTOD command" on page 465

"STOPOD command" on page 477

UPDATE IMSCON TYPE(ODBM) command (Commands)

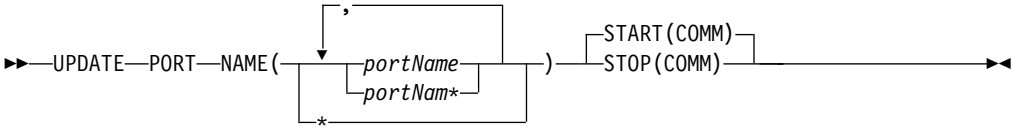
IMS Connect UPDATE PORT command

This command is used to update the port that is used by IMS Connect. UPDATE PORT performs similar functions as the OPENPORT and STOPPORT commands.

Subsections:

- "Syntax"
- "Keywords"
- "Equivalent IMS Connect type-2 and WTOR commands"
- "Examples" on page 602

Syntax



Keywords

UPDATE or UPD is used to update the status or attributes of a specified resource. PORT is an IMS Connect managed resource. The command, UPDATE PORT, is used to update the status of a requested port.

NAME ( )
Specifies the port name to be updated. You can specify a single port, a wildcard name, or a list of port names separated by commas. You can specify name(*) for the command to be processed for all ports.

portName
Specifies the port to be displayed.

START ( )
Specifies the attributes to be started.

COMM
Starts the communication with the TCP/IP port.

STOP ( )
Specifies the attributes to be stopped.

COMM
Stops the communication with the TCP/IP port.

Equivalent IMS Connect type-2 and WTOR commands

The following table lists IMS Connect type-2 commands and IMS Connect WTOR commands that perform similar functions as the UPDATE PORT command.

UPDATE PORT command	Equivalent IMS Connect type-2 command	Equivalent WTOR command
UPDATE PORT NAME(*port_name*) START(COMM)	UDPATE IMSCON TYPE(PORT) NAME(*port_name*) START(COMM)	OPENPORT *port_id* STARTPT *port_id*
UPDATE PORT NAME(*port_name*) STOP(COMM)	UDPATE IMSCON TYPE(PORT) NAME(*port_name*) STOP(COMM)	STOPPORT *port_id*

Examples

Command input:

```
F HWS1,UPD PORT NAME(9999) STOP(COMM)
```

Command output:

```
HWSS0770I LISTENING ON PORT=9999    TERMINATED;
M=SSCH
```

Explanation: The port has been updated.

Related reference:

"OPENPORT command" on page 449

"STARTPT command" on page 466

"STOPPORT command" on page 478

UPDATE IMSCON TYPE(PORT) command (Commands)

IMS Connect UPDATE RACFUID command

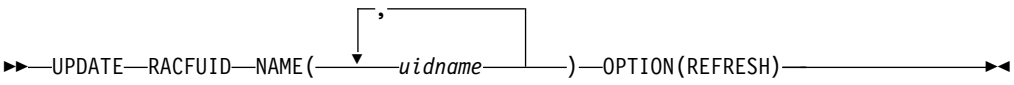
Use the UPDATE RACFUID command to refresh one or more cached IDs stored in the IMS Connect RACF user ID cache.

This command performs the same function as the WTOR command REFRESH RACFUID.

Subsections:

- "Syntax"
- "Keywords"
- "Usage notes" on page 603
- "Equivalent IMS Connect type-2 and WTOR commands" on page 603
- "Examples" on page 603

Syntax



Keywords

The following keywords are valid for the UPDATE RACFUID command:

uidname
Specifies the RACF user ID for IMS Connect to refresh.

OPTION(REFRESH)
Required keyword.

Usage notes

Issue the UPDATE RACFUID command to IMS Connect with a z/OS modify identifier.

Equivalent IMS Connect type-2 and WTOR commands

The following table lists IMS Connect type-2 commands and IMS Connect WTOR commands that perform similar functions as the UPDATE RACFUID command.

UPDATE RACFUID command	Equivalent IMS Connect type-2 command	Equivalent WTOR command
UPDATE RACFUID NAME(*userid*) OPTION(REFRESH)	UPDATE IMSCON TYPE(RACFUID) NAME(*userid*) OPTION(REFRESH)	REFRESH RACFUID NAME(*userid*)

Examples

Command input:

```
F HWS1,UPDATE RACFUID NAME(USRT001)
```

Command output:

```
HWSP1504I RACF USERID USRT001 WAS SUCCESSFULLY REFRESHED,M=SDRC
```

Related concepts:

IMS Connect RACF user ID cache (Communications and Connections)

Related reference:

"REFRESH RACFUID command" on page 453

UPDATE IMSCON TYPE(RACFUID) command (Commands)

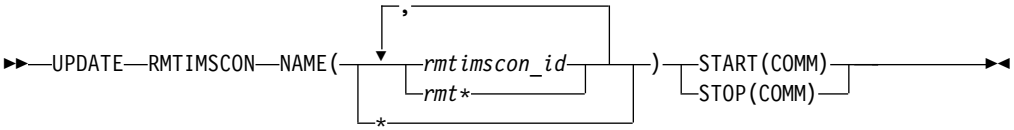
IMS Connect UPDATE RMTIMSCON command

Use the z/OS MODIFY command UPDATE RMTIMSCON in a local IMS Connect instance to start or stop the communications with a remote IMS Connect instance. The command performs similar functions as the WTOR commands STARTRMT and STOPRMT.

Subsections:

- "Syntax" on page 604
- "Keywords" on page 604
- "Usage" on page 604
- "Equivalent IMS Connect type-2 and WTOR commands" on page 605
- "Examples" on page 605

Syntax



Keywords

UPDATE or UPD is used to update the status or attributes of a specified resource. RMTIMSCON represents a connection to a remote IMS Connect instance. The connection is a resource managed by IMS Connect. The command UPDATE RMTIMSCON is used to update the current status of the specified connection.

NAME( )
The name of the remote IMS Connect connection to be updated, as specified on the ID parameter of the RMTIMSCON statement that defines the connection to the local IMS Connect instance. You can specify a single connection, a list of connections separated by commas, or a partial name with a wildcard character. Specify NAME(*) to issue the command against all remote IMS Connect connections defined to the local IMS Connect instance.

START( )
Specifies the attributes to be started.

COMM
Starts communications on the specified remote IMS Connect connections.

STOP( )
Specifies the attributes to be stopped.

COMM
Stops the communication on the specified remote IMS Connect connections.

Usage

When the UPDATE RMTIMSCON command is issued with the START(COMM) keyword, IMS Connect resumes communications on the specified connection to a remote IMS Connect instance. IMS Connect changes the status of the connection to NOT ACTIVE until new sockets are opened for communications with the remote instance of IMS Connect.

The UPDATE RMTIMSCON START(COMM) command is required only when communications on a connection with a remote IMS Connect instance have been previously stopped by either a STOPRMT command or an UPDATE RMTIMSCON STOP(COMM) command.

If the specified RMTIMSCON connection is defined with AUTOCONN=Y, when the UPDATE RMTIMSCON START(COMM) command is issued, IMS Connect automatically creates socket connections to the remote IMS Connect. The number of socket connections that IMS Connect opens is determined by the RESVSOC parameter of the RMTIMSCON configuration statement.

OTMA communications can resume as soon as IMS Connect processes the UPDATE RMTIMSCON START(COMM) command.

For MSC communications, if MSC communications were also stopped in IMS Connect by a STOPMSC command, after you issue the UPDATE RMTIMSCON START(COMM) command, you must also issue either the STARTMSC command or the UPDATE MSC START(COMM) command to reestablish the MSC physical link with the remote IMS Connect.

When the UPDATE RMTIMSCON command is issued with the STOP(COMM) keyword, IMS Connect completes the following operations:

- Stops all communications with a remote IMS Connect instance on the connection specified in the command.
- Informs IMS that the connection to the remote IMS Connect instance has been stopped so that IMS can shut down the affected MSC logical links.
- Terminates existing socket connections and prevents new socket connections until either a STARTRMT command or an UPDATE RMTIMSCON START(COMM) command is issued.
- Changes the status of the connection to STOPPED.
- Issues message HWST3505I.

After the UPDATE RMTIMSCON STOP(COMM) command is issued, if IMS Connect receives a message for the connection in the STOPPED state, IMS Connect returns a NAK response to OTMA and issues message HWST3575W. OTMA retains the message at the front of the queue.

Use the QUERY RMTIMSCON command to display information about the current connections to a remote IMS Connect instance that are defined to the local IMS Connect instance.

Equivalent IMS Connect type-2 and WTOR commands

The following table lists IMS Connect type-2 commands and IMS Connect WTOR commands that perform similar functions as the UPDATE RMTIMSCON command.

UPDATE RMTIMSCON command	Equivalent IMS Connect type-2 command	Equivalent WTOR command
UPDATE RMTIMSCON NAME(*rmtimsconName*) START(COMM)	UPDATE IMSCON TYPE(RMTIMSCON) NAME(*rmtimscon_name*) START(COMM)	STARTRMT *rmtimsconName*
UPDATE RMTIMSCON NAME(*rmtimsconName*) STOP(COMM)	UPDATE IMSCON TYPE(RMTIMSCON) NAME(*rmtimscon_name*) STOP(COMM)	STOPRMT *rmtimsconName*

Examples

Command input for UPDATE RMTIMSCON STOP(COMM):

```
F ICON1,UPDATE RMTIMSCON NAME(CONNECT2) STOP(COMM)
```

Command output for UPDATE RMTIMSCON STOP(COMM):

```
HWST3505I COMMUNICATIONS WITH REMOTE IMS CONNECT CONNECT2 STOPPED; M=TSCH
```

Command input for UPDATE RMTIMSCON START(COMM):

```
F ICON1,UPDATE RMTIMSCON NAME(CONNECT2) START(COMM)
```

Command output for UPDATE RMTIMSCON START(COMM):

```
HWST3500I COMMUNICATIONS WITH REMOTE IMS CONNECT CONNECT2 STARTED; M=TSCH
```

Related reference:

"STARTRMT command" on page 467
"STOPRMT command" on page 480

UPDATE IMSCON TYPE(RMTIMSCON) command (Commands)

Part 7. IMS control region commands

Use the IMS control region commands to work with the IMS region.

- Chapter 25, “MODIFY imsproc commands,” on page 609

© Copyright IBM Corp. 1974, 2017

Chapter 25. MODIFY imsproc commands

Use the MODIFY (F) *imsproc* commands to control an IMS region.

Related reference:

Chapter 29, "TRACE CT command," on page 635

F jobname,DUMP

The `MODIFY (F) jobname,DUMP` command forces a termination of IMS with an offline dump.

Syntax

Usage notes

For DBCTL, the following are abended, but are not rolled back until the following /ERESTART command:

- Active threads
- BMPs
- DEDB utilities

Example

Entry z/OS SC:

```
F jobname,DUMP
```

Response z/OS SC:

```
DFS628I ABNORMAL TERMINATION SCHEDULED
DFS629I IMS STM TCB ABEND - IMS 0020
```

F jobname,DUMPxxxx

The `MODIFY (F) jobname,DUMP`*xxxx* command causes one or more attached external subsystem connections to be terminated with a dump. "External subsystem" here means a non-CCTL subsystem.

Syntax

Usage notes

If *xxxx* is specified as ESS, the parent external subsystem task and all external subsystem subtasks are terminated. If *xxxx* is a specific external subsystem name, only that external subsystem is terminated. The subsystem specified must be

© Copyright IBM Corp. 1974, 2017

defined in the IMS external subsystem PROCLIB member.

Examples

Here are some examples of using the F jobname,DUMPxxxx command.

Example 1 for F jobname,DUMPxxxx

Entry z/OS SC:

```
F jobname,DUMPname
```

Response z/OS SC:

```
DFS628I ABNORMAL TERMINATION SCHEDULED
DFS629I IMS ESI TCB DUMP - IMS 0020 IES
DFS3611I EXTERNAL SUBSYSTEM name CONNECTION TERMINATED
```

Example 2 for F jobname,STOPxxxx

Entry z/OS SC:

```
F jobname,DUMPESS
```

Response z/OS SC:

```
DFS628I ABNORMAL TERMINATION SCHEDULED
DFS629I IMS ESS TCB DUMP - IMS 0020
DFS629I IMS ESS TCB DUMP - IMS 4095 IES
DFS3611I EXTERNAL SUBSYSTEM name CONNECTION TERMINATED
```

F jobname,FORCExxxx

The MODIFY (F) jobname,FORCE*xxxx* command causes one or more attached external subsystem connections to be terminated without a dump.

Syntax

Usage notes

If *xxxx* is specified as ESS, the parent external subsystem task and all external subsystem subtasks are terminated. If *xxxx* is a specific external subsystem name, only that external subsystem is terminated. The subsystem specified must be defined in the IMS external subsystem PROCLIB member.

The FORCE option is not applicable to any other IMS task, for example, to the control task. IMS does not communicate with the external subsystem termination exits, for example, terminate subsystem exits and terminate identify exits.

The reason IMS does not communicate is to enable the installation to break the IMS-to-other-subsystem connection in the event of an error (such as a WAIT STATE) in the external subsystem. If IMS were to invoke one of the exits, as it does with the STOP/DUMP options, the exit can once again start a WAIT STATE, and not enable the termination of that subsystem connection.

Use the FORCE option when either STOP or DUMP fails to break the subsystem connection.

Examples

Here are some examples of using the `F jobname,FORCExxxx` command.

Example 1 for F jobname,FORCExxxx

Entry z/OS SC:

```
F jobname,FORCEname
```

Response z/OS SC:

```
DFS628I ABNORMAL TERMINATION SCHEDULED
DFS629I IMS ESI TCB DUMP - IMS 0020 IES
DFS3611I EXTERNAL SUBSYSTEM name CONNECTION TERMINATED
```

Example 2 for F jobname,FORCExxxx

Entry z/OS SC:

```
F jobname,FORCEESS
```

Response z/OS SC:

```
DFS628I ABNORMAL TERMINATION SCHEDULED
DFS629I IMS ESS TCB DUMP - IMS 0020
DFS629I IMS ESS TCB DUMP - IMS 4095 IES
DFS3611I EXTERNAL SUBSYSTEM name CONNECTION TERMINATED
```

F jobname,RECONNECT

The `MODIFY (F) jobname,RECONNECT` command causes the IMS system on which it is entered to be reconnected to a restarted IRLM.

Syntax

Usage notes

jobname is the job name or procedure name for the IMS control region. When this command is successfully completed, a message indicating that the IRLM has been reconnected is sent to both the IMS master terminal operator and the z/OS console operator. Then the IMS MTO can start any Fast Path DEDB areas that were involved in block-level sharing at the time of the IRLM failure.

Example

In this example, two IMS subsystems are connected to the same IRLM when it terminates.

z/OS SYSTEM SC RECEIVES:

```
DXR121I KRLM212 END-OF-TASK CLEAN-UP SUCCESSFUL - HI-CSA  410K -
   HI-ACCT-CSA   0K
```

IMS 1 MT RECEIVES:

```
DFS2500I IMSA DATABASE dbname SUCCESSFULLY DEALLOCATED
```

IMS 2 MT RECEIVES:

```
DFS2500I IMSB DATABASE dbname SUCCESSFULLY DEALLOCATED
```

z/OS SYSTEM SC:

```
S KRLM1
```

z/OS SYSTEM SC RECEIVES:

```
DXR180I KRLM212 AUTOMATIC RESTART MANAGER IS NOT ENABLED
                OR
     DXR172I KRLM212 armelementname  ARM READY COMPLETED

     DXR117I KRLM212 INITIALIZATION COMPLETE
```

z/OS SYSTEM SC:

```
F jobname,RECONNECT
```

IMS 1 MT AND z/OS SYSTEM SC RECEIVE:

```
DFS626I KRLM RECONNECT COMMAND SUCCESSFUL. IMSA
```

z/OS SYSTEM SC:

```
F jobname,RECONNECT
```

IMS 1 MT AND z/OS SYSTEM SC RECEIVE:

```
DFS626I KRLM RECONNECT COMMAND SUCCESSFUL. IMSB
```

IMS MT:

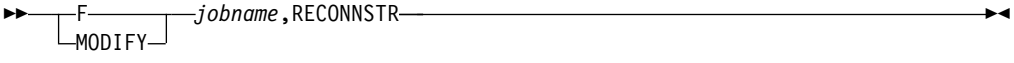
```
/START AREA areaname
```

Explanation: The IRLM has abnormally terminated. When the IRLM has been successfully restarted, the z/OS operator modifies the IMS systems to reconnect to the restarted IRLM. The areas can then be restarted using the /START AREA command.

F jobname,RECONNSTR

Use this command to cause the IMS subsystem on which you enter it to reconnect to IRLM, OSAM, and VSAM coupling facility structures.

Syntax



Usage notes

jobname is the job name or procedure name for the IMS control region.

Example

Entry z/OS SC:

```
F jobname,RECONNSTR
```

Explanation: If the connection to IRLM, OSAM, or VSAM structures has abnormally terminated, usually IMS automatically reconnects to these structures. If IMS does not reconnect to the structures, the operator can use this command to ensure IMS reconnects to them.

F jobname,STOP

The MODIFY (F) jobname,STOP command forces a termination of IMS without a dump.

Syntax

Usage notes

Active threads, BMPs, and DEDB utilities are abended, but not rolled back until the following /ERESTART command.

Normally, IMS is terminated with an IMS /CHECKPOINT shutdown command. Whenever it becomes necessary to force termination of IMS, the z/OS MODIFY command should be used. When MODIFY is used, a dump of the IMS control region can be requested. (Consult the appropriate z/OS operator's guide for specific MODIFY formats.)

Example

Entry z/OS SC:

```
F jobname,STOP
```

Response z/OS SC:

```
DFS628I ABNORMAL TERMINATION SCHEDULED
DFS629I IMS STM TCB ABEND - IMS 0020
```

F jobname,STOPxxxx

The MODIFY (F) jobname,STOP*xxxx* command causes one or more attached external subsystem connections to be terminated without a dump. “External subsystems” here refer to non-CCTL subsystems.

Syntax

Usage notes

If *xxxx* is specified as ESS, the parent external subsystem task and all external subsystem subtasks are terminated. If *xxxx* is a specific external subsystem name, only that external subsystem is terminated. The subsystem specified must be defined in the IMS external subsystem PROCLIB member.

When this command is issued, MPP regions must be brought down and back up again (recycled).

Examples

Here are some examples of using the `F jobname,STOPxxxx` command.

Example 1 for F jobname,STOPxxxx

Entry z/OS SC:

```
F jobname,STOPname
```

Response z/OS SC:

```
DFS628I ABNORMAL TERMINATION SCHEDULED
DFS629I IMS ESI TCB DUMP - IMS 0020 IES
DFS3611I EXTERNAL SUBSYSTEM name CONNECTION TERMINATED
```

Example 2 for F jobname,STOPxxxx

Entry z/OS SC:

```
F jobname,STOPESS
```

Response z/OS SC:

```
DFS628I ABNORMAL TERMINATION SCHEDULED
DFS629I IMS ESS TCB DUMP - IMS 0020
DFS629I IMS ESS TCB DUMP - IMS 4095 IES
DFS3611I EXTERNAL SUBSYSTEM name CONNECTION TERMINATED
```

Part 8. IRLM commands

Use the IRLM commands to control the IRLM component.

- Chapter 26, “MODIFY irlmproc commands,” on page 617
- Chapter 27, “START irlmproc command,” on page 629
- Chapter 28, “STOP irlmproc command,” on page 633
- Chapter 29, “TRACE CT command,” on page 635

© Copyright IBM Corp. 1974, 2017

Chapter 26. MODIFY irlmproc commands

The MODIFY (F) *irlmproc* commands are used to display the status of an IRLM and abnormally terminate an IRLM.

These commands can be issued only from a z/OS console.

F irlmproc,ABEND

The MODIFY *irlmproc,*ABEND command is used to abnormally terminate IRLM, whether or not any IMS subsystems are identified to IRLM.

Syntax

Usage notes

The *irlmproc* identifies the procedure name of the IRLM to be terminated. Prior to terminating, IRLM initiates all active IMS STATUS exit routines to inform them of the pending termination.

If the NODUMP parameter is specified, IRLM terminates without generating a dump. If the NODUMP parameter is not specified, the system dump is taken to the SYS1.DUMPxx data set.

Recommendation: If IRLM does not terminate because of outstanding IMS requests in process (the DXR011I irlmproc END-OF-TASK CLEAN-UP SUCCESSFUL message is not received), use the following methods (in this order) to terminate IRLM:

1. Issue the `MODIFY irlmproc,ABEND,NODUMP` command to terminate IRLM without attempting to clean up the outstanding IMS requests.
2. Issue the z/OS CANCEL command.
3. Issue the z/OS FORCE command. This method is least preferred because it might prevent the irlmproc from completing the cleanup required to permit the subsystem to be restarted.

Examples

Here are some examples of using the F *irlmproc,*ABEND command.

Example 1 for F irlmproc,ABEND

z/OS SYSTEM 1 SC:

```
F IR21J,ABEND
```

Response SYSTEM 1 SC: (partial messages)

```
DXR124E JR21212 ABENDED VIA MODIFY COMMAND
DUMPID=001 REQUESTED BY JOB (IR21J   )
IEA794I SVC DUMP HAS CAPTURED: 857
```

© Copyright IBM Corp. 1974, 2017

```
DUMP TITLE=JR21 ESTAE ENTERED. ABEND U2020 MODULE DXRRL020+0A98
     APAR PQ15432  1999/007 14:14:03
IXL030I CONNECTOR STATISTICS FOR LOCK STRUCTURE LOCK2, 860
  CONNECTOR IRLMGRPR$JR21212:
IXL031I CONNECTOR CLEANUP FOR LOCK STRUCTURE LOCK2, 861
  CONNECTOR IRLMGRPR$JR21212, HAS COMPLETED.
DXR121I JR21212 END-OF-TASK CLEANUP SUCCESSFUL - HI-CSA    410K  -
HI-ACCT-CSA      0K
IEF450I IR21J IR21J - ABEND=S000 U2020 REASON=00000000
IEF352I ADDRESS SPACE UNAVAILABLE
$HASP395 IR21J    ENDED
```

Response SYSTEM 2 SC: (partial messages)

```
DXR137I IR21001 GROUP STATUS CHANGED. JR21 212 HAS BEEN DISCONNECTED
FROM THE DATA SHARING GROUP
```

Explanation: The operator on system 1 has terminated the procedure named IR21J. The operator on system 2 is informed that the IRLM on system 1 has failed.

Example 2 for F irlmproc,ABEND

z/OS SYSTEM 2 SC:

```
F IR21J,ABEND,NODUMP
```

Response SYSTEM 2 SC: (partial messages)

```
DXR165I JR21212 TERMINATED VIA IRLM MODIFY COMMAND.
  IXL030I CONNECTOR STATISTICS FOR LOCK STRUCTURE LOCK2, 976
  CONNECTOR IRLMGRPR$JR21212:
  ....
   IXL020I CLEANUP FOR LOCK STRUCTURE LOCK2, 980
      CONNECTION ID 02, STARTED BY CONNECTOR IRLMGRPR$IR21001
   IXL021I GLOBAL CLEANUP FOR LOCK STRUCTURE LOCK2, 981
      CONNECTION ID 02, BY CONNECTOR IRLMGRPR$IR21001
      HAS COMPLETED.
   DXR121I JR21212 END-OF-TASK CLEANUP SUCCESSFUL - HI-CSA    343K  -
      HI-ACCT-CSA      0K
   IEF352I ADDRESS SPACE UNAVAILABLE
   $HASP395 IR21J    ENDED
```

Response SYSTEM 1 SC: (partial messages)

```
DXR137I IR21001 GROUP STATUS CHANGED. JR21 212 HAS BEEN DISCONNECTED
      FROM THE DATA SHARING GROUP
```

Explanation: The operator on system 2 has terminated the procedure named IR21J and has suppressed the SDUMP dump. The operator on system 1 is informed that the IRLM on system 2 has failed.

F irlmproc,DIAG,HANG

HANG is used under the direction of IBM service for collecting IRLM sysplex dumps when DEADLOCK or TIMEOUT issues are suspected.

Syntax

Usage notes

This command takes dumps during DEADLOCK processing and causes DEADLOCK processing to STOP while the dynamic deadlock storage is collected. z/OS DUMP services then schedules an SRB to restart DEADLOCK processing. DXR183I is issued by each IRLM as the SRB resumes IRLM deadlock.

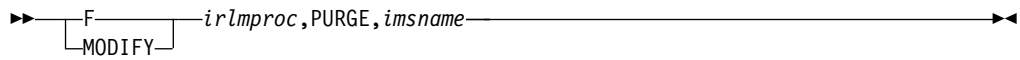
If this message is not issued, that IRLM must be terminated and restarted.

Note: Always start the IRLM SCF CTRACE internally and wait 30 seconds before issuing this command.

F irlmproc,PURGE,imsname

The MODIFY (F) *irlmproc*,PURGE,*imsname* command releases IRLM locks retained for a specific IMS subsystem because of an IMS, IRLM, or system failure. The command causes STATUS exit routines to be activated for each identified IMS.

Syntax



Usage notes

This command is used in the following recovery situations:

- Database cleanup was performed and reflected in the RECON data set, but the IRLM was not made aware of it. This typically occurs when a batch backout is performed before the IRLMs are reconnected. After the PURGE command is issued, data that was locked is made available to the active IMS systems.
- Recovery will be performed later and the databases must be unavailable for use by other IMS systems.

Restriction: Ensure that IMS is inactive (down) prior to issuing this command. In a DBCTL environment, use the /CHANGE CCTL command when IMS is active (up).

The *imsname* is the IMS name displayed by the STATUS command. The *irlmproc* identifies the procedure name for each IRLM. If multiple IRLMs exist in the same system, each procedure must have a unique procedure name. The *irlmproc* must be a procedure name of an active IRLM which is connected to the same sysplex group as the failed member. Issuing a purge request using an inactive IRLM, will cause the error, `IEE341I irlmproc NOT ACTIVE`.

Example

Here are some examples of using the F *irlmproc*,PURGE,*imsname* command.

To clean up the retained locks from a down IMS (for example, IMS1), first check if the irlmproc (for example, ims1irlm) is up by issuing the following command:

```
D A,L
```

If the IRLM is up (active), check the status of the IMS subsystem, by issuing the following command:

```
F ims1irlm,STATUS
```

If IMS is down (inactive), then issue the following PURGE command:

```
F ims1irlm,PURGE,ims1
```

In a sysplex environment, if the IMS is down and its IRLM is stopped or disconnected, you must use one of the other active IRLM members to query retained locks and issue the following PURGE command:

```
F irlmproc,STATUS,ALLD
```

After you issue the PURGE command, you must issue a purge request on the system with an active IRLM. For example, if system B is up on an IMS2 subsystem with an irlmproc named ims2irlm, issue the following command to display all up and down subsystems in a data-sharing sysplex:

```
F ims2irlm,STATUS,ALLD
```

Then, if the IMS2 subsystem is down, issue the PURGE request, by entering the following command:

```
F ims2irlm,PURGE,ims2
```

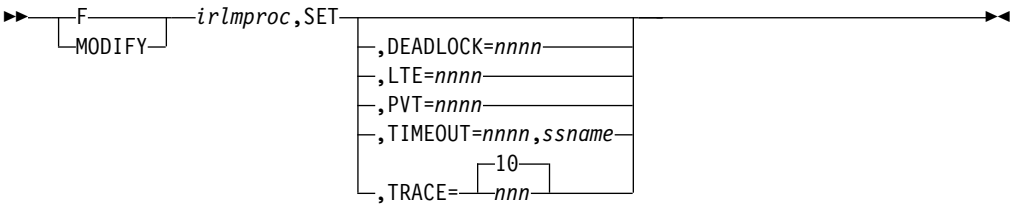
The response on the z/OS system console for completed purge request is:

```
DXR109I IRS2002 PURGE COMMAND COMPLETED FOR IMS1
```

F irlmproc,SET

The Modify (F) irlmproc,SET command dynamically alters a limited set of parameters known to IRLM.

Syntax



Usage notes

The following keywords can be specified for the F irlmproc,SET command.

irlmproc
Specifies the IRLM that is to process the command.

DEADLOCK= *nnnn*

nnnn The number in milliseconds to be used for the local deadlock frequency. The value range is 100 - 5000 milliseconds. If a member of a sysplex group and all IRLMs are not enabled for subsecond deadlock processing, DXR106E is issued.

When the IRLM supporting subsecond deadlock joins a group which has a member which does not support subsecond deadlock, the value range used for the new member joining, is 1 - 5 seconds. All members supporting subsecond processing will issue the DXR177I message after changing the timer value.

Note: When IMS TIMEOUT candidates have timed out, they remain timeout candidates and are presented to the timeout exit each Global deadlock cycle. IMS creates SMF 79.15 records when candidates are presented. They are then written to the SMF data sets, if enabled. If timeout candidates are found and the value for *nnnn* is subsecond, there will be many SMF 79.15 records written per second until the tasks are no longer waiting in IRLM.

LTE=*nnnn*

nnnn The number of Lock Table Entries to be specified on the next CONNECT to the z/OS cross-system coupling facility LOCK structure. The value range must be 0 - 1024 and be an even power of 2. Each increment in value represents 1,048,576 Lock Table Entries.

Any syntax error in issuing the command will receive DXR106E. Syntax errors include LTE value out-of-range. If this IRLM is not connected to the group, and the value is valid, it will issue DXR177I, but the value will not be sent to any other member. If the member is already in the group, the value is sent to the Global Deadlock Manager (GDM) IRLM to be broadcast to all other members. If the GDM does not have the code applied, no DXR177I response will be issued on any member. If the GDM has the code, then all members with the code applied will issue the DXR177I as the command is processed.

This value is used only if the IRLM is the first to join the data-sharing group causing structure allocation or during a REBUILD. If any IRLM joins later, they will not have the updated value. If multiple MODIFY commands are issued on the same or multiple IRLMs, some DXR177I might be missing. The last DXR177I issued is the value to be used on the next CONNECT.

The value for the number of Lock Table Entries specified during normal group startup and during REBUILD are used in the following order:

1. The value specified on the MODIFY irlmproc,SET,LTE= command if it is greater than zero.
2. The value from the LTE= in the IRLMPROC if it is greater than zero.
3. The existing logic, which determines the nearest power of 2 after dividing the QUERY size returned by 2 times LTE width based on MAXUSRS.

If an attempt is made to use a nonzero value from either step 1 or 2 and that value is too large for the structure size returned on the QUERY, then the value from the next lower-order sequence is used.

PVT=*nnnn*

Specifies the upper limit of private storage, below the two-gigabyte bar. *nnnn* must be a four-digit number from 1 through 1800. You can specify this value in megabytes or gigabytes by specifying M (for megabytes) or G (for gigabytes) after the value, as follows, *nnnn* M or *nnnn* G. IRLM monitors the amount of private storage used for locks. If the specified limit is reached, new lock requests are rejected unless they are “must complete”. If the specified value is out of range or if IRLM's use of private storage is already larger than the specified value, the command is rejected with message DXR106E. No reserve for “must complete” locks is calculated from the specified PVT= value.

TIMEOUT=*nnnn,ssname*

nnnn The new timeout value in the range of 1 - 3600 seconds.

ssname
The identified subsystem name as displayed by the MODIFY irlmproc,STATUS command.

Any syntax error in issuing the command will receive DXR106E. Syntax errors include TIMEOUT value out-of-range or invalid identified subsystem name. A syntax error message will also be given if the DXR177I message has not been received for the prior command completion.

The TIMEOUT value must be a multiple of the local deadlock parameter. If the value entered is not an even multiple of the local deadlock parameter, IRLM increases the timeout value to the next highest multiple. The value used by IRLM for timeout is displayed in the DXR177I message, which is issued during deadlock processing. This new value is used until the IRLM or identified subsystem is terminated, or the timeout is changed again by the operator. The value specified on the command does not affect the timeout value in the DB2® ZParms.

TRACE=*nnn*
Requests that IRLM dynamically sets the maximum number of 64 KB trace buffers per trace type to the value you specify in *nnn*. This value is used only when the external CTRACE writer is not activated. The trace buffers are allocated from ECSA. Valid values are from 10 through 255. If you specify a value outside of this range, IRLM automatically adjusts the value to a value within the range. The default is 10. IRLM does not immediately acquire the number of trace buffers you set, but allocates buffers as needed, not to exceed the number of buffers you set in this command. If the number of trace buffers you set is less than the number of currently allocated buffers, IRLM brings the number to within your specified range by releasing the oldest buffers at the end of the next deadlock or timeout cycle.

Restrictions:

1. Do not modify the CSA value without first contacting the systems programmer to determine the amount of CSA storage that can be used for IRLM.
2. The values you set in the MODIFY command do not persist through a stop and restart of IRLM. The number of trace buffers for each trace type returns to the default value of 10, and the value for MAXCSA returns to the value you set for the MAXCSA parameter of the IRLM startup procedure.

Examples

Here are some examples of using the F *irlmproc*,SET command.

Example 1 for F irlmproc,SET

Entry on a z/OS system console:

```
F IRLMPROC,SET,TRACE=20
```

Response on z/OS system console:

```
DXR177I IRLMX THE VALUE FOR TRACE IS SET TO 20
```

Example 2 for F irlmproc,SET

Entry on a z/OS system console:

```
F IRLMPROC,SET,TIMEOUT=60,DBMS
```

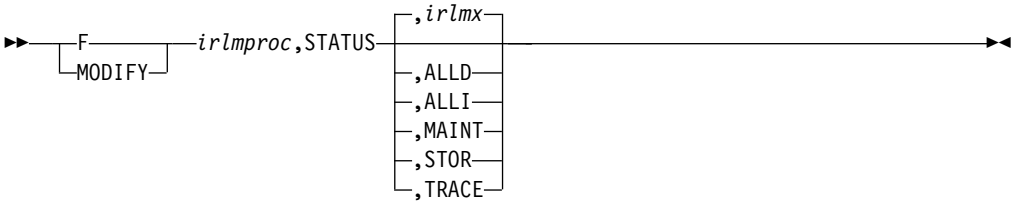
Response on z/OS system console:

```
DXR177I IRLMX THE VALUE FOR TIMEOUT IS SET TO 60 FOR DBMS
```

F irlmproc,STATUS

The MODIFY (F) irlmproc,STATUS command is used to display the status of an IRLM.

Syntax



Usage notes

The following keywords can be specified for the `F irlmproc,STATUS` command.

The *irlmproc* identifies the IRLM that is to process the command.

irlmx
Specifies which IRLM's status is to be displayed. The *irlmx* parameter is the concatenation of the IRLM subsystem name (IRLMN) and a three-digit IRLM member ID (IRLMID) specified in the IRLM startup procedure. An example is DB2G002 (ID is 002).

ALLD
Requests the DBMS names and status of subsystems, active or failed, on any IRLM in the data sharing group. This keyword also shows any RETAINED LOCKS the systems might own and the IRLM they are currently associated with.

ALLI
Requests the IRLM names and status for all IRLMs in a data sharing group.

MAINT
Allows the operator to display the IRLM maintenance levels for IRLM load modules, with the exceptions of: DXRRLM50, DXRRL183, DXRRL186, DXRRLFTB, and in a non-sysplex environment DXRRLM70.

STOR
For this IRLM, displays the current and maximum specified allocation for CSA and ECSA storage.

TRACE
For this IRLM, requests information about IRLM subcomponent trace types. The information returned includes whether a subcomponent trace type is active, how many trace buffers are used by the trace, and whether the component trace external writer is active for the trace.

You can activate or deactivate traces by using the `TRACE CT` command of z/OS. You cannot turn off the EXP and INT traces. The XIT (for data sharing), EXP, and INT traces are automatically activated when you start IRLM. Traces are automatically activated with `IRLMPROC TRACE=YES`.

Message DXR001I is issued if the *irlmx* parameter is omitted or specifies the IRLM receiving the command. This multiline message contains work unit and lock information for each IMS identified to this IRLM.

Message DXR001I is also issued if the *irlmx* parameter is specified, but the IRLM receiving the command has no knowledge of an IRLM with an IRLMID of *irlmx*.

Message DXR102I is issued if *irlmx* is specified. This multiline message contains the IMS names and status of each IMS subsystem identified to the IRLM with an IRLMN and IRLMID of *irlmx*. Message DXR102I is also issued if ALLD is specified. It contains the names and status of each IMS subsystem identified to the IRLM in the data sharing group.

Message DXR103I is issued if ALLI is specified. This multiline message contains the IRLM names and status of all IRLMs in the data sharing group.

Examples

Here are some examples of using the `F irlmproc,STATUS` command.

Example 1 for F irlmproc,STATUS

Assume you are in a non-data sharing environment. Enter on a z/OS system console:

```
F IRTPROC,STATUS
```

Response on the z/OS system console:

```
DXR101I IR2T001 STATUS SCOPE=LOCAL
        SUBSYSTEMS IDENTIFIED                PT01
        NAME      STATUS    UNITS     HELD      WAITING     RET_LKS
        DSNT1     UP-NS     0005      0010      0002        0
```

Explanation: The operator on system 1 has requested information about the IMS systems connected to the IRLM identified by the IRLM procedure named IRTPROC.

Example 2 for F irlmproc,STATUS

Assume you have a data sharing group. Enter on a z/OS system console:

```
F DB1GIRLM,STATUS,ALLD
```

Response on the z/OS system console:

```
  14.02.10 STC00086  DXR102I DJ1G001 STATUS IRLMID=001
          SUBSYSTEMS IDENTIFIED                PT01
          NAME     STATUS    RET_LKS   IRLMID  IRLM_NAME  IRLM_LEVL
          DB4G     UP              0    004      DJ4G        1.009
          DB3G     UP              0    003      DJ3G        1.009
          DB2G     UP              0    002      DJ2G        1.009
          DB1G     UP              0    001      DJ1G        1.009
```

Explanation: The output shows all the IMS subsystems that are connected to IRLMs in this data sharing group (the group to which the IRLM processing the request belongs). Other possible information include:

STATUS
Indicates the status of the IMS subsystem:

UP The IMS is active.

DOWN
The IMS is failed.

UP-RO
IMS is active and is currently identified to IRLM as a read-only subsystem.

CLEANUP
IRLM is waiting for the IMS subsystem to respond that cleanup has completed after a failure condition.

SFAIL The IRLM that IMS is identified to has been disconnected from the data sharing group. Any "modify" type locks held by IMS have been retained by IRLM.

RET_LKS
The number of retained locks that are owned by this IMS subsystem.

IRLMID
The ID of the IRLM to which this IMS subsystem is identified.

IRLM_NAME
The name of the IRLM to which this IMS subsystem is identified.

IRLM_LEVL
The IRLM release and function level that this IMS subsystem requested on the identify to IRLM. This is in the form of *r.fff* .

Example 3 for F irlmproc,STATUS

Again, assume data sharing is in effect. Enter the following on a z/OS system console:

```
F DB1GIRLM,STATUS,ALLI
```

Response on the z/OS console is:

```
DXR103I IR21010 STATUS
IRLMS PARTICIPATING IN DATA SHARING GROUP FUNCTION LEVEL=024
IRLM_NAME IRLMID STATUS LEVEL SERVICE MIN_LEVEL MIN_SERVICE
  IRLM     001     UP    096  PQ92110    022      PQ52360
  IR21     010     UP    024  PQ92110    012      PN90337
  JRLM*    002     UP    096  PQ92110    022      PQ52360
DXR103I End of display
```

Explanation: The output shows the IRLMs that are participating in this data sharing group (the group that includes the IRLM processing the request). An asterisk (*) in the IRLM_NAME field indicates that this IRLM is the Global Deadlock Manager (GDM). Other information includes:

STATUS
The value "UP" in the STATUS field indicates that the IRLM is active. STATUS shows "DOWN" if the IRLM is failed. An IRLM is known to be "DOWN" only if the IMS subsystem that was identified to it holds retained locks. This connection between a failed IMS subsystem and IRLM is lost after a REBUILD or a group restart.

LEVEL
The current IRLM release and function level in the form of *r.fff* .

SERVICE
The IRLM service or release that corresponds to the release and function level that is given in "LEVEL".

MIN_LEVEL
The minimum IRLM function level with which this IRLM can coexist.

MIN_SERVICE
The IRLM service or release that corresponds to the function level given in "MIN-LEVEL."

Group Function Level
The IRLM release and function level that is in use by all the IRLMs in this data sharing group.

Example 4 for F irlmproc,STATUS

Enter the following command on a z/OS system console:

```
F IRLME2N,STATUS,STOR

DXR100I IRLM241 STOR STATS
PC: YES  LTEW:  2 LTE:     0M RLE:    2646  RLEUSE:       0
BB PVT:  1500M  AB PVT (MEMLIMIT):  16383P
CSA USE: ACNT:     0K  AHWM:     0K  CUR:  1057K  HWM:  1057K
        ABOVE 16M:    27   1057K     BELOW 16M:     0      0K
PVT USE:   BB CUR:  7453K        AB CUR:     5M
CLASS   TYPE  SEGS     MEM   TYPE  SEGS     MEM   TYPE  SEGS     MEM
ACCNT    T-1     2      4M    T-2     1      1M    T-3     1      4K
PROC     WRK     4     20K    SRB     1      1K    OTH     1      1K
MISC     VAR    20   6086K    N-V    12    323K    FIX     1     24K
    DXR100I End of display
```

Explanation: The example shows that current storage allocated for IRLM is 1057 KB, and the greatest amount that has been allocated since the last time IRLM was started is 1057 KB. The storage for the locking structures (RHB and RLB) is contained within IRLM private storage. Use the following information to interpret the display output:

PC The current value for the PC option of the IRLM startup procedure.

BB PVT
The current amount of private storage below the bar. The PVT value is 1500 MB in this example.

AB PVT (MEMLIMIT)
The current limit to private storage above the bar, also known as the MEMLIMIT, which is managed by MVS. This storage is used for locks. The AB PVT value is 16383 PB in this example.

LTE The number of the lock table entries that were available in the coupling facility the last time this IRLM was connected to the group. Each unit consists of 1 048 576 entries. If LTE is less than one unit, the value will be zero.

LTEW The lock table entry width is the number of bytes needed for each lock table entry in the lock structure. The more users in the group, the more bytes are needed to manage each lock table entry. For up to 6 users, the LTEW is 2, for up to 22 users, the LTEW is 4, and for up to 32 users, the LTEW is 8.

RLE The number of the record table entries that were available in the coupling facility the last time this IRLM was connected to the group.

RLEUSE
The number of the RLEs that were in use in the coupling facility at the time you issued the MODIFY command. If the IRLM is disconnected from the CF, this number represents the RLE that were in use when the IRLM last updated prior to DISCONNECT.

CSA USE
CSA USE is displayed for compatibility reasons only.

ACNT When PC=YES, this value is the accountable extended private storage used for lock control structures. This value is specified as either *xxx*K for kilobytes or *xxx*M for megabytes.

AHWM
When PC=YES, this value is the High Water Mark or the greatest amount of extended private storage allocated by IRLM during this initialization period for lock control structures. Its value is specified as either xxxK for kilobytes or xxxM for megabytes.

CUR The total current CSA and ECSA usage. In this case, the current usage (CUR) is 1057 KB, and the high water mark (HWM) is 1057 KB. The accountable storage is a subset of this total storage.

PVT USE
The current below-the-bar and above-the-bar private usage.

BB CUR
The current below-the-bar private usage. The value is specified as either *xxxx*K for kilobytes or *xxxx*M for megabytes.

AB CUR
The current above-the-bar-private usage. The value is specified as either *xxxx*M for megabytes or *xxxx*G for gigabytes.

ACCNT
The ACCNT row of the report is a breakdown of lock control block structures and their storage use.

T-1 Type one structures are for resources. In this case, it shows that two storage segments are held for a total of 4 MB.

T-2 Type two structures are for all resource requests after the first request for a specific resource. This example shows that one storage segment is held for a total of 1 MB.

T-3 Type three structures are for requesters or work units that are waiting for or hold resources. This example shows that one storage segment is held for a total of 4 KB.

PROC and MISC rows
These rows contain usage information for private storage that is used to process IMS requests. Use this information under the guidance of IBM Software Support for diagnosing problems.

Example 5 for F irlmproc,STATUS

When the following command is entered on a z/OS system console:

```
F PR21PROC,STATUS,TRACE
```

The response on the z/OS console is:

```
DXR179I PR21034 TRACE USAGE
TRACE BUFFER STORAGE IN USE:   256KB
MAXIMUM NUMBER OF TRACE BUFFERS ALLOWED PER TRACE TYPE:  10
TRACE TYPE   ACTIVE   BUFFERS IN USE   CTRACE WRITER
----------   ------   --------------   -------------
   SLM          N            0              N
   XIT          Y            2              N
   XCF          N            0              N
```

```
DBM          N            0              N
EXP          Y            1              N
INT          Y            1              N
```

The example shows the storage currently allocated for IRLM tracing is 256 KB, the maximum number of trace buffers allowed per trace type is set to 10, and the external CTRACE writer is not active. The trace size for each buffer is 64 KB. You can change the maximum number of trace buffers that can be used by issuing the following command: MODIFY *irlmproc*,SET,TRACE=*nnn*.

Chapter 27. START irlmproc command

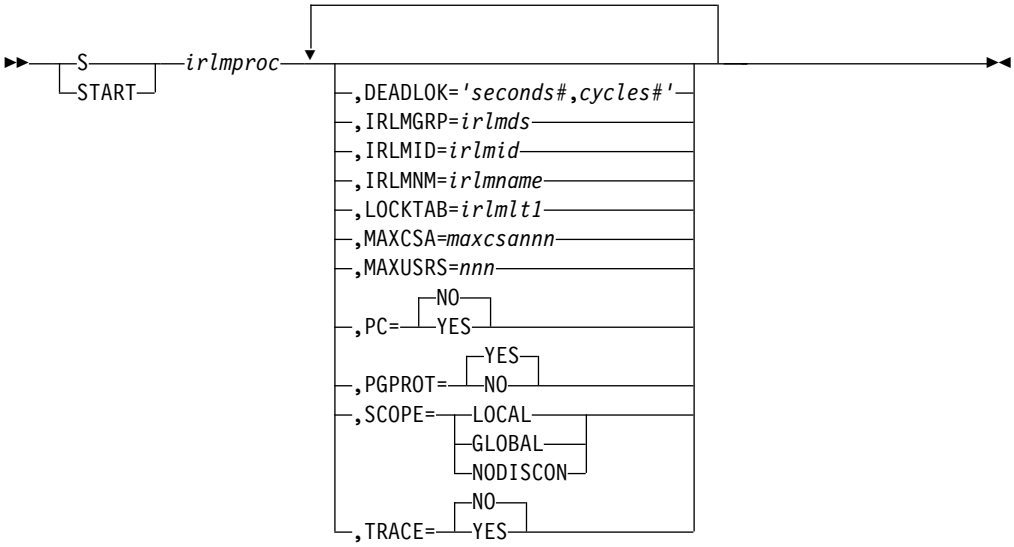
The START *irlmproc* command starts an IRLM component with a procedure put in place by your installation.

Symbolic parameters in the procedure can be overridden using the START *irlmproc* command. IRLM must be started before any IMS subsystem that needs the facilities provided by IRLM.

Subsections:

- "Syntax"
- "Keywords"

Syntax



Keywords

The following keywords can be specified for the START *irlmproc* command.

The *irlmproc* identifies the procedure name of the IRLM to be started.

Any IRLM parameter can be overridden using the START command.

DEADLOK=
Specifies the local deadlock-detection interval (in seconds), and the number of local cycles that are to occur before a global detection is initiated. You can specify the following:

seconds#
This 1- to 4-digit numeric value, ranging from 1 to 9999, specifies the interval (in seconds) between IRLM local deadlock-detection cycles. Although values above 5 are accepted, the maximum used is 5.

© Copyright IBM Corp. 1974, 2017

Recommendation: Set this parameter to be no greater than 5. This is because it takes multiple local cycles to complete one global cycle. Otherwise, global deadlocks may not be broken frequently enough to ensure maximum throughput.

cycles#
This 1- to 4-digit numeric value, ranging from 1 to 9999, specifies the number of local deadlock cycles that must expire before global deadlock-detection is performed. Although values above 1 are accepted, the maximum used is 1.

Recommendation: Set this parameter to be no greater than 5. This is because it takes multiple local cycles to complete one global cycle. Otherwise, global deadlocks may not be broken frequently enough to ensure maximum throughput.

IRLMGRP=
Specifies the name of the z/OS cross-system coupling facility (XCF) group for which this IRLM will serve as lock manager.

irlmds
The name of the XCF group.

IRLMID=
Specifies either a decimal number or a printable character that is used to distinguish between multiple IRLMs. The IRLM with the lowest ID value in the group becomes the global deadlock manager for the group. You must specify a unique name for every IRLM in the group. The total number of IRLMs that can be included in a group is not related to the value you specify; the total number is limited by your hardware, and may be lower than 255.

The *irlmid* can be either a one- to three-digit number ranging in value from 1 to 255, or a printable EBCDIC character. When you specify a character, you must surround the character with enough single quotation marks to permit IRLM to understand that this value is indeed a printable character. IRLM converts the character to an EBCDIC value between 1 and 255.

Because of the way that z/OS interprets quotation marks, you must use 7 quotation marks on either side of the character.

Example: If you want to specify the printable character 'D', you must specify `IRLMID='''''''D'''''''`.

IRLMNM=
Specifies the 1- to 4-byte z/OS subsystem name assigned to this IRLM.

LOCKTAB
Specifies the lock table to be used by the data sharing group specified in the GROUP parameter.

irlmlt1
Is the name of the lock table.

MAXCSA=
An obsolete keyword that is preserved in the IRLM EXEC statement for compatibility purposes. IRLM ignores this keyword if specified.

MAXUSRS=
Specifies the maximum number of users, from 2 to 248, that can connect to the data sharing group.

PC=NO | YES
An obsolete keyword that is preserved in the IRLM EXEC statement for compatibility purposes. IRLM ignores this keyword if specified.

PGPROT=YES | NO
Tells IRLM whether to place its COMMON STORAGE load modules into z/OS Page Protected Storage. PGPROT=YES is the default and tells IRLM to place its COMMON Storage load modules into z/OS Page Protected Storage. If PGRPROT=NO is specified, IRLM will not place load modules in Page Protected Storage.

SCOPE=
Specifies whether the IRLM is to be used in a data-sharing environment and, if so, whether the IRLM should automatically disconnect when the associated IMS terminates.

LOCAL
Specifies that the IRLM is in a non-data-sharing environment and there is no intersystem sharing.

GLOBAL
Specifies that the IRLM is in a data-sharing environment and that intersystem sharing is to be performed. The IRLM disconnects from the data-sharing group when there are no IMS systems identified to it. Both XCF and SLM are required.

NODISCON
Specifies that the IRLM is in a data-sharing environment and that intersystem sharing is to be performed. The IRLM remains connected to the data-sharing group even when there are no IMS systems identified to it. Normal DISCONNECT from the sharing group does not occur. You must explicitly stop the IRLM when NODISCON is specified. Both XCF and SLM are required.

With NODISCON, there is less impact on other systems when an IMS fails because z/OS is not required to perform certain recovery actions that it normally perform when an IRLM terminates. Another benefit of the NODISCON parameter is that IMS restarts more quickly after a normal or abnormal termination because it does not have to wait for IRLM to rejoin the IRLM data-sharing group.

TRACE=
Specifies whether the IRLM should trace activity during IRLM startup. Traces are written to wrap-around trace buffers, that is, the first buffer is reused after the last buffer is full.

NO Do not initialize trace activity during IRLM startup. NO is the default.

To trace IRLM activity, use the TRACE CT command.

YES
Initialize IRLM trace activity during startup.

Recommendation: Turn on tracing in all data-sharing environments.

Chapter 28. STOP irlmproc command

The STOP (P) *irlmproc* command is used to normally shut down IRLM. The command is rejected if any active IMS subsystems are currently identified to IRLM.

The *irlmproc* identifies the procedure name for the IRLM to be stopped.

Subsections:

- "Syntax"
- "Example"

Syntax

Example

z/OS SYSTEM 1 SC:

```
P IR21I
```

Response SYSTEM 1 SC:

```
DXR110I IR21001 STOP COMMAND ACCEPTED
IXL030I CONNECTOR STATISTICS FOR LOCK STRUCTURE LOCK2, 070
  CONNECTOR IRLMGRPR$IR21001:
IXL031I CONNECTOR CLEANUP FOR LOCK STRUCTURE LOCK2, 071
  CONNECTOR IRLMGRPR$IR21001, HAS COMPLETED.
DXR121I IR21001 END-OF-TASK CLEANUP SUCCESSFUL - HI-CSA    732K  -
   HI-ACCT-CSA    132K
IEF352I ADDRESS SPACE UNAVAILABLE
$HASP395 IR21I    ENDED
```

Response SYSTEM 2 SC:

```
NONE
```

Explanation: The operator on system 1 has terminated the IRLM procedure named IR21I. No operator action on system 2 is required.

© Copyright IBM Corp. 1974, 2017

Chapter 29. TRACE CT command

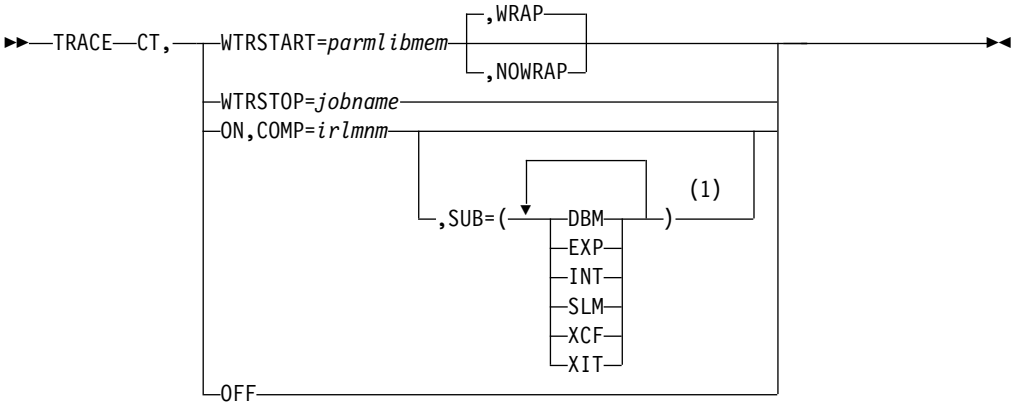
Use the z/OS TRACE CT command to start, stop, or modify an IRLM diagnostic trace.

IRLM does not support all the options available on the TRACE CT command.

Subsections:

- "Syntax"
- "Keywords"
- "Usage notes" on page 637
- "Examples" on page 637

Syntax



Notes:

1 The same trace type can be specified only once.

Keywords

The following keywords can be specified for the TRACE CT command.

CT Specifies the component trace (instead of the other trace options available on the z/OS TRACE command).

WTRSTART=*parmlibmem*
Identifies the name of the member that contains the source JCL that invokes the external writer and defines the data set to which external writer writes the trace buffers. The member can be a SYS1.PROCLIB cataloged procedure or a job. Many installations use a cataloged procedure in SYS1.PROCLIB.

After you enter a TRACE CT,WTRSTART command, turn the trace on and connect the writer using the WTR parameter in the reply for a TRACE CT command.

WRAP
Specifies that when the system reaches the end of the data set or group

© Copyright IBM Corp. 1974, 2017

of data sets, it writes over the oldest data at the start of the data set or the start of the first data set in the group. The primary extents of the data set are used.

NOWRAP
Specifies that the system stops writing to the data set or data sets when they are full. The primary and secondary extents of the data sets are used.

WTRSTOP=*jobname*
Identifies the job name of a component trace external writer to be stopped. The data sets that the writer used are closed.

The job name is either:

- A member name, if the source JCL is a procedure.
- The name on a JOB statement within the source JCL.

ON Turns on the trace.

COMP=*irlmssnm*
The IRLM subsystem name.

SUB=*subname*
Identifies the type of sublevel trace desired. This parameter can be any of the following values:

DBM
Trace interactions with the identified DBMS.

EXP
Trace any exception condition. EXP is automatically turned on when IRLM starts running and is permanently turned on.

INT
Trace member and group events outside of normal locking activity. INT is automatically turned on when IRLM starts running and is permanently turned on.

SLM
Trace interactions with the z/OS locking component.

XCF
Trace all interactions with z/OS Cross-System coupling services.

XIT
Trace just asynchronous interactions with the z/OS locking component. XIT is automatically turned on when IRLM starts running, but it can be turned off using the z/OS TRACE CT command.

OFF
Turns off the trace. If IRLM is connected to a component trace external writer, the system forces an implicit disconnect.

In the case of the EXP and INT sublevel traces, the OFF parameter stops the traces from writing to the external writer. However they continue to write to buffers. These traces are used by IMS for debugging purposes.

Sample external writer procedure

Here is a sample procedure for the IRLM external writer:

```
//CTWTR    PROC
//         EXEC PGM=ITTTRCWR
//TRCOUT01 DD   DSNAME=SYS1.WTR1,DISP=OLD
//TRCOUT02 DD   DSNAME=SYS1.WTR2,DISP=OLD
```

Usage notes

The impact of setting TRACE CT ON is that each active subname type requires up to .7 MB of ECSA. Because IRLM initializes its own traces when it starts, the DISPLAY TRACE command shows that all traces are off. After you issue the TRACE ON command, the reports are accurate except for the two subname types, INT and EXT, which cannot be turned off.

This command can be entered only from the master console. The command requires an appropriate level of z/OS authority.

Because this command uses z/OS component trace services, make sure that the IRLM start and stop load module, DXRRL183, is in the z/OS link list.

Examples

Here are some examples of using the TRACE CT command.

Example 1 for TRACE CT command

Here is an example sequence of commands to start and stop an IRLM DBM trace. In this example, the trace data is written to an external writer data set identified in procedure CTWTR:

```
TRACE CT,WTRSTART=CTWTR
TRACE CT,ON,COMP=IRLM,SUB=(DBM)
⋮
  (z/OS asks for a reply.)
⋮
R 15,WTR=CTWTR,END
TRACE CT,OFF,COMP=IRLM,SUB=(DBM)
⋮
  (Wait a while to make sure trace buffers are externalized.)
TRACE CT,WTRSTOP=CTWTR
```

Example 2 for TRACE CT command

Here is a sample procedure to start and stop traces in wrap-around mode. The traces captured in this procedure are saved in a limited number of buffers that are provided by IRLM. Each buffer is reused when the previous buffer is filled.

```
TRACE CT,ON,COMP=IRLM
⋮
  (z/OS asks for a reply.)
⋮
R 15,END
TRACE CT,OFF,COMP=IRLM
```

Related reference:

z/OS: System commands grouped according to system command authority

Part 9. Open Database Access commands

Use the Open Database Access (ODBA) commands to work with an ODBA application address space.

- Chapter 30, “CANCEL ODBA command,” on page 641
- Chapter 31, “FORCE ODBA command,” on page 643

© Copyright IBM Corp. 1974, 2017

Chapter 30. CANCEL ODBA command

Use the CANCEL and FORCE commands to terminate an ODBA application address space with an active connection to an IMS subsystem.

Subsections:

- "Syntax"
- "Usage notes"

Syntax

Usage notes

The CANCEL *jobname* command is used to terminate an ODBA application address space with an active connection to an IMS subsystem. In this command, *jobname* is the ODBA application address space.

If there are no active threads under this connection to IMS, enable the CANCEL command to be processed by z/OS. If there appear to be active threads (IDTTHDCT is greater than zero), the CANCEL command is rejected and message DFS08051 is issued.

Once the CANCEL command has been issued, no additional threads will be allowed to schedule from the target address space until the connection to IMS has been successfully terminated and re-initialized.

The CANCEL is rejected to protect the IMS control region from ABENDU0113 or other related abends. If it is invalid for threads to be active at this time, gather documentation to determine the reason for the active thread(s).

© Copyright IBM Corp. 1974, 2017

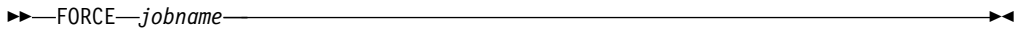
Chapter 31. FORCE ODBA command

The CANCEL and FORCE commands are used to terminate an ODBA application address space with an active connection to an IMS subsystem.

Subsections:

- "Syntax"
- "Usage notes"

Syntax



Usage notes

The FORCE *jobname* command is used to terminate an ODBA application address space with an active connection to an IMS subsystem if the CANCEL command was rejected. In this command, *jobname* is the ODBA application address space.

Recommendation: Attempt the CANCEL command prior to attempting the FORCE command.

If this is the first time that FORCE has been issued against the current iteration of the ODBA application address space, DFS1S100 will change the FORCE to a CANCEL and enable z/OS to process the CANCEL command.

If the CANCEL command was rejected and a prior FORCE command was issued against the current iteration of the ODBA application address space and was not successful, issue a second FORCE command to enable z/OS to process the second FORCE command.

Use the FORCE command with caution. By allowing z/OS to CANCEL the address space, it is possible that the IMS control region may terminate abnormally depending on the actual state of the active thread(s). Resolve the outstanding active threads prior to issuing the FORCE command.

© Copyright IBM Corp. 1974, 2017

Part 10. Repository Server commands

Use the Repository Server (RS) commands to display and update the RS parameters or to shut down RS address spaces.

- Chapter 32, “MODIFY reposervername commands,” on page 647
- Chapter 33, “P reposervername command,” on page 657

© Copyright IBM Corp. 1974, 2017

Chapter 32. MODIFY reposervername commands

Use the MODIFY (F) *reposervername* commands for the Repository Server (RS) address space to display or update RS parameters.

Subsections:

- "F reposervername,ADMIN"
- "F reposervername,AUDIT" on page 651
- "F reposervername,SECURITY" on page 652
- "F reposervername,SHUTDOWN" on page 653

Some MODIFY (F) and FRPBATCH commands are equivalent.

Table 43. Equivalent Modify (F) and FRPBATCH commands

MODIFY (F)	FRPBATCH	Note
--	ADD	
ADMIN DISPLAY	LIST	
ADMIN START	START	
ADMIN STOP	STOP	Stops the IMSRSC repository
--	RENAME	
--	DELETE	
ADMIN DSCHANGE	DSCHANGE	
--	UPDATE	
AUDIT	--	Changes the audit level
SECURITY	--	Refreshes in-storage profiles
SHUTDOWN	--	Stops the RS. Similar to the STOP command through the z/OS STOP (P) interface.

Related concepts:

Overview of the IMSRSC repository (System Definition)

IMSRSC repository administration (System Administration)

Related information:

Commands for FRPBATCH (System Programming APIs)

F reposervername,ADMIN

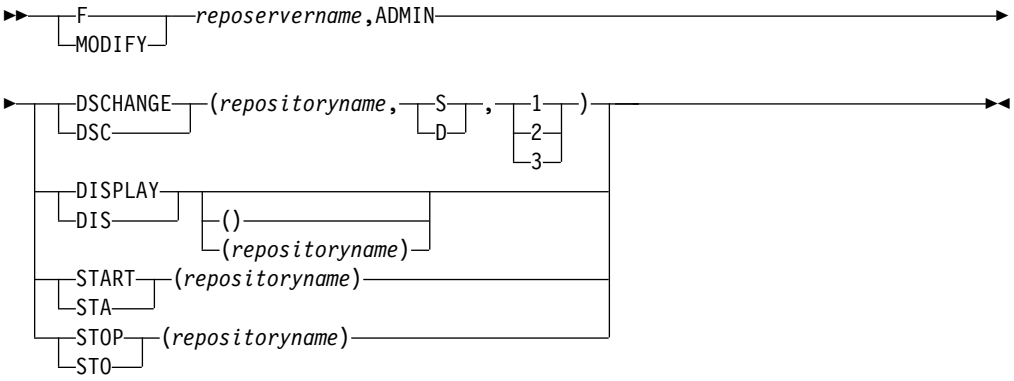
Use the F *reposervername*,ADMIN command to perform administrative tasks on the IMSRSC repository.

Subsections:

- "Syntax" on page 648
- "Keywords" on page 648
- "Equivalent FRPBATCH commands" on page 649
- "Examples" on page 649

© Copyright IBM Corp. 1974, 2017

Syntax



Keywords

The following keywords can be specified for this command:

reposervername
Specifies the name of the Repository Server (RS) on which you want to perform administrative tasks.

DSCHANGE
Changes the status of a pair of COPY1, COPY2, or spare repository data sets to DISCARD, or changes the status of a discarded repository data set to SPARE.

repositoryname
Specifies the name of the repository whose data sets you want to change. You cannot specify CATALOG because it is reserved.

S | D
Specifies the DSCHANGE action to be applied to the repository data sets specified in the repository data set parameter.

S The DSCHANGE action of SPARE is to be applied to the repository data sets specified.

D The DSCHANGE action of DISCARD is to be applied to the repository data sets specified.

1 | 2 | 3
Specifies a number in the range of 1-3 to indicate the repository data set pair to which the requested DSCHANGE action is to apply.

1 Applies the status to the COPY1 repository data sets.

2 Applies the status to the COPY2 repository data sets.

3 Applies the status to the SPARE repository data sets.

DISPLAY
Displays a list of repositories defined to the RS catalog repository. This command is similar to the batch LIST STATUS command. If *repositoryname* is not specified, only the repository names are returned. If *repositoryname* is specified, details of the repository, such as the data set names, are returned.

repositoryname
Specifies the name of the repository whose information is to be displayed. You cannot specify CATALOG because it is reserved.

START
Starts the repository.

repositoryname
Specifies the name of the repository to be started. You cannot specify CATALOG because it is reserved.

STOP
Stops the repository.

repositoryname
Specifies the name of the repository to be stopped. You cannot specify CATALOG because it is reserved.

Equivalent FRPBATCH commands

The following table shows the FRPBATCH commands that perform similar functions as the F *reposervername*,ADMIN command.

Table 44. FRPBATCH equivalents for the F reposervername,ADMIN command

F reposervername,ADMIN command	Similar FRPBATCH commands
F *reposervername*,ADMIN DISPLAY	LIST
F *reposervername*,ADMIN DSCHANGE	DSCHANGE
F *reposervername*,ADMIN START	START
F *reposervername*,ADMIN STOP	STOP

Examples

The following are examples of the F *reposervername*,ADMIN command:

Example 1 for F reposervername,ADMIN command

Command input:

```
F FRP1,ADMIN START(IMS_REPOS)
```

Command output:

```
FRP2014I - Repository start request initiated: IMS_REPOS
                   REPO1RP
```

Explanation: The request to start the repository named IMS_REPOS is initiated. When the repository is started, a FRP2012I message is issued. The command is processed by the master Repository Server, which has RSNAME=REPO1 specified in the FRPCFG member. The RSID of REPO1RP is appended to the command output message issued by the master RS.

Example 2 for F reposervername,ADMIN command

Command input:

```
F FRP1,ADMIN STOP(IMS_REPOS)
```

Command output:

```
FRP2020I - Repository stop request initiated: IMS_REPOS
                  REPO1RP
```

Explanation: The request to stop the repository named IMS_REPOS is initiated. When the repository is stopped, a FRP2012I message is issued. The command is processed by the master Repository Server, which has RSNAME=REPO1 specified in the FRPCFG member. The RSID of REPO1RP is appended to the command output message issued by the master RS.

Example 3 for F reposervername,ADMIN command

Command input:

```
F FRP1,ADMIN DISPLAY(IMS_REPOS)
```

Command output:

```
BPE0032I ADMIN COMMAND COMPLETED REPO1RP
FRP2100I - ADMIN DISPLAY repository IMS_REPOS 597
         - Last updated date/time  : 2011/05/06 20:26:43  USRT004
         - Status  . . . . . . . . : OPEN
         - Auto-open . . . . . . . : YES
         - Security Class  . . . . : NOT DEFINED REPO1RP
FRP2101I - ADMIN DISPLAY repository RDS1: 598
         - Index (RID) . . : IMSTESTS.REPO.IMSPRI.RID
         - Member (RMD)  . : IMSTESTS.REPO.IMSPRI.RMD
         - Status  . . . . : COPY1    REPO1RP
FRP2101I - ADMIN DISPLAY repository RDS2: 599
         - Index (RID) . . : IMSTESTS.REPO.IMSSEC.RID
         - Member (RMD)  . : IMSTESTS.REPO.IMSSEC.RMD
         - Status  . . . . : COPY2    REPO1RP
FRP2101I - ADMIN DISPLAY repository RDS3: 600
         - Index (RID) . . : IMSTESTS.REPO.IMSSPR.RID
         - Member (RMD)  . : IMSTESTS.REPO.IMSSPR.RMD
         - Status  . . . . : DISCARD  REPO1RP
```

Explanation: The ADMIN command to display information about repository IMS_REPOS is processed. The repository status and the information about the data sets for COPY1, COPY2, and SPARE are returned. The command is processed by the master Repository Server, which has RSNAME=REPO1 specified in the FRPCFG member. The RSID of REPO1RP is appended to the command output message issued by the master RS.

Example 4 for F reposervername,ADMIN command

Command input:

```
F FRP1,ADMIN DISPLAY
```

Command output:

```
BPE0032I ADMIN DISPLAY COMMAND COMPLETED REPO1RP
FRP2102I - IMS_REPOS                                    OPEN
2011/05/06 USRT004  COPY1    COPY2    DISCARD  REPO1RP
```

Explanation: The ADMIN command to display information about the repositories that are managed by the Repository Server is processed. The output does not contain the data set information. The command is processed by the master Repository Server, which has RSNAME=REPO1 specified in the FRPCFG member. The RSID of REPO1RP is appended to the command output message issued by the master RS.

Related concepts:

- Overview of the IMSRSC repository (System Definition)
- IMS repository data set states (System Definition)

Related tasks:

Stopping the Repository Server (Operations and Automation)

Allocating the IMSRSC repository data sets (System Definition)

Related reference:

"F reposervername,SHUTDOWN" on page 653

LIST command for FRPBATCH (System Programming APIs)

START command for FRPBATCH (System Programming APIs)

STOP command for FRPBATCH (System Programming APIs)

DSCHANGE command for FRPBATCH (System Programming APIs)

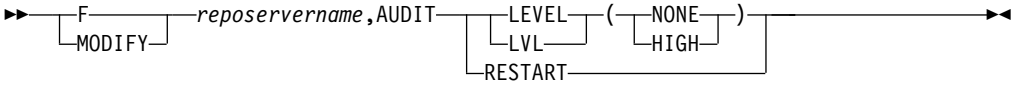
F reposervername,AUDIT

Use the F *reposervername*,AUDIT command to dynamically change the auditing level from the value that is set by the AUDIT_LEVEL parameter in the FRPCFG member of the IMS PROCLIB data set.

Subsections:

- "Syntax"
- "Keywords"
- "Examples"

Syntax



Keywords

The following keywords can be specified for this command:

LEVEL
Determines whether audit records are written to the audit log stream.

NONE
Audit records are not written.

HIGH
Audit records are written.

RESTART
If AUDIT_FAIL=CONTINUE is specified in the FRPCFG member of the IMS PROCLIB data set and an error occurs while connecting to the audit log stream, the audit logging is suspended. You can use RESTART to resume audit logging.

Examples

The following are examples of the F *reposervername*,AUDIT command:

Example 1 for F reposervername,AUDIT command

Command input:

```
F FRP1,AUDIT LEVEL(NONE)
```

Command output:

```
BPE0032I AUDIT COMMAND COMPLETED REPO1RP
FRP2103I - Audit level changed from HIGH to NONE REPO1RP
```

Explanation: The AUDIT command is processed, and the audit level is changed from HIGH to NONE. The command is processed by the master Repository Server, which has RSNAME=REPO1 specified in the FRPCFG member. The RSID of REPO1RP is appended to the command output message issued by the master RS.

Example 2 for F reposervername,AUDIT command

Command input:

```
F FRP1,AUDIT LEVEL(HIGH)
```

Command output:

```
BPE0032I AUDIT COMMAND COMPLETED REPO1RP
FRP2103I - Audit level changed from NONE to HIGH REPO1RP
```

Explanation: The AUDIT command is processed and the audit level is changed from NONE to HIGH. The command is processed by the master Repository Server, which has RSNAME=REPO1 specified in the FRPCFG member. The RSID of REPO1RP is appended to the command output message issued by the master RS.

Example 3 for F reposervername,AUDIT command

Command input:

```
F FRP1,AUDIT RESTART
```

Command output:

```
BPE0032I AUDIT RESTART COMMAND COMPLETED REPO1RP
```

Explanation: The AUDIT command is processed to restart the audit processing. The command is processed by the master Repository Server, which has RSNAME=REPO1 specified in the FRPCFG member. The RSID of REPO1RP is appended to the command output message issued by the master RS.

Related concepts:

Overview of the IMSRSC repository (System Definition)

Managing Repository Server audit log records (Diagnosis)

Related reference:

FRPCFG member of the IMS PROCLIB data set (System Definition)

F reposervername,SECURITY

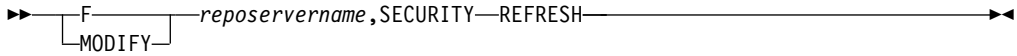
Use the F *reposervername*,SECURITY command to dynamically refresh the in-storage security tables, to pick up changes to the profiles as defined in RACF or an equivalent security product. This command does not reload Repository Server configuration file parameters or reread repository definitions.

Subsections:

- "Syntax" on page 653

- "Examples"

Syntax



Examples

The following are examples of the F *reposervername*,SECURITY command:

Example 1 for F reposervername,SECURITY command

Command input:

```
F FRP1,SECURITY REFRESH
```

Command output:

```
FRP2106E - Security request rejected, CLASS not defined REPO1RP
BPE0032I SECURITY REFRESH COMMAND COMPLETED REPO1RP
```

Explanation: The SECURITY command is processed to refresh the security profiles. The command is processed by the master Repository Server, which has RSNAME=REPO1 specified in the FRPCFG member. The RSID of REPO1RP is appended to the command output message issued by the master RS.

Related concepts:

Updating IMSRSC repository specifications in the RS catalog repository (System Administration)

Overview of the IMSRSC repository (System Definition)

Related reference:

Example RACF definition profile for the Repository Server (System Administration)

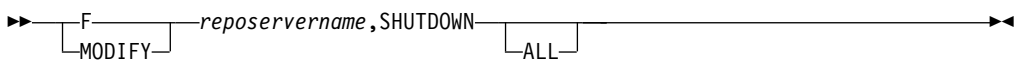
F reposervername,SHUTDOWN

Use the F *reposervername*,SHUTDOWN command to shut down the specified Repository Server (RS) or all the RS address spaces.

Subsections:

- "Syntax"
- "Keywords" on page 654
- "Usage notes" on page 654
- "Examples" on page 654

Syntax



Keywords

The following keywords can be specified for this command:

ALL
Shuts down all RS address spaces that use the same z/OS cross-system coupling facility group as the specified RS, including any active subordinate servers.

Usage notes

If ALL is not specified, only the RS on the z/OS image where the command is issued is shut down. If the master server is shut down, and SHUTDOWN ALL is not issued, all subordinate servers start the subordinate process and one subordinate server becomes the new master.

Examples

The following are examples of the F *reposervername*,SHUTDOWN command:

Example 1 for F reposervername,SHUTDOWN command

Command input:

```
F FRP1,SHUTDOWN
```

Command output:

```
FRP2005I - Shutdown command received, server terminating REPO1RP
```

Explanation: The SHUTDOWN command to shut down the Repository Server address space is received, and the server will be shutting down. The command is processed by the master Repository Server, which has RSNAME=REPO1 specified in the FRPCFG member. The RSID of REPO1RP is appended to the command output message issued by the master RS.

Example 2 for F reposervername,SHUTDOWN command

Command input:

```
F FRP1,SHUTDOWN ALL
```

Command output:

```
FRP2007I - Shutdown command received, notifying all repository servers
to shut down REPO1RP
```

Explanation: The SHUTDOWN command to shut down the Repository Server address space and to notify all other Repository Servers is received. The command is processed by the master Repository Server, which has RSNAME=REPO1 specified in the FRPCFG member. The RSID of REPO1RP is appended to the command output message issued by the master RS.

Related concepts:

Overview of the IMSRSC repository (System Definition)

Related tasks:

Stopping the Repository Server (Operations and Automation)

Related reference:

"F reposervername,ADMIN" on page 647

Chapter 33, “P reposervername command,” on page 657

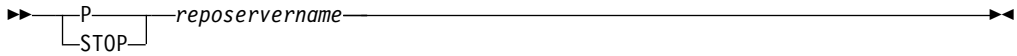
Chapter 33. P reposervername command

Use the STOP command through the z/OS STOP (P) interface to stop and shut down the specified Repository Server. If the master server is shut down, all subordinate servers start the subordinate process and one subordinate server becomes the new master.

Subsections:

- "Syntax"

Syntax



Related concepts:

Overview of the IMSRSC repository (System Definition)

Related reference:

"F reposervername,SHUTDOWN" on page 653

© Copyright IBM Corp. 1974, 2017

Part 11. Appendixes

© Copyright IBM Corp. 1974, 2017

Notices

This information was developed for products and services offered in the US. 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 right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not 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
US

For license inquiries regarding double-byte character set (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

© Copyright IBM Corp. 1974, 2017

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.

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

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

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

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

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

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

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.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

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

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as follows:
© (your company name) (year).
Portions of this code are derived from IBM Corp. Sample Programs.
© Copyright IBM Corp. _enter the year or years_.

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 http://www.ibm.com/legal/copytrade.shtml.

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

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.

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

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.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

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

For more information about the use of various technologies, including cookies, for these purposes, See IBM’s Privacy Policy at http://www.ibm.com/privacy and IBM’s Online Privacy Statement at http://www.ibm.com/privacy/details the section entitled “Cookies, Web Beacons and Other Technologies” and the “IBM Software Products and Software-as-a-Service Privacy Statement” at http://www.ibm.com/software/info/product-privacy.

Bibliography

This bibliography lists all of the publications in the IMS 15 library.

Title	Acronym	Order number
IMS Version 15 Application Programming	APG	SC27-6778
IMS Version 15 Application Programming APIs	APR	SC27-6779
IMS Version 15 Commands, Volume 1: IMS Commands A-M	CR1	SC27-6780
IMS Version 15 Commands, Volume 2: IMS Commands N-V	CR2	SC27-6781
IMS Version 15 Commands, Volume 3: IMS Component and z/OS Commands	CR3	SC27-6782
IMS Version 15 Communications and Connections	CCG	SC27-6783
IMS Version 15 Database Administration	DAG	SC27-6784
IMS Version 15 Database Utilities	DUR	SC27-6785
IMS Version 15 Diagnosis	DGR	GC27-6786
IMS Version 15 Exit Routines	ERR	SC27-6787SC19-4208
IMS Version 15 Installation	INS	SC27-6788
IMS Version 15 Licensed Program Specifications	LPS	GC27-6799
IMS Version 15 Messages and Codes, Volume 1: DFS Messages	MC1	GC27-6789
IMS Version 15 Messages and Codes, Volume 2: Non-DFS Messages	MC2	GC27-6790
IMS Version 15 Messages and Codes, Volume 3: IMS Abend Codes	MC3	GC27-6791
IMS Version 15 Messages and Codes, Volume 4: IMS Component Codes	MC4	GC27-6792
IMS Version 15 Operations and Automation	OAG	SC27-6793
IMS Version 15 Release Planning	RPG	GC27-6794
IMS Version 15 System Administration	SAG	SC27-6795
IMS Version 15 System Definition	SDG	GC27-6796
IMS Version 15 System Programming APIs	SPR	SC27-6797
IMS Version 15 System Utilities	SUR	SC27-6798

© Copyright IBM Corp. 1974, 2017

Index

Special characters

/NRESTART
  INIT.DB 290
/RMxxxxxx command
  authorization of 50
%ALLDSSN keyword 246
%ALLSEL keyword 246
%ALLTIME keyword 246
%ALLUSID keyword 246
%CADSN keyword 244
%CAFSEQ keyword 244
%CALGTM keyword 244
%CAODSN keyword 231
%CASEL keyword 244
%CATIME keyword 244
%CAUNIT keyword 244
%CAVCNT keyword 244
%CAVOLS keyword 244
%COPIES keyword 272
%DALTIME keyword 246
%DBADSAV keyword 247
%DBADSN keyword 247
%DBDDN keyword 246, 247
%DBDSN keyword 247
%DBDSNRV keyword 247
%DBDSSEL keyword 247
%DBNAME keyword 246, 247
%DBTYPE keyword 246, 247
%DBUSID keyword 247
%DDNAME keyword 228
%DELETE keyword 230
%ENDDEL keyword 230
%ENDSEL keyword 227
%IC2DSN keyword 242
%IC2FSEQ keyword 242
%IC2SEL keyword 242
%IC2UNIT keyword 242
%IC2VCNT keyword 242
%IC2VOLS keyword 242
%ICCAT keyword 242
%ICDSN keyword 241
%ICDSN2 keyword 272
%ICDSN3 keyword 272
%ICDSN4 keyword 272
%ICFSEQ keyword 241
%ICSEL keyword 241
%ICSTOP keyword 242
%ICTIME keyword 242
%ICTYPE keyword 241
%ICUNIT keyword 242
%ICUNIT2 keyword 272
%ICUNIT3 keyword 272
%ICUNIT4 keyword 272
%ICUSID keyword 242
%ICVCNT keyword 242
%ICVCNT2 keyword 272
%ICVCNT3 keyword 272
%ICVCNT4 keyword 272
%ICVOLS keyword 242
%ICVOLS2 keyword 272
%ICVOLS3 keyword 272
%ICVOLS4 keyword 272
%LOGDSN keyword 239
%LOGETIM keyword 240
%LOGFRID keyword 240
%LOGFSEQ keyword 239
%LOGLRID keyword 240
%LOGMERG keyword 240
%LOGONL keyword 240
%LOGRMT keyword 240
%LOGSEL keyword 240
%LOGSTIM keyword 239
%LOGUNIT keyword 239
%LOGVOLS keyword 239
%OLDCTIM keyword 234
%OLDFRID keyword 234
%OLDLRID keyword 234
%OLDOTIM keyword 234
%OLDSDDN keyword 234
%OLDSDSN keyword 234
%OLDSSEL keyword 234
%OLDSTYP keyword 234
%PLGTIME keyword 246
%SELECT keyword 235
  description 227
  specifying the record types 228
  specifying the syntax 233, 247
%SET MEMBER keyword 231
%SET TIMEFMT keyword 232
%SLDCHKT keyword 237
%SLDETIM keyword 236
%SLDFRID keyword 236
%SLDFSEQ keyword 236
%SLDLRID keyword 236
%SLDRMT keyword 236
%SLDSDDN keyword 236
%SLDSSEL keyword 236
%SLDSTIM keyword 236
%SLDUNIT keyword 236
%SLDVOLS keyword 236
%SSID keyword 226
%TIME keyword
  as a symbolic keyword 226
  description 250

A

ABNORMAL keyword commands
  CHANGE.SUBSYS 144
  NOTIFY.SUBSYS 385
accessibility
  features ix
  keyboard shortcuts ix
ADD keyword
  CHANGE.CAGRP command 69
ADDDB keyword
  CHANGE.DBDSGRP command 94
ADDEQE keyword
  CHANGE.DBDS command 87
adding information to the RECON data set
  allocation or deallocation 345
adding information to the RECON data set *(continued)*
  backout records 347
  Database Change Accumulation utility 348
  database data set or area recovery 366
  image copy 352
  nonstandard image copy data sets 386
  primary online log data set 356
  primary recovery log data set 359
  primary system log data set 363
  secondary online log data set 374
  secondary recovery log data set 377
  secondary system log data set 381
  subsystem 384
ADDN keyword commands
  CHANGE.ADS 63
  DELETE.ADS 161
  INIT.ADS 284
  NOTIFY.RECOV 368
ADDNNEW keyword
  CHANGE.ADS command 63
ADSN keyword commands
  CHANGE.ADS 63
  INIT.ADS 284
ALL keyword
  commands
    CHANGE.DB 74
    CHANGE.SUBSYS 143
    GENJCL.ARCHIVE 177
    LIST.BKOUT 321
    LIST.CAGRP 322
    LIST.DB 324
    LIST.DBDSGRP 330
    LIST.SUBSYS 342
  description 74
ALLD parameter
  F irlmproc,STATUS command 623
ALLI parameter
  F irlmproc,STATUS command 623
ALLTIME keyword
  NOTIFY.ALLOC command 345
ALTER keyword
  NOTIFY.REORG 373
ARCHIVED keyword
  CHANGE.PRILOG command (for OLDS) 107
ARCHJCL skeletal JCL execution member 259
AREA parameter
  commands
    CHANGE.ADS 63
    CHANGE.DBDS 86
    CHANGE.IC 97
    CHANGE.UIC 146
    DELETE.ADS 161
    DELETE.ALLOC 162
    DELETE.DBDS 166
    DELETE.IC 168

© Copyright IBM Corp. 1974, 2017

AREA parameter *(continued)*
commands *(continued)*
DELETE.RECOV 173
DELETE.UIC 176
GENJCL.IC 192
GENJCL.OIC 208
GENJCL.RECOV 213
INIT.ADS 284
INIT.DBDS 295
INIT.IC 303
LIST.DBDS 327
LIST.HISTORY 332
NOTIFY.ALLOC 346
NOTIFY.IC 352
NOTIFY.RECOV 368
NOTIFY.UIC 387
defining a data set 283
deleting information from RECON 161
AREANEW keyword
CHANGE.DBDS command 88
ARNEEDED keyword
CHANGE.PRILOG command (for OLDS) 107
ARSCHED keyword
CHANGE.PRILOG command (for OLDS) 107
ARSTART keyword
CHANGE.PRILOG command (for OLDS) 107
authorization commands
CHANGE.DB 72, 76
CHANGE.DBDS 84, 88
Automatic RECON Loss Notification
DSPSCIX0 124
AVAIL keyword commands
CHANGE.ADS 63
CHANGE.PRILOG (for OLDS) 108
CHANGE.SECLOG (for OLDS) 132, 133
INIT.ADS 284

B

BACKIRLM keyword commands
CHANGE.SUBSYS 144
BACKOUT keyword
CHANGE.DB command 77
backup
creating a copy of RECON 61
BACKUP.RECON command
description 61
example 62, 391
keywords
BOTH 61
RECON1 61
RECON2 61
Base Primitive Environment (BPE) commands 1
batch backout utility
adding backout records to the RECON data set 347
changing backout records in the RECON data set 64
deleting from RECON 163
batch environment
command
LIST.SUBSYS 342
NOTIFY.SUBSYS 385
NOTIFY.SUBSYS command 384
BKO (backout) keyword commands
CHANGE.BKOUT 66
NOTIFY.BKOUT 347
BOTH keyword
BACKUP.RECON command 61
BPE commands 1
DISPLAY USEREXIT 23
invocation 3
REFRESH USEREXIT 33
specify IMS component command parameters 3
syntax 3
Syntax, verb only format 3
Syntax, verb-resource type 3
wildcard character support 3
BPE DISPLAY VERSION command
examples 8
keywords 7
output 7
syntax 7
BPE user-supplied exit routines
refresh 33

C

CADSN keyword commands
CHANGE.CA 67
INIT.CA 285
NOTIFY.CA 348
CAJCL keyword
commands
CHANGE.CAGRP 69
INIT.CAGRP 287
CAJCL parameter
skeletal JCL execution member 263
CANCEL ODBA command
description 641
syntax diagram 641
CATALOG keyword
CHANGE.DBDS 86
CHANGE.PART 100
CHANGE.RECON 119
INIT.DB 290
INIT.DBDS 296, 371
INIT.PART 306
INIT.RECON 312
CATDS keyword commands
CHANGE.RECON 118
INIT.RECON 313
category of records
listing records 335
CATIME keyword
GENJCL.CA command 184
CFSTR1 keyword
CHANGE.DBDS command 87
CFSTR1 keyword INIT.DBDS command 298
CFSTR2 keyword
CHANGE.DBDS command 87
CFSTR2 keyword INIT.DBDS command 298
change accumulation
data set
defining 285
selecting 243
group
changing information in the RECON 69
defining 287
deleting information from RECON 164
listing 322
CHANGE.ADS command 63
CHANGE.BKOUT command 64
CHANGE.CA command 66
CHANGE.CAGRP command 69
CHANGE.DB command 72
CHANGE.DBDS command 84
CHANGE.DBDSGRP command 93
CHANGE.IC command 96
CHANGE.PART command 99
CHANGE.PRILOG command
for OLDS 107
for RLDS 109
for SLDS 113
CHANGE.RECON command 118
CHANGE.SECLOG command
for OLDS 132
for RLDS 134
for SLDS 139
CHANGE.SUBSYS command 143
CHANGE.UIC command 145
changing information
area data set 63
backout records 64
CA group record 69
database 72
database change accumulation utility 66
DBDS 84
DBDSGRP 93
IC data set 96
nonstandard image copy data set 145
primary online log data set 107
primary RLDS 109
primary SLDS 113
RECON header record 118
secondary online log data set 132
secondary RLDS 134
secondary SLDS 139
secondary subsystem entry 143
CHECK17 keyword commands
CHANGE.RECON 124
CHECK17 keyword commands
CHANGE.RECON 124
INIT.RECON 314
CHECK44 keyword commands
CHANGE.RECON 124
CHECK44 keyword commands
CHANGE.RECON 124
INIT.RECON 314
checking for inconsistencies 389
CHKINT keyword GENJCL.OIC command 207
CHKPTCT keyword commands
CHANGE.PRILOG (for RLDS) 110
CHANGE.PRILOG (for SLDS) 114
CHANGE.SECLOG (for RLDS) 135

CHKPTCT keyword commands *(continued)*
CHANGE.SECLOG (for SLDS) 140
NOTIFY.PRILOG (for RLDS) 360
NOTIFY.PRILOG (for SLDS) 364
NOTIFY.SECLOG (for RLDS) 378
NOTIFY.SECLOG (for SLDS) 382
CIC (concurrent image copy) 192
CIC keyword
commands
GENJCL.IC 192
NOTIFY.IC 352
CIC keyword, commands
GENJCL.IC 192
NOTIFY.IC 352
CLEANUP.RECON command
description 147
example 150
CLOSEHWS command 445
equivalent IMS Connect type-2 and z/OS commands 446
CMDAUTH
CHANGE.RECON 121
CMDAUTH keyword
INIT.RECON 313
cold start
commands
CHANGE.BKOUT 64
DELETE.BKOUT 163
LIST.BKOUT 321
command
CANCEL ODBA 641
CHANGE.PART 99
F fdbrproc 429
F fdbrproc,DUMP command 429
F fdbrproc,RECOVER command 429
F fdbrproc,STATUS command 429
F fdbrproc,STOP command 431
F fdbrproc,TERM command 431
F irlmproc,ABEND command 617
F irlmproc,DIAG,DELAY command 618
F irlmproc,PURGE,imsname command 619
F irlmproc,SET command 620
F irlmproc,STATUS command 623
F jobname,DUMP command 609
F jobname,DUMPxxxx command 609
F jobname,FORCExxxx 610
F jobname,RECONNECT command 611
F jobname,RECONNSTR command 612
F jobname,STOP command 613
F reposervername 647
F reposervername,ADMIN command 647
F reposervername,AUDIT command 651
F reposervername,SECURITY command 652
F reposervername,SHUTDOWN command 653
INIT.PART 305
online
/RMCHANGE command 50
/RMDELETE command 50
command *(continued)*
online *(continued)*
/RMGENJCL command 50
/RMINIT command 50
/RMLIST command 50
/RMNOTIFY command 50
P cqsjob 45
P CSL 41
P irlmproc 633
P reposervername command 657
S irlmproc command 629
TRACE CT 635
command syntax
comment 51
continuation characters 51
definition 51
description for DBRC utility 50
parameters 51
separators 51
commands
equivalent IMS Connect WTOR, z/OS, and type-2 commands 437
IMS Connect
REFRESH CONVERTER 452
REFRESH RACFUID 453
SETUIDC 459
STARTMSC 464
STARTRMT 467
STOPLINK 474
STOPMSC 476
STOPRMT 480
STOPSCLN 481
VIEWMSC 517
VIEWRMT 526
prerequisite knowledge vii
commands, IMS Connect
CLOSEHWS 445
OPENDS 447
OPENIP 448
OPENPORT 449
RECORDER 450
REFRESH CONVERTER 452
REFRESH RACFUID 453
SETOAUTO 454
SETPWMC 455
SETRACF 457
SETRRS 458
SETUIDC 459
STARTDS 460
STARTIA 461
STARTIP 462
STARTMSC 464
STARTOD 465
STARTPT 466
STARTRMT 467
STOPCLNT 469
STOPDS 470
STOPIA 471
STOPIP 473
STOPLINK 474
STOPMSC 476
STOPOD 477
STOPPORT 478
STOPRMT 480
STOPSCLN 481
VIEWDS 483
VIEWHWS 488
commands, IMS Connect *(continued)*
VIEWIA 512
VIEWIP 515
VIEWMSC 517
VIEWPORT 521
VIEWRMT 526
VIEWUOR 530
commands, MVS
DELETE CLIENT 534
QUERY ALIAS 539
QUERY DATASTORE 540
QUERY IMSPLEX 545
QUERY MEMBER 547
QUERY PORT 576
QUERY UOR 585
SHUTDOWN MEMBER 587
UPDATE ALIAS 588
UPDATE CONVERTER 589
UPDATE DATASTORE 591
UPDATE IMSPLEX 592
UPDATE MEMBER 594
UPDATE ODBM 599
UPDATE PORT 601
UPDATE RACFUID 602
wildcards 534
commands, z/OS 533
DELETE LINK 535
DELETE SENDCLNT 537
invocation 533
QUERY MSC 571
QUERY RMTIMSCON 581
syntax 533
UPDATE MSC 597
UPDATE RMTIMSCON 603
Common Queue Server (CQS) command 43
Common Service Layer (CSL) commands 37
COMP keyword commands
CHANGE.CA 68
NOTIFY.CA 348
complex expressions 231
concurrent image copy (CIC)
CIC keyword 192
continuation characters 51
control group, skeletal JCL 227
control keywords, skeletal JCL 227, 247
COPIES keyword commands
GENJCL.IC 191
GENJCL.OIC 208
coupling facility structures (CFSTR1 | 2) 298
CQS command 43
CSL commands 37
CSL SHUTDOWN
description 39
usage notes 39
CSLPLEX SHUTDOWN
description 39
syntax 39
usage notes 39
CSLSHUTDOWN
syntax 39
CURRENT keyword commands
NOTIFY.RECOV 368
NOTIFY.REORG 371
NOTIFY.UIC 387

D

DASDUNIT keyword commands
  CHANGE.RECON 122
  INIT.RECON 314
data management block (DMB) numbers 389
database
  changing information 72
  data set groups, changing information 93
  data set, changing information 84
  defining 289
  deleting information from RECON 165
  listing 324
Database Change Accumulation utility (DFSUCUM0)
  commands
    BACKUP.RECON 61
    CHANGE.ADS 63
    CHANGE.BKOUT 64
    CHANGE.CA 66
    CHANGE.CAGRP 69
    CHANGE.DBDSGRP 93
    CHANGE.IC 96
    CHANGE.PRILOG (for OLDS) 107
    CHANGE.PRILOG (for SLDS) 113
    CHANGE.RECON 118
    CHANGE.SECLOG (for OLDS) 132
    CHANGE.SECLOG (for RLDS) 134
    CHANGE.SECLOG (for SLDS) 139
    CHANGE.SUBSYS 143
    CHANGE.UIC 145
    DELETE.ADS 161
    DELETE.ALLOC 162
    DELETE.BKOUT 163
    DELETE.CA 163
    DELETE.CAGRP 164
    DELETE.DB 165
    DELETE.DBDS 166
    DELETE.DBDSGRP 167
    DELETE.IC 167
    DELETE.LOG (for OLDS) 168
    DELETE.LOG (for RLDS) 169
    DELETE.LOG (for SLDS) 169
    DELETE.RECOV 173
    DELETE.REORG 174
    DELETE.SUBSYS 175
    DELETE.UIC 176
    GENJCL.ARCHIVE 177
    GENJCL.CA 181
    GENJCL.CLOSE 186
    GENJCL.IC 189
    GENJCL.OIC 206
    GENJCL.RECOV 211
    GENJCL.USER 218
    INIT.ADS 283
    INIT.CA 285
    INIT.CAGRP 287
    INIT.DB 289
    INIT.DBDS 294
    INIT.DBDSGRP 301
Database Change Accumulation utility (DFSUCUM0) *(continued)*
  commands *(continued)*
    INIT.IC 303
    INIT.RECON 311
    LIST.BKOUT 321
    LIST.CAGRP 322
    LIST.DB 324
    LIST.DBDS 326
    LIST.DBDSGRP 329
    LIST.HISTORY 331
    LIST.LOG (for a category of records) 335
    LIST.LOG (for a PRILOG family) 334
    LIST.RECON 338
    NOTIFY.ALLOC 345
    NOTIFY.BKOUT 347
    NOTIFY.CA 348
    NOTIFY.IC 352
    NOTIFY.PRILOG (for OLDS) 356
    NOTIFY.PRILOG (for RLDS) 359
    NOTIFY.PRILOG (for SLDS) 363
    NOTIFY.RECOV 366
    NOTIFY.REORG 370
    NOTIFY.SECLOG (for OLDS) 374
    NOTIFY.SECLOG (for RLDS) 377
    NOTIFY.SECLOG (for SLDS) 381
    NOTIFY.UIC 386
    REPAIR.RECON 389
Database Change Accumulation utility JCL
  adding information to the RECON data set 348
  changing information about a run 66
  Database Change Accumulation utility JCL
    skeletal JCL 263
  deleting information from RECON 163
  generating a job 181
Database Image Copy utility (DFSUDMP0)
  adding information to the RECON data set 352
  generating a job 189
  skeletal JCL 268
database recovery
  defining groups 301
database recovery control
  changing information 96
  changing partition information 99
  data set, checking for inconsistent DMB numbers 389
  data set, creating a backup copy 61
  data set, deleting old information 147
  data set, rebuilding the DMB Table record 389
  data set, repairing inconsistent DMB numbers 389
Database Recovery Control (DBRC) commands 47
Database Recovery Control utility (DSPURX00)
  generating a job 211, 218
  GENJCL.RECOV command 211
databases
  RECON data set
    reorganization record 370
  reorganizing
    RECON data set, adding a reorganization record 370
DBCTL (database control)
  starting subsystem
    START irlmproc 629
DBD keyword commands
  CHANGE.ADS 63
  CHANGE.BKOUT 66
  CHANGE.DB 74
  CHANGE.DBDS 86
  CHANGE.IC 96
  CHANGE.UIC 145
  DELETE.ADS 161
  DELETE.ALLOC 162
  DELETE.DB 165
  DELETE.DBDS 166
  DELETE.IC 167
  DELETE.PART 172
  DELETE.RECOV 173
  DELETE.REORG 174
  DELETE.UIC 176
  GENJCL.IC 191
  GENJCL.OIC 207
  GENJCL.RECOV 212
  GENJCL.USER 219
  INIT.ADS 284
  INIT.DB 289
  INIT.DBDS 295
  INIT.IC 303
  LIST.DB 324
  LIST.DBDS 327
  LIST.HISTORY 331
  NOTIFY.ALLOC 345
  NOTIFY.BKOUT 347
  NOTIFY.IC 352
  NOTIFY.RECOV 367
  NOTIFY.UIC 386
DBDS (database data set)
  commands 213
    GENJCL.RECOV 213
    LIST.DB 324
    NOTIFY.RECOV 368
  defining 294
  listing 326
  qualifier 229
  RECON data set, deleting information from 166, 167
  selecting DBDS records 244, 246
DBDS (database data set) group
  defining 301
  deleting information from RECON 167
  LIST command 329
DBQUI keyword
  CHANGE.DB command 76
  CHANGE.DBDS command 88
DBQUIH keyword
  CHANGE.DB command 76
  CHANGE.DBDS command 88
DBQUIH keyword commands
  GENJCL.IC 192

DBRC
adding records
database reorganizations 370
reorganization record 370
DBRC (Database Recovery Control)
commands
introduction 49
online commands 50
time stamp 52
DBRC commands 47
NOTIFY.REORG 370
DDN keyword commands
CHANGE.DBDS 86
CHANGE.IC 97
CHANGE.UIC 146
DELETE.ALLOC 162
DELETE.DBDS 166
DELETE.IC 168
DELETE.RECOV 173
DELETE.REORG 174
DELETE.UIC 176
GENJCL.IC 192
GENJCL.OIC 208
GENJCL.RECOV 213
GENJCL.USER 219
INIT.DBDS 295
INIT.IC 303
LIST.DBDS 327
LIST.HISTORY 332
NOTIFY.ALLOC 346
NOTIFY.IC 352
NOTIFY.RECOV 368
NOTIFY.REORG 371
NOTIFY.UIC 387
DDNNEW keyword
CHANGE.DBDS command 88
DEADLOK= parameter
START irlmproc command 629
DEALTIME keyword command
NOTIFY.ALLOC 346
default members 224, 248
DEFAULTS keyword commands
GENJCL.ARCHIVE 178
GENJCL.CA 182
GENJCL.CLOSE 186
GENJCL.IC 198
GENJCL.OIC 208
GENJCL.RECOV 213
GENJCL.USER 220
defining
DB groups 301
DBDS groups 301
recovery groups 301
DEFLTJCL keyword
CHANGE.CAGRP 69
CHANGE.DBDS 88
INIT.CAGRP 287
DELEQE keyword
CHANGE.DBDS command 87
DELETE CLIENT command
equivalent IMS Connect type-2 and WTOR commands 535
example 534
syntax 534
usage 534
DELETE command 534
CLIENT 534
delete group 230
DELETE keyword commands
CHANGE.BKOUT 65
CHANGE.CAGRP 69
DELETE LINK command 535
equivalent IMS Connect type-2 and WTOR commands 537
DELETE SENDCLNT command 537
equivalent IMS Connect type-2 and WTOR commands 538
DELETE.ADS command 161
DELETE.ALLOC command 162
DELETE.BKOUT command 163
DELETE.CA command 163
DELETE.CAGRP command 164
DELETE.DB command 165
DELETE.DBDS command 166
DELETE.DBDSGRP command 167
DELETE.IC command 167
DELETE.LOG command (for OLDS) 168
DELETE.LOG command (for RLDS) 169
DELETE.LOG command (for SLDS) 169
DELETE.PART command
DBD keyword 172
PART keyword 172
syntax 172
DELETE.RECOV command 173
DELETE.REORG command 174
DELETE.SUBSYS command 175
DELETE.UIC command 176
deleting information
all change accumulation group records 164
all database data set records 166
all database records 165
allocation record of database data set 162
area data set 161
backout record 163
change accumulation run record 163
database data set group records 167
image copy data set records 167
nonstandard image copy data sets 176
online log data set records 168
recovery log data set records 169
recovery run record 173
reorganization records 174
subsystem records 175
system log data set records 169
DELMEM keyword, CHANGE.DBDSGRP command 94
DFSUCUM0 (Database Change Accumulation utility) 251
DFSUDMP0 (Database Image Copy utility) 253
DFSUICP0 (Online Database Image Copy utility) 253
DFSULTR0 (Log Recovery utility)
generating a job 186
DISPLAY TRACETABLE
keywords 9
output 9
syntax 9
usage notes 16
DISPLAY TRACETABLE command 9
BPE-defined trace table types 9
DISPLAY TRACETABLE command *(continued)*
CQS-defined trace tables 9
OM-defined trace tables 9
Repository-Server-defined trace table types 9
RM-defined trace tables 9
SCI-defined trace table types 9
DISPLAY USEREXIT command
BPE user exit types 24
CQS user exit types 24
DBRC user exit types 24
examples 28
HWS user exit types 24, 31
ODBM user exit types 25, 32
OM user exit types 25
output 28
RM user exit types 25
SCI user exit types 25
syntax 23
usage notes 23
DMB (data management block) numbers 389
DMB Table record 389
DSN commands
CHANGE.DBDS 89
CHANGE.PRILOG (for OLDS) 108
CHANGE.PRILOG (for RLDS) 110
CHANGE.PRILOG (for SLDS) 114
CHANGE.SECLOG (for OLDS) 133
CHANGE.SECLOG (for SLDS) 140
INIT.DBDS 295
NOTIFY.PRILOG (for OLDS) 356
NOTIFY.PRILOG (for RLDS) 360
NOTIFY.PRILOG (for SLDS) 363
NOTIFY.SECLOG (for OLDS) 375
NOTIFY.SECLOG (for RLDS) 378
DSPSCIX0 exit 124
DSSN keyword NOTIFY.ALLOC command 346
DSSTART keyword commands
CHANGE.PRILOG (for RLDS) 110
CHANGE.PRILOG (for SLDS) 115
CHANGE.SECLOG (for RLDS) 136
CHANGE.SECLOG (for SLDS) 140
DUAL keyword
CHANGE.RECON command 122

E

ENDRECOV keyword commands
CHANGE.SUBSYS 144
NOTIFY.SUBSYS 385
examples
F irlmproc,SET command
TIMEOUT 622
TRACE=nnn 622
F irlmproc,STATUS command 624
TRACE CT 637

F

F fdbrproc,DUMP command
description 429
syntax diagram 429

F fdbrproc,RECOVER command
description 429
syntax diagram 429
F fdbrproc,STATUS command
description 429
syntax diagram 429
F fdbrproc,STOP command
description 431
syntax diagram 431
F fdbrproc,TERM command
description 431
syntax diagram 431
F irlmproc,ABEND command
description 617
examples 617
syntax diagram 617
F irlmproc,DIAG,DELAY command
description 618
syntax diagram 618
F irlmproc,PURGE,imsname command
description 619
example 619
syntax diagram 619
F irlmproc,SET command
description 620
examples
TIMEOUT 622
TRACE=nnn 622
irlmproc parameter 620
restrictions 622
syntax diagram 620
TRACE=nnn parameter 622
F irlmproc,STATUS command
ALLD parameter 623
ALLI parameter 623
description 623
examples 624
irlmx parameter 623
STOR parameter 623
syntax diagram 623
TRACE parameter 623
F jobname,DUMP command
description 609
example 609
syntax diagram 609
F jobname,DUMPxxxx command
description 609
examples 610
syntax diagram 609
F jobname,FORCExxxx command
description 610
examples 611
syntax diagram 610
F jobname,RECONNECT command
description 611
example 611
syntax diagram 611
F jobname,RECONNSTR command
description 612
syntax diagram 612
F jobname,STOP command
description 613
example 613
syntax diagram 613
F jobname,STOPxxxx command
description 613
examples 614
F jobname,STOPxxxx command *(continued)*
syntax diagram 613
F reposervername,ADMIN command
description 647
examples 649
syntax diagram 647
F reposervername,AUDIT command
description 651
examples 651
syntax diagram 651
F reposervername,SECURITY command
description 652
examples 653
syntax diagram 652
F reposervername,SHUTDOWN command
description 653
examples 654
syntax diagram 653
Fast Database Recovery (FDBR) commands 427
FDBR commands 427
FILESEQ keyword commands
CHANGE.CA 67
CHANGE.IC 97
CHANGE.PRILOG (for RLDS) 111
CHANGE.PRILOG (for SLDS) 115
CHANGE.SECLOG (for RLDS) 136
CHANGE.SECLOG (for SLDS) 140
INIT.CA 285
INIT.IC 303
NOTIFY.CA 348
NOTIFY.IC 352
NOTIFY.PRILOG (for RLDS) 361
NOTIFY.PRILOG (for SLDS) 365
NOTIFY.REORG 372
NOTIFY.SECLOG (for RLDS) 379
NOTIFY.SECLOG (for SLDS) 382
FILESEQ2 keyword commands
CHANGE.IC 97
INIT.IC 303
NOTIFY.IC 352
NOTIFY.REORG 372
FORCE ODBA command
description 643
FORCER keyword commands
CHANGE.RECON 122
INIT.RECON 314
FPAREA keyword, GENJCL.RECOV command 213
FROMTIME keyword commands
LIST.HISTORY 332
LIST.LOG 336
full-precision time stamp parameters 59
FULLSEG keyword
CHANGE.DB command 76
CHANGE.DBDS 89
INIT.DB command 292
FULLSEG parameter
INIT.DBDS command 299

G

GAP keyword, CHANGE.PRILOG (for RLDS) 111, 112
Generalized Trace Facility (GTF) USR records 128
generating a job
Change Accumulation utility 181
Database Image Copy utility 189
Database Recovery utility 211, 218
Log Archive utility 177
Log Recovery utility 186
Online Database Image Copy utility 206
generating JCL 223
generating user output 223
GENJCL command 222
GENJCL commands, description 223
GENJCL.ARCHIVE command
syntax 177
TIMEFMT keyword 179
GENJCL.CA command 181
TIMEFMT keyword 183
GENJCL.CLOSE command 186
TIMEFMT keyword 187
GENJCL.IC command 189
SMSNOCIC keyword 192
SMSOFFLC keyword 192
SMSONLC keyword 192
TIMEFMT keyword 199
GENJCL.OIC command 206
TIMEFMT keyword 209
GENJCL.RECOV command 211
keywords
USEAREA 215
USEDBDS 215
USEIC 215
USEOLRRG 215
RESTORE keyword 215
TIMEFMT keyword 215
GENJCL.USER command 218
TIMEFMT keyword 221
GENMAX parameter
of the INIT.DBDS command 296
GROUP keyword commands 191
GENJCL.IC 191
GENJCL.OIC 207
GENJCL.RECOV 212
GENJCL.USER 219
LIST.DBDS 327
LIST.HISTORY 331
GRPMAX keyword commands
CHANGE.CAGRP 69
INIT.CAGRP 287
GRPMEM keyword
INIT.CAGRP command 287
GRPNAME keyword commands
CHANGE.CA 67
CHANGE.CAGRP 69
CHANGE.DBDSGRP 94
DELETE.CA 163
DELETE.CAGRP 164
DELETE.DBDSGRP 167
GENJCL.CA 182
INIT.CA 285
INIT.CAGRP 287
INIT.DBDSGRP 301
LIST.CAGRP 322
LIST.DBDSGRP 330
NOTIFY.CA 348

GTF (Generalized Trace Facility) USR
records 128

H

HALDB partition
registering 305
HISTORY command 331

I

ICDSN keyword commands
CHANGE.IC 97
defining 303
INIT.IC 303
NOTIFY.IC 352
NOTIFY.REORG 372
RECON
adding information 366
changing information 96
selecting 240
ICDSN2 keyword commands
CHANGE.IC 97
DELETE.IC 168
INIT.IC 303
NOTIFY.IC 352
NOTIFY.REORG 372
ICJCL keyword
commands
CHANGE.DBDS 89
GENJCL.CA 182
GENJCL.CLOSE 186
GENJCL.OIC 208
GENJCL.RECOV 214
GENJCL.USER 220
ICJCL parameter
skeletal JCL execution member 268
ICOFF keyword
CHANGE.DBDS command 89
ICON keyword
CHANGE.DBDS command 89
ICRCVJCL parameter
skeletal JCL execution member 275
ICREQ keyword
CHANGE.DB command 77
INIT.DB command 291
ICREQ keyword INIT.DB command 291
image copy 2 JCL 268
IMS Connect
commands 445
DELETE LINK 535
DELETE SENDCLNT 537
QUERY MSC command 571
QUERY RMTIMSCON command 581
type-2 commands 443
UPDATE MSC 597
UPDATE RMTIMSCON 603
IMS control region commands 607
IMS subsystem
starting
START irlmproc 629
IMSplex
OPENIP command 448
STARTIP command 462
STOPIP command 473
VIEWIP command 515
IMSPLEX
CHANGE.RECON 123
DSPSCIX0 124
INACTIVE keyword
DELETE.LOG command (for RLDS and SLDS) 169
INIT.ADS command
description 283
INIT.CA command 285
INIT.CAGRP command 287
INIT.DB command
description 289
INIT.DBDS command
description 294
INIT.DBDSGRP command 301
INIT.IC command 303
INIT.PART command 305
INIT.RECON command
description 311
initialize
area data set 283
change accumulation data set 285
change accumulation group 287
database 289
database data set 294
database data set groups 301
image copy data sets 303
RECON header records 311
INTERIM keyword commands
DELETE.LOG (for OLDS) 169
DELETE.LOG (for RLDS and SLDS) 171
INVALID keyword commands
CHANGE.CA 67
CHANGE.IC 97
INVALID2 keyword
CHANGE.IC command 97
description 97
IRLM commands 615
IRLMGRP= parameter
START irlmproc command 630
IRLMID keyword commands
CHANGE.SUBSYS 144
NOTIFY.SUBSYS 384
IRLMID= parameter
START irlmproc command 630
IRLMNM= parameter
START irlmproc command 630
irlmproc parameter
F irlmproc,SET command 620
irlmx parameter
F irlmproc,STATUS command 623

J

JCL (job control language)
generating 223
JCLOUT keyword
command GENJCL.ARCHIVE 178
commands
GENJCL.IC 198
JCLOUT parameter
skeletal JCL 224
JCLPDS keyword
commands 178
GENJCL.ARCHIVE 178
GENJCL.CA 182
JCLPDS keyword *(continued)*
commands *(continued)*
GENJCL.CLOSE 186
GENJCL.IC 198
GENJCL.OIC 208
GENJCL.RECOV 214
GENJCL.USER 220
JCLPDS parameter
skeletal JCL 224
JOB keyword commands
GENJCL.ARCHIVE 178
GENJCL.CA 182
GENJCL.CLOSE 187
GENJCL.IC 198
GENJCL.OIC 208
GENJCL.RECOV 214
GENJCL.USER 220
JOBJCL
skeletal JCL execution member 258

K

keyboard shortcuts ix
keyword
%IC 272
%ICDSN2 272
%ICDSN3 272
%ICDSN4 272
%ICUNIT2 272
%ICUNIT3 272
%ICUNIT4 272
%ICVCNT2 272
%ICVCNT3 272
%ICVCNT4 272
%ICVOLS2 272
%ICVOLS3 272
%ICVOLS4 272
substitution 226
keywords
NOOWNER 80
NORECOV 373
OFFLINE 372
OLRBYTES 79
OLRCAP 79, 292
OLRDBDS 80
OLRIMSID 80
OLRNOCAP 79, 292
OLRNOREL 79
OLRREL 79
OLRRGOFF 80
OLRRGON 80
OLRROOTS 79
OLRSEGS 79
ONLINE 372
RECOV 373
STOPTIME 373
keywords, symbolic 223

L

legal notices
notices 661
trademarks 661, 663
LIST keyword commands
GENJCL.ARCHIVE 178
GENJCL.CA 183

LIST keyword commands *(continued)*
GENJCL.CLOSE 187
GENJCL.IC 198
GENJCL.OIC 209
GENJCL.RECOV 214
GENJCL.USER 220
LIST.BKOUT command 321
LIST.CAGRP command 322
LIST.DB command 324
LIST.DBDS command 326
LIST.DBDSGRP command 329
LIST.HISTORY command 331
LIST.LOG (for a category of records) command 335
LIST.LOG (for a PRILOG family) command 334
LIST.RECON command
ALLOC record fields 426
BACKOUT record fields 425
CA record fields 425
CAGRP record fields 425
Data group record fields 426
DB (Fast Path) record fields 426
DB (HALDB) record fields 426
DB (IMS) record fields 426
DB (PART) record fields 426
DBDS (Fast Path) record fields 426
DBDS (non-Fast Path) record fields 426
description 338
fields displayed by command 420, 425, 426
IMAGE record fields 426
Log record fields 425
Online log record fields 425
RECON record fields 421
REORG record fields 426
sample listing 393
SSYS record fields 425
LIST.SUBSYS command 342
listing
backout records 321
category of records 335
change accumulation group 322
database 324
database data set 326
database groups 329
database histories 331
PRILOG family 334
RECON 338, 393
subsystem 342
LKASID keyword
CHANGE.DBDS command 90
LKASID parameter
INIT.DBDS command 298
LOCKTAB= parameter
START irlmproc command 630
Log Recovery utility (DFSULTR0)
generating a job 186
LOGCLJCL skeletal JCL execution member 266
LOGRET keyword of CHANGE.RECON 125
LOGRET parameter
INIT.RECON 315

M

MAS parameter
INIT.DBDS command 299
MAXCSA= parameter
START irlmproc command 630
maximum number of generations, image copy data sets, GENMAX keyword 89, 296
MAXOLDS keyword GENJCL.ARCHIVE command 179
MAXUSRS= parameter
START irlmproc command 630
MEMBER keyword
GENJCL.ARCHIVE command 179
GENJCL.CA 183
GENJCL.CLOSE 187
GENJCL.IC 199
GENJCL.OIC 209
GENJCL.RECOV 214
GENJCL.USER 219
INIT.DBDSGRP 301
members 224
MODIFY FDBRPROC commands
DUMP 429
RECOVER 429
STATUS 429
STOP 431
TERM 431
MODIFY IMS commands
DUMP 609
DUMPxxxx 609
FORCExxxx 610
RECONNECT 611
RECONNSTR 612
STOP 613
STOPxxxx 613
MODIFY irlmproc commands
ABEND 617
DIAG,DELAY 618
PURGE,imsname 619
SET 620
STATUS 623
MODIFY reposervername commands
ADMIN 647
AUDIT 651
SECURITY 652
SHUTDOWN 653
MULTIJOB keyword commands
GENJCL.IC 199
GENJCL.OIC 209
GENJCL.RECOV 214
GENJCL.USER 220

N

NEWTIME keyword commands
CHANGE.PRILOG (for RLDS) 111
CHANGE.PRILOG (for SLDS) 115
CHANGE.SECLOG (for RLDS) 136
CHANGE.SECLOG (for SLDS) 141
NEWVOL keyword commands
CHANGE.PRILOG (for RLDS) 111
CHANGE.PRILOG (for SLDS) 116
CHANGE.SECLOG (for RLDS) 136
CHANGE.SECLOG (for SLDS) 141
NOAUTH keyword commands
CHANGE.DB 76
CHANGE.DBDS 88
NOBACK keyword
CHANGE.DB command 77
NOBACKUP keyword
CHANGE.SUBSYS command 144
NOCATALG keyword
CHANGE.DBDS 86
CHANGE.PART 100
CHANGE.RECON 119
INIT.DB 290
INIT.DBDS 296, 371
INIT.PART 306
INIT.RECON 312
NOCATDS keyword commands
CHANGE.RECON 118
INIT.RECON 313
NOCFSTR2 keyword
CHANGE.DBDS command 88
NOCHECK keyword
CHANGE.RECON 124
NOCHECK parameter
INIT.RECON 314
NODBQUI keyword
CHANGE.DB command 76
CHANGE.DBDS command 88
NODBQUIH keyword
CHANGE.DB command 76
CHANGE.DBDS command 88
NODBQUIH keyword commands
GENJCL.IC 192
NODEFLT keyword commands
CHANGE.CAGRP 69
GENJCL.CA 183
GENJCL.IC 199
GENJCL.OIC 209
GENJCL.RECOV 215
GENJCL.USER 221
NOFORCER keyword commands
CHANGE.RECON 122
INIT.RECON 314
NOFULLSG keyword
CHANGE.DB command 76
CHANGE.DBDS 89
INIT.DB command 292
NOFULLSG parameter
INIT.DBDS command 299
NOICREQ keyword
CHANGE.DB command 77
INIT.DB command 291
NOICREQ keyword INIT.DB command 291
NOJOB keyword commands
GENJCL.ARCHIVE 178
GENJCL.CA 182
GENJCL.CLOSE 187
GENJCL.IC 198
GENJCL.OIC 208
GENJCL.RECOV 214
GENJCL.USER 220
NOLIST keyword commands
GENJCL.ARCHIVE 178
GENJCL.CA 183
GENJCL.CLOSE 187
GENJCL.IC 198
GENJCL.OIC 209

NOLIST keyword commands *(continued)*
GENJCL.RECOV 214
GENJCL.USER 220
NOLKASID keyword
CHANGE.DBDS command 90
NOLKASID parameter
INIT.DBDS command 298
NOMAS parameter
INIT.DBDS command 299
NONEW keyword commands 317
CHANGE.RECON 127
INIT.RECON 317
NONRECOV keyword
CHANGE.DB 78
nonstandard image copy data sets
RECON
adding information 386
changing information 145
deleting information 176
NOOWNER keyword 80
NOPREL keyword
CHANGE.DBDS command 90
NOPREO keyword
CHANGE.DBDS command 91
NORECOV keyword
CHANGE.DBDS command 91
NOTIFY.REORG 373
NOREUSE keyword
CHANGE.DBDS 91
command
CHANGE.CAGRP 69
INIT.CAGRP 287
INIT.DBDS 296
NORMAL keyword commands
CHANGE.PRILOG (for OLDS) 108
CHANGE.PRILOG (for RLDS) 110
CHANGE.PRILOG (for SLDS) 115
CHANGE.SECLOG (for OLDS) 133
CHANGE.SECLOG (for RLDS) 136
CHANGE.SECLOG (for SLDS) 140
CHANGE.SUBSYS 144
NOTIFY.SUBSYS 385
NOTIFY.ALLOC command 345
NOTIFY.BKOUT command 347
NOTIFY.CA command 348
NOTIFY.PRILOG command (for OLDS) 356
NOTIFY.PRILOG command (for RLDS) 359
NOTIFY.PRILOG command (for SLDS) 363
NOTIFY.RECOV command 366
NOTIFY.REORG
ALTER 373
NOTIFY.REORG command 370
NOTIFY.SECLOG command (for OLDS) 374
NOTIFY.SECLOG command (for RLDS) 377
NOTIFY.SECLOG command (for SLDS) 381
NOTIFY.SUBSYS command 384
NOTIFY.UIC command
description 386
NOVSO keyword 92
CHANGE.DBDS 92
INIT.DBDS 298
NXTOLDS keyword commands
NOTIFY.SECLOG (for OLDS) 376

O

ODBA commands 639
OFFLINE keyword 372
OICJCL keyword
commands 91
CHANGE.DBDS 91
INIT.DBDS 297
OICJCL parameter
skeletal JCL execution member 273
OLDS (online log data set)
changing information 107, 132
commands
CHANGE.PRILOG 107
CHANGE.SECLOG 133
DELETE.LOG (for OLDS) 168
GENJCL.ARCHIVE 177
GENJCL.CLOSE 187
NOTIFY.PRILOG (for OLDS) 356
NOTIFY.SECLOG (for OLDS) 375
RECON
adding information 356, 374
deleting information 168
selecting in JCL 233
OLDVOL keyword commands
CHANGE.PRILOG (for RLDS) 112
CHANGE.PRILOG (for SLDS) 116
CHANGE.SECLOG (for RLDS) 137
CHANGE.SECLOG (for SLDS) 141
OLRBYTES keyword 79
OLRCAP keyword 79, 292
OLRDBDS keyword 80
OLRIMSID keyword 80
OLRNOCAP keyword 79, 292
OLRNOREL keyword 79
OLRREL keyword 79
OLRRGOFF keyword 80
OLRRGON keyword 80
OLRROOTS keyword 79
OLRSEGS keyword 79
ONEJOB keyword commands
GENJCL.IC 199
GENJCL.OIC 209
GENJCL.RECOV 214
GENJCL.USER 220
online commands 50
Online Database Image Copy utility (DFSUICP0)
generating a job 206
Online Database Image Copy utility JCL 273
ONLINE keyword 372
ONLINE keyword commands
NOTIFY.IC 352
NOTIFY.SUBSYS 385
ONLINE keyword commands
LIST.SUBSYS 342
Open Database Access (ODBA) commands 639
OPENDS command 447
equivalent IMS Connect type-2 and z/OS commands 448
OPENIP command 448
OPENIP command *(continued)*
equivalent IMS Connect type-2 and z/OS commands 449
OPENPORT command 449
equivalent IMS Connect type-2 and z/OS commands 450

P

P cqsjob command
description 45
syntax diagram 45
P CSL command
description 41
keywords 41
shutdown 41
syntax diagram 41
P irlmproc command
description 633
example 633
syntax diagram 633
P reposervername command
description 657
syntax diagram 657
parameter
record type 228
selection criteria 229
PART keyword
DELETE.PART 172
PC= parameter
START irlmproc command 631
PRELOAD keyword
CHANGE.DBDS command 90
PREOPEN keyword
CHANGE.DBDS command 91
PRILOG family
listing records 334
PSB keyword commands 66
CHANGE.BKOUT 66
GENJCL.OIC 207
GENJCL.USER 221
NOTIFY.BKOUT 347
PURGLIST keyword
NOTIFY.CA command 348

Q

QUERY ALIAS
example 539
syntax 539
usage 539
QUERY ALIAS command 539
equivalent IMS Connect type-2 and WTOR commands 540
QUERY command
DB keyword
command comparison 649
similar to IMS commands 649
QUERY DATASTORE command 540
equivalent IMS Connect type-2 and WTOR commands 541
example 540
syntax 540
usage 540
QUERY IMSPLEX command 545

QUERY IMSPLEX command *(continued)*
equivalent IMS Connect type-2 and WTOR commands 546
example 545
syntax 545
usage 545
QUERY MEMBER command 547
equivalent IMS Connect type-2 and WTOR commands 547
example 547
output fields 548
syntax 547
usage 547
QUERY MSC command 571
equivalent IMS Connect type-2 and WTOR commands 572
output fields 572
QUERY PORT command 576
equivalent IMS Connect type-2 and WTOR commands 576
example 576
syntax 576
usage 576
usage notes 576
QUERY RMTIMSCON command 581
equivalent IMS Connect type-2 and WTOR commands 582
QUERY UOR command 585
equivalent IMS Connect type-2 and WTOR commands 586
example 585
syntax 585
usage 585

R

RCVTIME keyword
GENJCL.RECOV 215
NOTIFY.RECOV 368
READOFF keyword
CHANGE.DB command 81
READON keyword
CHANGE.DB command 81
rebuilding 389
RECDCT keyword commands
CHANGE.IC 98
NOTIFY.IC 352
NOTIFY.REORG 372
RECON
sample listing 393
RECON data set
adding records
database reorganizations 370
reorganization record 370
RECON1 keyword
BACKUP.RECON command 61
record type parameter 228
RECORDER command 450
equivalent IMS Connect type-2 and z/OS commands 451
RECOV keyword
NOTIFY.REORG 373
RECOVABL keyword
CHANGE.DB 78
INIT.DB 290
recovery
CHANGE.DBDS 91
recovery *(continued)*
database data set or area, adding information to the RECON data set 366
defining groups 301
log data set
deleting information 169
log data set, adding information to the RECON data set 359, 377
recovery control
data set
commands to generate JCL and user-defined output 223
listing 338
RECOVJCL keyword
commands
CHANGE.DBDS 92
INIT.DBDS 297
RECOVJCL parameter
skeletal JCL execution member 277
RECOVPD keyword
commands
INIT.CAGRP 287
INIT.DBDS 298
RECOVPD keyword commands
CHANGE.DBDS 92
INIT.DBDS 297
RECTIME keyword commands
CHANGE.CA 67
CHANGE.IC 97
CHANGE.UIC 146
DELETE.ALLOC 162
DELETE.CA 164
DELETE.IC 168
DELETE.RECOV 173
DELETE.REORG 174
DELETE.UIC 176
RECVJCL keyword
CHANGE.DBDS command 92
REFRESH CONVERTER command
equivalent IMS Connect type-2 and z/OS commands 452
REFRESH RACFUID command
equivalent IMS Connect type-2 and z/OS commands 453
REFRESH USEREXIT command
considerations 34
examples 35
output 35
recommendations 35
syntax 30
usage 30
reorganization of the RECON data set
deleting information 174
REPAIR.RECON command 389
repairing inconsistencies 389
REPLACE keyword
CHANGE.RECON command 122
RESTORE keyword
GENJCL.RECOV 215
restrictions
F irlmproc,SET command 622
REUSE keyword commands
CHANGE.CAGRP 69
CHANGE.DBDS 91
INIT.CAGRP 287
INIT.DBDS 296
RLDS (recovery log data set) 109
commands 110
CHANGE.PRILOG (for RLDS) 110
CHANGE.SECLOG (for RLDS) 134
deleting information from RECON 169
selecting 238
RUNTIMES keyword commands
CHANGE.PRILOG (for RLDS) 112
CHANGE.PRILOG (for SLDS) 116
CHANGE.SECLOG (for RLDS) 137
CHANGE.SECLOG (for SLDS) 141
NOTIFY.CA 348
NOTIFY.IC 352
NOTIFY.PRILOG (for OLDS) 356
NOTIFY.PRILOG (for RLDS) 360
NOTIFY.PRILOG (for SLDS) 363
NOTIFY.REORG 371
NOTIFY.SECLOG (for OLDS) 375
NOTIFY.SECLOG (for RLDS) 378
NOTIFY.UIC 387

S

SCOPE= parameter
START irlmproc command 631
select group, skeletal JCL 227
selection criteria parameter 229
service utilities
control statement parameters 228
SETOAUTO command 454
equivalent IMS Connect type-2 and z/OS commands 455
SETPWMC command 455
equivalent IMS Connect type-2 and z/OS commands 456
SETRACF command 457
equivalent IMS Connect type-2 and z/OS commands 457
SETRRS command 458
equivalent IMS Connect type-2 and z/OS commands 458
SETUIDC command
equivalent IMS Connect type-2 and z/OS commands 459
SHADOW parameter
INIT.DBDS command 300
SHARELVL keyword
INIT.DB 305
SHARELVL parameter
CHANGE.DB 81
SHUTDOWN CSL
description 39
syntax 39
usage notes 39
SHUTDOWN CSLPLEX
description 39
syntax 39
usage notes 39
SHUTDOWN MEMBER command 587
equivalent IMS Connect type-2 and WTOR commands 587
example 587
syntax 587
usage 587

skeletal JCL 275
coding execution members 226, 247
data set 224
default members explained 224
execution members explained 224
generating JCL and user-defined output 223
IBM-supplied 224, 258
modifying 249
writing your own 249
skeletal JCL for DBRC 222
SLDS (system log data set)
adding information to the RECON data set 363, 381
CHANGE.PRILOG (for SLDS) 113
CHANGE.SECLOG (for SLDS) 139
changing information 113, 139
deleting information from RECON 169
NOTIFY.PRILOG (for SLDS) 363
NOTIFY.SECLOG (for SLDS) 381
selecting 235
SMS concurrent image copy, SMSCIC keyword 192
SMSCIC (SMS concurrent image copy) 192
SMSCIC keyword command
GENJCL.IC 192
SMSNOCIC keyword
GENJCL.IC 192
SMSOFFLC keyword
GENJCL.IC 192
SMSONLC keyword
GENJCL.IC 192
specifying log retention intervals, CHANGE.RECON 125
SSID keyword commands
CHANGE.BKOUT 64
CHANGE.DB 82
CHANGE.PRILOG (for OLDS) 108
CHANGE.PRILOG (for SLDS) 116
CHANGE.RECON 126
CHANGE.SECLOG (for OLDS) 134
CHANGE.SECLOG (for SLDS) 142
CHANGE.SUBSYS 143
DELETE.BKOUT 163
DELETE.LOG (for OLDS) 169
DELETE.SUBSYS 175
GENJCL.ARCHIVE 179
GENJCL.CLOSE 187
GENJCL.USER 221
INIT.RECON 316
LIST.BKOUT 321
LIST.LOG (for a category of records) 338
LIST.LOG (for a PRILOG family) 335
LIST.RECON 340
LIST.SUBSYS 342, 343
NOTIFY.BKOUT 347
NOTIFY.PRILOG (for OLDS) 358
NOTIFY.PRILOG (for RLDS) 361
NOTIFY.SECLOG (for OLDS) 376
NOTIFY.SECLOG (for RLDS) 380
NOTIFY.SUBSYS 384
standard form of time stamps, parameters of DBRC commands 53
START FDBRPROC command
description 433
syntax diagram 433
START irlmproc command
DEADLOK= parameter 629
description 629
IRLMGRP= parameter 630
IRLMID= parameter 630
IRLMNM= parameter 630
LOCKTAB= parameter 630
MAXCSA= parameter 630
MAXUSRS= parameter 630
PC= parameter 631
SCOPE= parameter 631
syntax diagram 629
TRACE= parameter 631
STARTDS command 460
equivalent IMS Connect type-2 and z/OS commands 461
STARTIA command 461
equivalent IMS Connect type-2 and z/OS commands 462
STARTIME keyword commands
CHANGE.PRILOG (for RLDS) 110
CHANGE.PRILOG (for SLDS) 114
CHANGE.SECLOG (for RLDS) 134
CHANGE.SECLOG (for SLDS) 139
DELETE.LOG (for RLDS and SLDS) 169
NOTIFY.ALLOC 346
NOTIFY.PRILOG (for OLDS) 357
NOTIFY.PRILOG (for RLDS) 360
NOTIFY.PRILOG (for SLDS) 364
NOTIFY.SECLOG (for OLDS) 375
NOTIFY.SECLOG (for RLDS) 378
NOTIFY.SECLOG (for SLDS) 382
STARTIP command 462
equivalent IMS Connect type-2 and z/OS commands 463
STARTMSC command 452, 464
equivalent IMS Connect type-2 and z/OS commands 464
STARTNEW keyword commands 127
CHANGE.RECON 127
INIT.RECON 317
STARTOD command 465
equivalent IMS Connect type-2 and z/OS commands 466
STARTPT command 466
equivalent IMS Connect type-2 and z/OS commands 467
STARTRCV keyword commands
CHANGE.SUBSYS 144
NOTIFY.SUBSYS 385
STARTRMT command 467
equivalent IMS Connect type-2 and z/OS commands 468
STATUS keyword
LIST.RECON command 339
STOP 45
STOP cqsjob command
description 45
STOP CSL command
description 41
keywords 41
shutdown 41
STOP CSL command *(continued)*
syntax diagram 41
STOP irlmproc command
description 633
example 633
STOP reposervername command 657
STOPCLNT command 469
equivalent IMS Connect type-2 and z/OS commands 470
STOPDS command 470
equivalent IMS Connect type-2 and z/OS commands 471
STOPIA command 471
equivalent IMS Connect type-2 and z/OS commands 472
STOPIP command 473
equivalent IMS Connect type-2 and z/OS commands 474
STOPLINK command 474
equivalent IMS Connect type-2 and z/OS commands 475
STOPMSC command 476
equivalent IMS Connect type-2 and z/OS commands 477
STOPOD command 477
equivalent IMS Connect type-2 and z/OS commands 478
STOPPORT command 478
equivalent IMS Connect type-2 and z/OS commands 479
STOPRMT command 480
equivalent IMS Connect type-2 and z/OS commands 481
STOPSCLN command 481
equivalent IMS Connect type-2 and z/OS commands 482
STOPTIME 373
STOPTIME keyword commands
CHANGE.IC 98
NOTIFY.IC 352
STOPTIME keyword commands
NOTIFY.CA 348
STOR parameter
F irlmproc,STATUS command 623
SUBSET keyword commands
CHANGE.CA 68
NOTIFY.CA 348
subsystem
changing information 143
listing 342
RECON 175
adding information 384
deleting information 175
symbolic keywords 226
control keywords 226
%DELETE 230
%ENDDEL 230
%ENDSEL 227
%SELECT 227, 233
%SET MEMBER 227, 231
%SET TIMEFMT 227, 232
description 224, 227, 247
JCL execution member
%ALLSEL 246
%ALLTIME 246
%ALLUSID 246
%CADSN 244

symbolic keywords *(continued)*
JCL execution member *(continued)*
%CAFSEQ 244
%CALGTM 244
%CAODSN 231
%CASEL 244
%CATIME 244
%CAUNIT 244
%CAVCNT 244
%CAVOLS 244
%DALTIME 246
%DBADDN 247
%DBADSAV 247
%DBDDN 246, 247
%DBDSDEL 247
%DBDSN 247
%DBDSNRV 247
%DBNAME 246, 247
%DBTYPE 246, 247
%DBUSID 247
%DDNAME 228
%ICCAT 242
%ICDSN 241
%ICFSEQ 241
%ICSEL 241
%ICSTOP 242
%ICTIME 242
%ICTYPE 241
%ICUNIT 242
%ICUSID 242
%ICVCNT 242
%ICVOLS 242
%LOGDSN 239
%LOGETIM 240
%LOGFRID 240
%LOGFSEQ 239
%LOGLRID 240
%LOGMERG 240
%LOGONL 240
%LOGRMT 240
%LOGSEL 240
%LOGSTIM 239
%LOGUNIT 239
%LOGVOLS 239
%OLDCTIM 234
%OLDFRID 234
%OLDLRID 234
%OLDOTIM 234
%OLDSDDN 234
%OLDSDSN 234
%OLDSSEL 234
%OLDSTYP 234
%PLGTIME 246
%SLDETIM 236
%SLDFRID 236
%SLDFSEQ 236
%SLDLRID 236
%SLDRMT 236
%SLDSDDN 236
%SLDSSEL 236
%SLDSTIM 236
%SLDUNIT 236
%SLDVOLS 236
%SSID 226
%TIME 226
symbolic keywords 226, 246
recognized by DBRC 249
symbolic keywords *(continued)*
user-defined 227
symbolic keywords, JCL execution member
substitution 223
symbolic keywords
%SLDCHKT 237
syntax diagram
how to read viii
syntax, DBRC command 50

T

TAPEUNIT keyword commands
CHANGE.RECON 127
INIT.RECON 317
time qualifier 229
time stamp 52
conversions and examples 57
DBRC commands affected by format 58
specifying zero values 56
standard default settings for values 58
standard format 53
TIMEFMT parameter 54
two-digit year input 57
TIMEFMT keyword
GENJCL.ARCHIVE 179
GENJCL.CA 183
GENJCL.CLOSE 187
GENJCL.IC 199
GENJCL.OIC 209
GENJCL.RECOV 215
GENJCL.USER 221
TIMEFMT keyword sublist
CHANGE.RECON 129
TIMEFMT parameter sublist
default settings 56
order of precedence of the subparameters 56
TIMEZIN keyword
CHANGE.RECON command 129
TIMEZONE keyword
CHANGE.RECON command 128
TOTIME keyword commands
DELETE.LOG (for RLDS and SLDS) 169
GENJCL.ARCHIVE 178
LIST.HISTORY 332
LIST.LOG 336
TRACE CT command
description 635
examples 637
IRLM external writer sample procedure 636
IRLM load module 635
keywords 635
parameters 635
syntax diagram 635
z/OS link list 635
TRACE parameter
F irlmproc,STATUS command 623
TRACE= parameter
START irlmproc command 631
TRACE=nnn parameter
F irlmproc,SET command 622
TRACEOFF keyword
CHANGE.RECON command 127
TRACEON keyword
CHANGE.RECON command 127
TRACETABLE 9
command verbs 9
TRACK keyword
NOTIFY.RECOV command 368
TRACKING parameter
CHANGE.SUBSYS command 145
trademarks 661, 663
TYPEFP keyword commands
CHANGE.DB 82
INIT.DB 292
LIST.DB 324
TYPEIMS keyword commands
CHANGE.DB 82
INIT.DB 292
LIST.DB 324

U

UDATA keyword commands
CHANGE.UIC 146
NOTIFY.CA 348
NOTIFY.UIC 387
UNAUTH keyword
CHANGE.DB command 82
UNAUTH keyword
restrictions 74
using 74
UNAVAIL keyword commands
CHANGE.ADS 63
CHANGE.PRILOG (for OLDS) 108
CHANGE.SECLOG (for OLDS) 133
INIT.ADS 284
UNIT keyword commands
CHANGE.CA 68
CHANGE.IC 98
CHANGE.PRILOG (for RLDS) 112
CHANGE.PRILOG (for SLDS) 117
CHANGE.SECLOG (for RLDS) 137
CHANGE.SECLOG (for SLDS) 142
GENJCL.CA 183
GENJCL.IC 199, 200
GENJCL.OIC 209
INIT.CA 285
INIT.IC 303
NOTIFY.CA 348
NOTIFY.IC 352
NOTIFY.PRILOG 365
NOTIFY.PRILOG (for RLDS) 362
NOTIFY.REORG 373
NOTIFY.SECLOG (for RLDS) 380
NOTIFY.SECLOG (for SLDS) 383
UNIT2 keyword commands
CHANGE.IC 98
GENJCL.OIC 210
INIT.IC 303
NOTIFY.IC 352
NOTIFY.REORG 373
UOR (unit of recovery) keyword commands
CHANGE.BKOUT 64
NOTIFY.BKOUT 347
UORTIME keyword
CHANGE.BKOUT command 65

UPDATE ALIAS
example 588
syntax 588
usage 588
UPDATE ALIAS command 588
equivalent IMS Connect type-2 and WTOR commands 589
UPDATE CONVERTER
equivalent IMS Connect type-2 and WTOR commands 590
example 589
syntax 589
usage 589
UPDATE CONVERTER command 589
UPDATE DATASTORE
equivalent IMS Connect type-2 and WTOR commands 591
example 591
syntax 591
usage 591
UPDATE DATASTORE command 591
UPDATE IMSPLEX
equivalent IMS Connect type-2 and WTOR commands 593
example 592
syntax 592
usage 592
UPDATE IMSPLEX command 592
UPDATE MEMBER command 594
equivalent IMS Connect type-2 and WTOR commands 596
example 594
syntax 594
usage 594
UPDATE MSC command 597
equivalent IMS Connect type-2 and WTOR commands 599
UPDATE ODBM
example 599
syntax 599
usage 599
UPDATE ODBM command 599
equivalent IMS Connect type-2 and WTOR commands 600
UPDATE PORT command 601
equivalent IMS Connect type-2 and WTOR commands 601
example 601
syntax 601
usage 601
UPDATE RACFUID
example 602
syntax 602
usage 602
UPDATE RACFUID command 602
equivalent IMS Connect type-2 and WTOR commands 603
UPDATE RMTIMSCON command 603
equivalent IMS Connect type-2 and WTOR commands 605
UPDATE TRACETABLE
output 20
syntax 15
UPDATE TRACETABLE command 15
BPE-defined trace table types 15
CQS-defined trace table types 17
DBRC-defined trace table types 18
UPDATE TRACETABLE command *(continued)*
IMS Connect-defined trace table types 9, 17
OM-defined trace table types 18
RM-defined trace table types 18
SCI-defined trace table types 18
USEAREA keyword, GENJCL.RECOV command 215
USEDBDS keyword, GENJCL.RECOV command 215
USEIC keyword, GENJCL.RECOV command 215
USEOLRRG keyword, GENJCL.RECOV command 215
user-supplied exit routines
refresh 33
USEREXIT commands 23
USERKEYS keyword
GENJCL.ARCHIVE command 179
GENJCL.CA 183
GENJCL.CLOSE 188
GENJCL.IC 200
GENJCL.OIC 210
GENJCL.RECOV 216
GENJCL.USER 221
USERRCOV keyword
CHANGE.DB 78
USR records, GTF (Generalized Trace Facility) 128

V

VALID keyword commands
CHANGE.CA 67
CHANGE.IC 97
VALID2 keyword
CHANGE.IC command 97
VIEWDS command 483
examples 487
keywords 483
output fields 484
syntax 483
usage notes 484
VIEWHWS command 488
examples 504
keywords 488
output fields 489
syntax 488
usage notes 488
VIEWIA command 512
VIEWIP command 515
VIEWMSC command 517
equivalent IMS Connect type-2 and z/OS commands 517
output fields 518
VIEWPORT command 521
equivalent IMS Connect type-2 and z/OS commands 522
examples 524
keywords 521
output fields 522
syntax 521
usage notes 522
VIEWRMT command 526
equivalent IMS Connect type-2 and z/OS commands 527
VIEWRMT command *(continued)*
output fields 527
VIEWUOR command 530
equivalent IMS Connect type-2 and z/OS commands 531
VOLLIST keyword commands
CHANGE.CA 68
CHANGE.IC 98
CHANGE.PRILOG (for RLDS) 112
CHANGE.PRILOG (for SLDS) 117
CHANGE.SECLOG (for RLDS) 137
CHANGE.SECLOG (for SLDS) 142
GENJCL.CA 184
GENJCL.IC 200
GENJCL.OIC 210
INIT.CA 285
INIT.IC 303
NOTIFY.CA 348
NOTIFY.IC 352
NOTIFY.REORG 373
VOLLIST2 keyword commands
CHANGE.IC 98
GENJCL.IC 201
GENJCL.OIC 210
INIT.IC 303
NOTIFY.IC 352
NOTIFY.REORG 373
VOLNUM keyword
GENJCL.CA command 184
VOLSER keyword commands
NOTIFY.PRILOG (for RLDS) 362
NOTIFY.PRILOG (for SLDS) 365
NOTIFY.SECLOG (for RLDS) 380
NOTIFY.SECLOG (for SLDS) 383
VSO keyword
CHANGE.DBDS 92
INIT.DBDS 298

Z

z/OS
command
CANCEL 641
F fdbrproc 429
F fdbrproc,DUMP command 429
F fdbrproc,RECOVER command 429
F fdbrproc,STATUS command 429
F fdbrproc,STOP command 431
F fdbrproc,TERM command 431
F irlmproc,ABEND command 617
F irlmproc,DIAG,DELAY command 618
F irlmproc,PURGE,imsname command 619
F irlmproc,SET command 620
F irlmproc,STATUS command 623
F jobname,DUMP command 609
F jobname,FORCExxxx command 610
F jobname,RECONNECT command 611
F jobname,RECONNSTR command 612
F jobname,STOP command 613
F jobname,STOPxxxx command 613

z/OS *(continued)*
command *(continued)*
F reposervername 647
F reposervername,ADMIN command 647
F reposervername,AUDIT command 651
F reposervername,SECURITY command 652
F reposervername,SHUTDOWN command 653
FORCE 643
P cqsjob 45
P CSL command 41
P irlmproc command 633
P reposervername command 657
S irlmproc command 629
TRACE CT 635
z/OS command
IMS Connect 435
START FDBRPROC 433
z/OS commands
DELETE CLIENT 534
DELETE LINK 535
DELETE SENDCLNT 537
invocation 533
QUERY ALIAS 539
QUERY DATASTORE 540
QUERY IMSPLEX 545
QUERY MEMBER 547
QUERY MSC 571
QUERY PORT 576
QUERY RMTIMSCON 581
QUERY UOR 585
SHUTDOWN MEMBER 587
syntax 533
UPDATE ALIAS 588
UPDATE CONVERTER 589
UPDATE DATASTORE 591
UPDATE IMSPLEX 592
UPDATE MEMBER 594
UPDATE MSC 597
UPDATE ODBM 599
UPDATE PORT 601
UPDATE RACFUID 602
UPDATE RMTIMSCON 603
wildcard support 534

Product Number: 5635-A06
5655-DS5
5655-TM4

Printed in USA

SC27-6782-00

Spine information:

IMS **Version 15.1.0** **Commands, Volume 3: IMS Component and z/OS Commands** IBM