

IMS  
Version 14

*Commands, Volume 1:  
IMS Commands A - M  
(March 1, 2018 edition)*

**IBM**



IMS  
Version 14

*Commands, Volume 1:  
IMS Commands A - M  
(March 1, 2018 edition)*

**IBM**

**Note**

Before you use this information and the product it supports, read the information in “Notices” on page 977.

March 1, 2018 edition.

This edition applies to IMS 14 (program number 5635-A05), IMS Database Value Unit Edition, V14.01.00 (program number 5655-DSE), IMS Transaction Manager Value Unit Edition, V14.01.00 (program number 5655-TM3), 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

<b>About this information . . . . .</b>	<b>vii</b>
Prerequisite knowledge . . . . .	vii
How new and changed information is identified . . . . .	vii
How to read syntax diagrams . . . . .	viii
Accessibility features for IMS 14 . . . . .	ix
How to send your comments . . . . .	x

## **Chapter 1. IMS command language overview . . . . . 1**

How to enter IMS commands . . . . .	2
IMS MFS 3270 master terminal format . . . . .	2
Using Multiple Console Support (MCS) consoles . . . . .	5
Outstanding reply numbers on z/OS consoles . . . . .	5
Maximum length of command input from z/OS consoles . . . . .	5
Multisegment command input . . . . .	6
Commands in a DBCTL environment . . . . .	7
Commands from an LU 6.2 device . . . . .	8
Qualifying network LU names and commands . . . . .	9
Issuing commands to the OM API . . . . .	9
Batch SPOC format . . . . .	10
Command responses . . . . .	12
Command responses to OM . . . . .	13
Responses to LU 6.2 devices . . . . .	14
Sending messages to the z/OS system console . . . . .	14
Sending messages to the IMS master terminal . . . . .	14
IMS terminal command examples . . . . .	15
TSO SPOC command examples . . . . .	15
OM API command examples . . . . .	16
IMS type-1 command format . . . . .	16
IMS type-2 command format . . . . .	22
Sending commands to the IMSplex . . . . .	25
Command processing in an IMSplex . . . . .	25
Command characteristics . . . . .	27
Terminal security defaults for IMS type-1 commands . . . . .	27
Commands recovered during emergency restart . . . . .	29
Command processing in a dynamic resource definition environment . . . . .	30
IMS type-1 commands logged to the secondary master terminal . . . . .	31
IMS type-1 commands supported from an AO application . . . . .	31
Command security when using OM . . . . .	37
Commands mirrored on an XRF alternate . . . . .	37
Commands supported on the XRF alternate . . . . .	38
Commands and keywords supported on an RSR tracking subsystem . . . . .	40
IMS-supported terminals in an RSR environment . . . . .	43
IMS type-1 commands supported from LU 6.2 devices and OTMA . . . . .	43
Terminal security defaults for LU 6.2 devices and OTMA . . . . .	44
Commands and keywords supported by the OM API . . . . .	45
Equivalent IMS type-1 and type-2 commands . . . . .	48

Equivalent IMS Connect WTOR, z/OS, and type-2 commands . . . . .	57
--	----

## **Chapter 2. Command keywords and their synonyms . . . . . 63**

## **Chapter 3. Commands and keywords valid in DBCTL . . . . . 85**

## **Chapter 4. Commands and keywords valid in DCCTL . . . . . 87**

## **Chapter 5. List of reserved words . . . . . 91**

## **Chapter 6. IMS type-1 commands that are valid in ETO . . . . . 95**

## **Chapter 7. /ACTIVATE command . . . . . 97**

## **Chapter 8. /ALLOCATE command . . . . . 99**

## **Chapter 9. /ASSIGN command . . . . . 101**

## **Chapter 10. /BROADCAST command . . . . . 121**

## **Chapter 11. /CANCEL command . . . . . 127**

## **Chapter 12. /CHANGE commands . . . . . 129**

/CHANGE APPC command . . . . .	129
/CHANGE CCTL command . . . . .	130
/CHANGE CPLOG command . . . . .	132
/CHANGE DESC command . . . . .	132
/CHANGE DIR MFS command . . . . .	134
/CHANGE FDR command . . . . .	135
/CHANGE LINK command . . . . .	136
/CHANGE NODE command . . . . .	140
/CHANGE SUBSYS command . . . . .	144
/CHANGE SURV command . . . . .	145
/CHANGE TRAN command . . . . .	148
/CHANGE UOR command . . . . .	149
/CHANGE USER command . . . . .	151

## **Chapter 13. /CHECKPOINT command . . . . . 153**

## **Chapter 14. /CLSDST command . . . . . 161**

## **Chapter 15. /COMPT command . . . . . 165**

## **Chapter 16. /CQCHKPT command . . . . . 169**

## **Chapter 17. /CQQUERY command . . . . . 171**

<b>Chapter 18. /CQSET command . . . .</b>	<b>175</b>	/DISPLAY AFFIN command . . . . .	554
<b>Chapter 19. CREATE commands . . . .</b>	<b>177</b>	/DISPLAY AOITKN command . . . . .	556
CREATE DB command . . . . .	177	/DISPLAY APPC command . . . . .	557
CREATE DBDESC command . . . . .	185	/DISPLAY AREA command . . . . .	560
CREATE IMSCON commands . . . . .	191	/DISPLAY ASMT command . . . . .	565
CREATE IMSCON TYPE(DATASTORE)		/DISPLAY CCTL command . . . . .	575
command . . . . .	191	/DISPLAY CONV command . . . . .	578
CREATE IMSCON TYPE(IMPSPLEX) command	202	/DISPLAY CPLOG command . . . . .	582
CREATE IMSCON TYPE(PORT) command . . . . .	208	/DISPLAY CQS command . . . . .	583
CREATE LTERM command . . . . .	218	/DISPLAY DB command . . . . .	583
CREATE MSLINK command . . . . .	222	/DISPLAY DBD command . . . . .	594
CREATE MSNAME command . . . . .	231	/DISPLAY DESC command . . . . .	595
CREATE MSPLINK command . . . . .	236	/DISPLAY FDR command . . . . .	596
CREATE OTMADESC command . . . . .	250	/DISPLAY FPV command . . . . .	597
CREATE PGM command . . . . .	260	/DISPLAY HSB command . . . . .	599
CREATE PGMDESC command . . . . .	271	/DISPLAY HSSP command . . . . .	602
CREATE RTC command . . . . .	281	/DISPLAY LINE command . . . . .	603
CREATE RTCDESC command . . . . .	287	/DISPLAY LINK command . . . . .	607
CREATE TRAN command . . . . .	292	/DISPLAY LTERM command . . . . .	611
CREATE TRANDESC command . . . . .	316	/DISPLAY LUNAME command . . . . .	614
		/DISPLAY MASTER command . . . . .	620
		/DISPLAY MODIFY command . . . . .	621
<b>Chapter 20. /DBDUMP command . . . .</b>	<b>339</b>	/DISPLAY MSNAME command . . . . .	635
		/DISPLAY NODE command . . . . .	637
<b>Chapter 21. /DBRECOVERY command</b>	<b>345</b>	/DISPLAY OASN SUBSYS command . . . . .	648
		/DISPLAY OLDS command . . . . .	649
<b>Chapter 22. /DELETE command . . . .</b>	<b>355</b>	/DISPLAY OTMA command . . . . .	652
		/DISPLAY OVERFLOWQ command . . . . .	658
<b>Chapter 23. DELETE commands . . . .</b>	<b>357</b>	/DISPLAY PGM command . . . . .	661
DELETE DB command . . . . .	357	/DISPLAY POOL command . . . . .	663
DELETE DBDESC command . . . . .	365	/DISPLAY PSB command . . . . .	682
DELETE DEFN command . . . . .	371	/DISPLAY PTERM command . . . . .	684
DELETE IMSCON commands . . . . .	390	/DISPLAY Q command . . . . .	686
DELETE IMSCON TYPE(DATASTORE)		/DISPLAY QCNT command . . . . .	689
command . . . . .	390	/DISPLAY RECOVERY command . . . . .	694
DELETE IMSCON TYPE(IMPSPLEX) command	395	/DISPLAY RTCODE command . . . . .	700
DELETE IMSCON TYPE(PORT) command . . . . .	400	/DISPLAY SHUTDOWN STATUS command . . . . .	701
DELETE LE command . . . . .	405	/DISPLAY STATUS command . . . . .	707
DELETE LTERM command . . . . .	412	/DISPLAY STRUCTURE command . . . . .	714
DELETE MSLINK command . . . . .	416	/DISPLAY SUBSYS command . . . . .	716
DELETE MSNAME command . . . . .	421	/DISPLAY SYSID TRANSACTION command . . . . .	719
DELETE MSPLINK command . . . . .	427	/DISPLAY TIMEOVER command . . . . .	720
DELETE OTMADESC command . . . . .	432	/DISPLAY TMEMBER command . . . . .	721
DELETE PGM command . . . . .	436	/DISPLAY TRACE command . . . . .	734
DELETE PGMDESC command . . . . .	441	/DISPLAY TRACKING STATUS command . . . . .	744
DELETE RTC command . . . . .	447	/DISPLAY TRAN command . . . . .	752
DELETE RTCDESC command . . . . .	452	/DISPLAY UOR command . . . . .	757
DELETE TRAN command . . . . .	457	/DISPLAY USER command . . . . .	760
DELETE TRANDESC command . . . . .	463		
<b>Chapter 24. /DEQUEUE command . . . .</b>	<b>471</b>	<b>Chapter 27. /END command . . . . .</b>	<b>767</b>
<b>Chapter 25. /DIAGNOSE commands</b>	<b>479</b>	<b>Chapter 28. /ERESTART command</b>	<b>769</b>
/DIAGNOSE SET command . . . . .	479	<b>Chapter 29. /EXCLUSIVE command</b>	<b>781</b>
/DIAGNOSE SNAP command . . . . .	480	<b>Chapter 30. /EXIT command . . . . .</b>	<b>783</b>
<b>Chapter 26. /DISPLAY commands . . . .</b>	<b>521</b>	<b>Chapter 31. EXPORT command . . . . .</b>	<b>787</b>
Status and attributes for the /DISPLAY command	523		
/DISPLAY ACT command . . . . .	537		

<b>Chapter 32. /FORMAT command . . . . .</b>	<b>819</b>	<b>Chapter 40. /MODIFY command . . . . .</b>	<b>959</b>
<b>Chapter 33. /HOLD command . . . . .</b>	<b>821</b>	<b>Chapter 41. /MONITOR command . . . . .</b>	<b>967</b>
<b>Chapter 34. /IDLE command . . . . .</b>	<b>823</b>	<b>Chapter 42. /MSASSIGN command . . . . .</b>	<b>969</b>
<b>Chapter 35. IMPORT commands . . . . .</b>	<b>827</b>	<b>Chapter 43. /MSVERIFY command . . . . .</b>	<b>973</b>
IMPORT DEFN SOURCE(CATALOG) command . . . . .	827	<b>Notices . . . . .</b>	<b>977</b>
IMPORT DEFN SOURCE(REPO   RDDS)		Programming interface information . . . . .	979
command . . . . .	848	Trademarks . . . . .	979
<b>Chapter 36. INITIATE commands . . . . .</b>	<b>891</b>	Terms and conditions for product documentation . . . . .	980
INITIATE OLC command . . . . .	891	IBM Online Privacy Statement. . . . .	980
INITIATE OLC error handling. . . . .	933	<b>Bibliography. . . . .</b>	<b>983</b>
INITIATE OLREORG command . . . . .	936	<b>Index . . . . .</b>	<b>X-1</b>
<b>Chapter 37. /LOCK command . . . . .</b>	<b>949</b>		
<b>Chapter 38. /LOG command . . . . .</b>	<b>955</b>		
<b>Chapter 39. /LOOPTEST command . . . . .</b>	<b>957</b>		



---

## About this information

These topics provide command syntax and usage information for the IMS™ type-1 and type-2 commands /ACTIVATE through /MSVERIFY. The topics also describe the IMS command language and how to send commands to IMS in different environments. Information about all non-type-1 and non-type-2 IMS commands is in *IMS Version 14 Commands, Volume 3: IMS Component and z/OS Commands*.

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 14 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).

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).



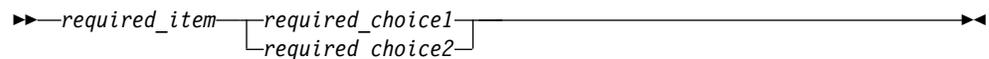
- Optional items appear below the main path.



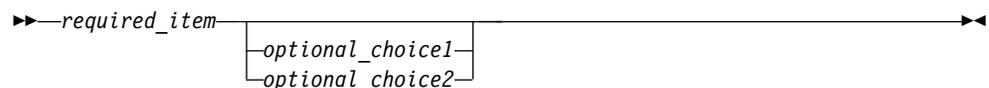
If an optional item appears above the main path, that item has no effect on the execution of the syntax element and is used only for readability.



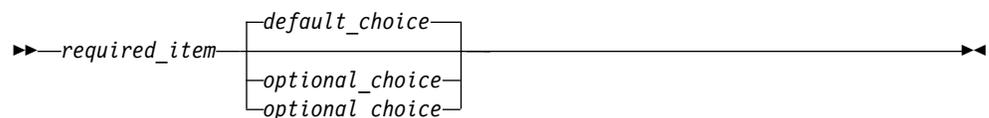
- If you can choose from two or more items, they appear vertically, in a stack. If you *must* choose one of the items, one item of the stack appears on the main path.



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



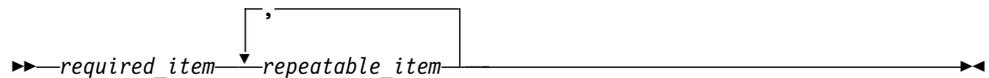
If one of the items is the default, it appears above the main path, and the remaining choices are shown below.



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

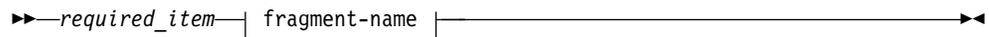


If the repeat arrow contains a comma, you must separate repeated items with a comma.



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 14

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 14. 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 14 ISPF panel functions by using a keyboard or keyboard shortcut keys.

For information about navigating the IMS 14 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 14 is available in IBM Knowledge Center.

## IBM and accessibility

See the *IBM Human Ability and Accessibility Center* at [www.ibm.com/able](http://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](mailto: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.

---

## Chapter 1. IMS command language overview

You can use the IMS master terminal, TSO SPOC, or the OM API to receive messages, issue commands, and receive command responses.

IMS supports two types of command formats to manage IMS systems and resources: IMS type-1 commands and IMS type-2 commands.

You can enter IMS type-1 commands through multiple terminal types and applications, and most can be entered through the OM API. IMS type-2 commands can be entered only from the OM API. z/OS commands can be entered only from a z/OS system console, not from an IMS master terminal.

### IMS type-1 commands

IMS type-1 commands can be entered from IMS terminals, master terminals, system consoles, EMCS consoles, IMS application programs through CMD calls and ICMD calls, and LU 6.2 and OTMA applications. IMS type-1 commands are in the original or classic IMS command syntax, and are preceded by a command recognition character if the command is issued from one of these interfaces.

Most IMS type-1 commands can also be entered from the OM API, but not all of them. If the command is issued from the OM API, the command recognition character is optional. Use RACF<sup>®</sup> or an equivalent security product for security checking. IMS type-1 command output from the OM API is in message format within an XML tag structure.

### IMS type-2 commands

IMS type-2 commands can be entered only from the OM API, not from a master or remote terminal. Either RACF (or an equivalent security product) or command security can be used for security checking. IMS type-2 command output is in XML tag structure.

Both IMS type-1 commands and type-2 commands can be entered from the OM API (for example, TSO SPOC or user-written programs using OM API).

Subsections:

- “How to enter IMS commands” on page 2
- “Command responses” on page 12
- “Sending messages to the z/OS system console” on page 14
- “Sending messages to the IMS master terminal” on page 14
- “IMS terminal command examples” on page 15
- “TSO SPOC command examples” on page 15
- “OM API command examples” on page 16
- “IMS type-1 command format” on page 16
- “Command processing in an IMSplex” on page 25
- “IMS type-2 command format” on page 22
- “Command characteristics” on page 27
- “Terminal security defaults for IMS type-1 commands” on page 27
- “Commands recovered during emergency restart” on page 29
- “IMS type-1 commands logged to the secondary master terminal” on page 31

- “Command security when using OM” on page 37
- “IMS type-1 commands supported from an AO application” on page 31
- “Commands mirrored on an XRF alternate” on page 37
- “Commands supported on the XRF alternate” on page 38
- “Commands and keywords supported on an RSR tracking subsystem” on page 40
- “IMS-supported terminals in an RSR environment” on page 43
- “IMS type-1 commands supported from LU 6.2 devices and OTMA” on page 43
- “Terminal security defaults for LU 6.2 devices and OTMA” on page 44
- “Commands and keywords supported by the OM API” on page 45
- “Equivalent IMS type-1 and type-2 commands” on page 48

**Related reference:**

“IMS type-1 command format” on page 16

“IMS type-2 command format” on page 22

“Commands and keywords supported by the OM API” on page 45

---

## How to enter IMS commands

These topics describe how to enter various types of IMS commands.

**Subsections:**

- “IMS MFS 3270 master terminal format”
- “Using Multiple Console Support (MCS) consoles” on page 5
- “Outstanding reply numbers on z/OS consoles” on page 5
- “Maximum length of command input from z/OS consoles” on page 5
- “Multisegment command input” on page 6
- “Commands in a DBCTL environment” on page 7
- “Commands from an LU 6.2 device” on page 8
- “Qualifying network LU names and commands” on page 9
- “Issuing commands to the OM API” on page 9
- “Batch SPOC format” on page 10

**Related concepts:**

 [REXX SPOC API \(Operations and Automation\)](#)

 [Controlling IMS with the TSO SPOC application \(Operations and Automation\)](#)

## IMS MFS 3270 master terminal format

You can use the IMS master terminal for messages, commands, and responses.

The IMS master terminal consists of two components of the IBM 3270 Information Display System:

- A 3270 display screen with 24 lines by 80 columns; a screen size of 1920 characters
- A 3270 printer

All IMS system messages for the master terminal are sent to the display screen.

The format of the display screen is normally divided into five areas:

- System status area (line 1)
- Message area (lines 2-10)
- Display area (lines 12-21)
- Warning message area (line 22)
- User input area (lines 23 and 24)

The following figure shows the format of the master terminal display screen.

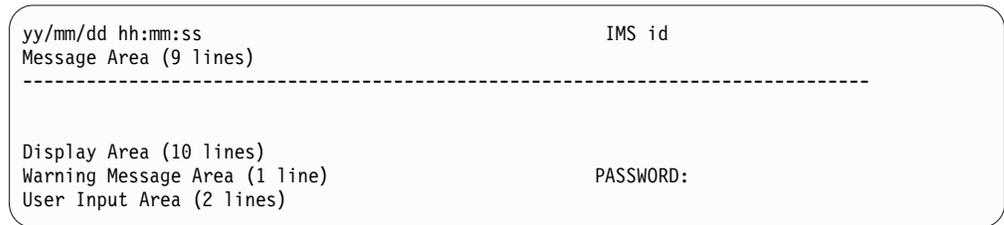


Figure 1. Master terminal format

The following figure shows the format of the master terminal display screen in an extended recovery facility (XRF) environment.

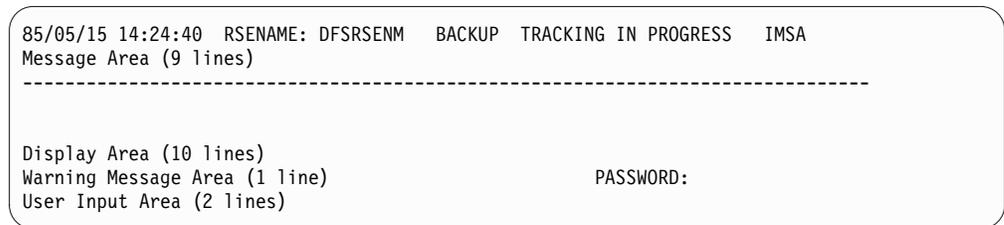


Figure 2. Master terminal format in XRF environment

## System status area

This area (line 1) displays the date, time, and IMS identifier:

**Date** The current date

**Time** The current time

### IMS Identifier

IMS system ID

If the system is XRF capable, the following information is also displayed on the status line:

### RSEname

Recoverable service element name

### System

ACTIVE or BACKUP (field is completed following the initialization phase)

**Phase** Blank or one of the following:

#### Initialization

Active or alternate system is in initialization phase

#### Synchronization

Alternate system processing SNAPQ checkpoint

**Tracking in progress**

Alternate system synchronized with active system and in the process of tracking active system

**Takeover in progress**

Alternate system in the process of taking over from active system

**Awaiting I/O prevention**

New active system is in I/O toleration mode and is awaiting completion of I/O prevention on the failing system prior to becoming a true active system

**Message area**

This area (lines 2 through 10) displays IMS system messages, messages from other terminals and programs, and responses to IMS commands, except responses to a /DISPLAY command. If the message area is full and more output needs to be displayed, a warning message is sent. Press the PA1 key to receive more output. New data is displayed on the top line followed by a blank line to separate new data from old messages.

**Display area**

This area (lines 12 through 21) displays the output from a /DISPLAY command. If the response to the command exceeds the size of the area, a warning message is sent.

**Warning message area**

This area (line 22) can contain one of the following warning messages:

**MASTER LINES WAITING**

A message being displayed in the message area has exceeded the available space and is only partially displayed.

**MASTER MESSAGE WAITING**

The message area is full, and a message from another terminal or program is waiting to be displayed.

**DISPLAY LINES WAITING**

The output of a /DISPLAY command exceeds the size of the display area.

**USER MESSAGE WAITING**

A message requiring the screen to be formatted differently is waiting to be displayed.

In all cases, press the PA1 key to receive the waiting output.

The literal PASSWORD is followed by an unprotected field with the nondisplay attribute at the end of line 22.

**User input area**

This area (lines 23 and 24) is used to enter IMS commands. It consists of two 79-character segments, so you can enter multisegment commands such as /BROADCAST, /CHANGE, /RMxxxxxx, and /SSR. For example, to send a message to all active terminals, enter /BROADCAST ACTIVE on line 23 and enter the message text on line 24. The cursor is initially positioned by IMS to line 23, your first input line.

## Using Multiple Console Support (MCS) consoles

You can enter IMS commands from Multiple Console Support (MCS) or Extended Multiple Console Support (E-MCS) consoles.

Use the CMDMCS keyword for the IMS, DBC, or DCC procedures during IMS system definition to enable commands to be entered from MCS or E-MCS consoles.

In a DB/DC environment, if Multiple Console is enabled (CMDMCS other than 'n' specified), IMS commands can be entered from an MCS or E-MCS console by prefixing the command by either the CRC or IMSID. For example, if the CRC is '#', a valid command would be #DIS A. If the IMSID is 'IMSP', a valid command would be IMSPDIS A.

Each segment of an IMS command must begin with one of the following:

- The command-recognition character for the IMS subsystem
- The IMS ID of the IMS subsystem

For multisegment commands, each segment, except the last, must end with the command recognition character or IMS ID followed by the ENTER key. The last segment must end with only the EOM (end-of-message, or ENTER) character.

You can send commands to all the IMS systems in an IMSplex by using an automated operations single-point-of-control (SPOC) application that sends commands and receives the responses to those commands through the Operations Manager (OM) application programming interface (API).

When you are using a Resource Manager (RM) structure in an IMSplex, it is best to issue commands to the Operations Manager (OM) through the TSO SPOC.

## Outstanding reply numbers on z/OS consoles

Outstanding reply numbers are used only on z/OS system consoles. They are used to pass input directly to IMS. Terminals defined to IMS do not use outstanding reply numbers.

As an example, IMS might display the following on the z/OS console:

```
*48 DFS996I *IMS READY*
```

To communicate with IMS through the z/OS system console, you must reply with the outstanding reply number (in this example, the number is 48). A reply might look like this:

```
R 48,/START DC
```

After IMS responds to the command, the next outstanding reply number is displayed with the DFS996I message, as follows:

```
DFS058I START COMMAND COMPLETED  
*49 DFS996I *IMS READY*
```

## Maximum length of command input from z/OS consoles

For single-segment commands that are entered from the z/OS console, the maximum character length (including command recognition characters) is 118.

This limit comes from the implementation of the WTOR macro.

## Multisegment command input

Multisegment commands require an EOS (end-of-segment) indicator for all segments preceding the last segment, and an EOM (end-of-message) indicator for the last segment.

The EOS and EOM will vary, depending on the type of terminal entering the command. For example, on the system console, EOS is indicated by the ENTER key and EOM is indicated by entering a period followed by the ENTER key.

The following are multisegment commands:

- /ALLOCATE
- /BROADCAST
- /CHANGE
- /ERESTART
- /LOOPTEST
- /OPNDST
- /RMxxxxxx
- /SSR

The period is used both as a delimiter for comments on IMS commands and as an EOM indicator for some terminal types. The following four examples illustrate the use of the period in both contexts.

| If support for special characters is enabled in RACF and the last parameter of the  
| /OPNDST command is a password, you must add a space before the period.

| For IMS 14 to support special characters in RACF passwords when special  
| characters is also enabled in RACF, install APAR/PTFs PI48111/UI34376/UI34377.  
| Also, ensure that all IMS systems that require support for RACF special characters  
| are migrated to IMS 14.

### *Example 1 for multisegment commands*

System console entry:

```
24,/BRO MASTER. END OF DAY MESSAGE  
25,I AM GOING HOME. PLEASE TAKE OVER.
```

Response on master terminal:

```
I AM GOING HOME. PLEASE TAKE OVER.
```

Explanation: The first input segment (reply 24) contains only the command and consists of a comment (END OF DAY MESSAGE) separated from the command itself (BRO MASTER) by a period. The second input segment (reply 25) consists of the data to be broadcast. The first period in segment 2 is part of the data and the second period in segment 2 is the EOM indicator for the system console.

### *Example 2 for multisegment commands*

System console entry:

```
26,/CHANGE DIRECT  
27,ORY MFS. MODIFICATION.
```

Explanation: MODIFICATION in segment 2 is a comment and is separated from the command by the first period. The second period in segment 2 is the EOM indicator for the system console.

The above 2 segments are combined and the following is input to the command processor:

```
/CHANGE DIRECTORY MFS. MODIFICATION
```

### *Example 3 for multisegment commands*

System console entry:

```
28,/CHANGE LINK 1  
29,2 3 FORCSESS.
```

Explanation: Certain forms of reply to the system console “outstanding reply” can compress out leading blanks from input segments. In example three, the command processed by the /CHANGE command processor would be: /CHANGE LINK 1 2 3 FORCSESS. To obtain the desired command, /CHANGE LINK 1 2 3 FORCSESS, one of the following forms of reply could be used:

```
R 29, '/CHANGE LINK 1'  
R 30, ' 2 3 FORCSESS.'
```

or

```
29,/CHANGE LINK 1  
30, 2 3 FORCSESS
```

### *Example 4 for multisegment commands*

LU2 or 3270 Display Terminal entry:

```
/FORMAT DFSM04  
/BRO LTERM WTOR (eos)  
this is segment 1 (eos)  
this is segment 2 (eos)  
this is segment 3 (eom)
```

Explanation: The remote terminal is first formatted by the /FORMAT command, where default format DFSM04 supports the input of 4 segments.

For LU2 and 3270 display terminals, enter a /FORMAT DFSM04 command before entering the multisegment command.

## **Commands in a DBCTL environment**

In a DBCTL environment, there are no outstanding reply numbers for z/OS consoles, and therefore none should be entered. For a command, just enter the command recognition character (slash or other character preceding the verb) or IMS ID, followed by the command. DBCTL does not respond with the DFS996I outstanding reply number.

### *Examples for DBCTL commands*

System console entry for single-segment command:

```
¢DIS DB PAYROLL
```

Explanation: ¢ is the command-recognition character for the DBCTL subsystem.

```
IMS1DIS DB PAYROLL
```

Explanation: IMS1 is the IMS ID for the DBCTL subsystem.

System console entry for multisegment command:

```

/RMI DBRC='IC DBD(DEDBDD01) AREA (DD01AR0) ICDSN(FVT31.DEDBDD01.DD01
AR0.IC.DUMMY1) ICDSN2/
/(FVT31.DEDBDD01.DD01AR0.IC2.DUMMY1) HSSP'

```

Explanation: The first segment of the /RMI command has the command recognition character at the beginning of the segment and at the end of the segment, to indicate another segment follows. The last segment has the command recognition character at the beginning of the segment only.

## Commands from an LU 6.2 device

To enter a command from an LU 6.2 device, allocate a conversation using an IMS command verb as the TPNAME.

The “/” must be included in the command word. The password option is not valid. (The password must be supplied as a part of the LU 6.2 ALLOCATE from the partner.) The normal LU 6.2 security provisions apply for this transaction.

**Restriction:** The APPC synchronization level of CONFIRM is ignored for commands from LU 6.2 devices.

The following three examples show the sequence of steps used to issue commands by allocating a synchronous conversation on an LU 6.2 device:

### *Example 1 for commands from an LU 6.2 device*

```

ALLOCATE
  -LUNAME=luname of IMS
  -TPNAME=/DISPLAY                (Single segment command)
SEND
  -DATA=LTERM VA01 VA02.comments (Data after period treated as comments)
PREPARE_TO_RECEIVE
RECEIVE                          (Get header)
RECEIVE                          (Get output for LTERM VA01)
RECEIVE                          (Get output for LTERM VA02)
RECEIVE                          (Get time stamp)
RC=DEALLOCATE

```

### *Example 2 for commands from an LU 6.2 device*

```

ALLOCATE
  -LUNAME=luname of IMS
  -TPNAME=/CHANGE                (Multisegment command)
SEND
  -DATA=NODE                    (Can spread over several SENDs)
SEND
  -DATA=VAT02 ASR OFF.comments (Data after period treated as comments)
PREPARE_TO_RECEIVE
RECEIVE                          (Get DFS058 CHANGE COMMAND COMPLETED)
RC=DEALLOCATE

```

### *Example 3 for commands from an LU 6.2 device*

```

ALLOCATE
  -LUNAME=luname of IMS
  -TPNAME=/BROADCAST
SEND
  -DATA=NODE VAT0*              (1st SEND: the destination)
SEND
  -DATA=HELLO, HOW ARE YOU ALL? (Subsequent SENDs: message to broadcast)
PREPARE_TO_RECEIVE
RECEIVE                          (Get DFS058 BROADCAST COMMAND COMPLETED)
RC=DEALLOCATE

```

## Qualifying network LU names and commands

Qualifying LU names gives the system administrator the liberty of using the same name for LUs on different systems by adding the network identifier. This eliminates the necessity to have unique names for every LU on every system in your complex.

A network-qualified LU name consists of a network identifier of the originating system that is 1 to 8 characters long, a period, and then the LU name, which is 1 to 8 characters long. IMS commands that include network-qualified LU names must be enclosed in single quotes. For example: 'NETID001.LUAPPC02'. No blanks are allowed in a network-qualified LU name.

The parameter ALL for either the network identifier or the LU name cannot be substituted in a command for a network-qualified LU name. ALL cannot represent all of the LU names or all of the networks.

The LU name in the LU 6.2 descriptors can be network-qualified.

The network-qualified LU name is optional on commands that support the LUNAME keyword.

## Issuing commands to the OM API

You can issue commands to the OM API and receive responses from the API by using a single point of control (SPOC) application such as the TSO SPOC, which is shipped with IMS.

### Related concepts:

 [CSL OM automated operator program clients \(System Programming APIs\)](#)

### TSO SPOC format

The TSO SPOC is an IBM-supplied application that can issue operator commands in an IMSplex.

The TSO SPOC application uses an ISPF panel interface and communicates with an Operations Manager (OM) address space. OM then communicates with all of the other address spaces in the IMSplex (for example, IMS) as required for operations. TSO SPOC can issue all type-2 commands and it has a set of panels dedicated to the task of managing resources (also known as issuing DRD-related commands).

There can be more than one TSO SPOC in an IMSplex. However, the TSO SPOC is optional in an IMSplex.

The TSO SPOC provides the following functions to an IMSplex:

- Presents a single system image for an IMSplex by allowing the user to issue commands to all IMS systems in the IMSplex from a single console.
- Displays consolidated command responses from multiple IMS address spaces.
- Sends a message to an IMS terminal connected to any IMS control region in the IMSplex by using the IMS /BROADCAST command.

There are several ways to issue commands in the IMS TSO SPOC application:

- By command line
- By retrieving a command
  - Using the ISPF RETRIEVE command

- Using a command listed in the response area
- Using the Command status panel
- By defining and using command shortcuts

You can use these methods in any combination at any time.

The following figure shows the format of the TSO SPOC screen.

```

File      Display      View      Options      Help
-----
PLEX1
Command ==> _____
----- Plex . ____ Route . ____ Wait . ____
Response for:
CSLM000I (C) Copyright IBM Corp. 2000. All rights reserved.
F1=Help   f3=Exit   F4=Showlog F6=Expand   F9=Retrieve   F12=Cancel

```

Figure 3. TSO SPOC screen format

You can issue both IMS type-1 commands and type-2 commands using the TSO SPOC interface. Enter the command next to the command prompt (Command ==> in the figure above). Enter the IMSplex name in the Plex field. Enter the list of IMS systems to which to route the command, if applicable, in the Route field. After you type the command, press Enter. The command issued is shown in the Response for: field and the actual command response is shown below the Response for: field.

### Entering comments from a TSO SPOC

For type-1 IMS commands, a period ends a command and anything after it is a comment. The following example shows a type-1 IMS command with a comment.

```
/DISPLAY ACT .a test comment
```

For type-2 commands, the comment must be enclosed in a slash and asterisk. At the end of the last parameter of the command, type a forward slash followed by an asterisk, then the comment text, followed by an asterisk and slash. The comment text must not have an embedded asterisk slash in it. The following is an example of a command with a comment:

```
QUERY TRAN NAME(PART) SHOW(QCNT) /*Show queue count for tran part*/
```

## Batch SPOC format

The Batch SPOC is a utility program that uses OM services to submit IMS operator commands to an IMSplex. The utility program is not specific to DRD commands; it accepts any commands that are supported by the OM API.

The Batch SPOC utility is invoked by use of standard JCL statements. The following example shows a simple invocation:

```
//SPOCJOB JOB ,
// MSGCLASS=H,NOTIFY=&SYSUID,USER=&SYSUID/*
//SPOC EXEC PGM=CSLUSPOC,
// PARM=(' IMSPLEX=PLEX1,ROUTE=IMS3,WAIT=30,F=WRAP')
//STEPLIB DD DISP=SHR,DSN=IMS.SDFSRESL
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
QRY IMSPLEX SHOW(JOB,TYPE, +
```

STATUS)

```
QRY TRAN NAME(INV1*) SHOW(ALL) /* inventory appl */
/*EOF
```

The program parameters define the IMSplex environment. The parameter keywords are:

**IMSPLEX**

Required parameter that specifies the 1- to 5-character suffix of the IMSplex name.

**F** Optional parameter that specifies the print format of the SPOC output. You can specify one of the following values:

**WRAP**

Wraps to the next line as needed. This is the default.

**BYCOL**

Lines of data are grouped together by the column.

**BYRSC**

Lines of data are grouped together by the resource.

**ROUTE**

Optional parameter that specifies the SYSIDs of IMSplex members that are to execute the command. If ROUTE is not specified, all members of the IMSplex will execute the command. If more than one member is specified, enclose the list in parenthesis and separate the names with commas. For example:

```
// PARM=('IMSPLEX=PLEX1,WAIT=30,ROUTE=(IMSZ,IMSA)')
```

If ROUTE=\* is specified, the command is routed to all registered command processing clients in the IMSplex. If ROUTE=% is specified, the command is routed to only one command processing client in the IMSplex that is registered for the command and that has MASTER capability. The Operations Manager chooses the command processing client.

**WAIT**

Optional parameter that specifies the wait time for individual commands. The wait value is in minutes and seconds (MMM:SS) or just seconds (SSSSS). OM will return a single response as soon as a response is received from all of the members of the IMSplex. If the interval expires, OM will return any responses from IMSplex members, plus an indication that some did not reply. The Batch SPOC utility will wait for each command to complete before issuing the next command. The default wait value is five minutes (5:00). The WAIT time applies to every command in the SYSIN file. The user can specify a wait time of zero seconds; in this case, the batch SPOC issues a command but does not wait for the response.

The SYSIN file is provided by the user and contains the commands that the user wants to run. The commands are run serially. When one command completes, the next command is run until all records from the SYSIN file are processed.

Continuation of the SYSIN control statements is specified by a plus sign (+) or a minus sign (-) as the last non-blank character of the line. A plus sign removes the leading spaces from the next line; a minus sign keeps leading spaces. Comments can be included within the SYSIN file and are specified as:

```
/* this is a comment */
```

The SYSPRINT file will have the formatted command response. If more than one command is issued, the responses will appear in the same order as the commands

appear in the SYSIN file. The default record length is 133. The command response is formatted in a similar format as the TSO SPOC display. If the records are too long, they wrap to the next line. The user may specify DCB information in the JCL or in the data set allocation to allow longer records in the SYSPRINT file.

System Display and Search Facility (SDSF) can be used to view batch job output. The following is an example of batch job output:

```
=====
Log for. . . : QRY IMSPLEX SHOW(MEMBER,TYPE,STATUS)
IMSpIplex . . . . . : PLEX1
Routing . . . . . :
Start time. . . . : 2005.132 15:36:28.11
Stop time . . . . : 2005.132 15:36:29.17
Return code . . . : 00000000
Reason code . . . : 00000000
Command master. . : SYS3
IMSpIplex MbrName  CC Member  Type  Status
CSLPLEX1 OM1      0 SYS3   DBDC  ACTIVE
CSLPLEX1 OM1      0 IMS2   DBDC  ACTIVE
=====
```

If no wait time, WAIT=0, is specified, the command response is not available and therefore are not printed. The SYSPRINT file will only have short summary information for each command. The following is an example of bath job output with no response:

```
=====
Log for. . . : QRY IMSPLEX SHOW(MEMBER,TYPE,STATUS)
IMSpIplex . . . . . : PLEX1
Routing . . . . . :
Start time. . . . : 2006.075 15:36:28.11
=====
```

**Related reference:**

-  Issuing Batch SPOC commands (Operations and Automation)
-  Batch SPOC utility (CSLUSPOC) (System Utilities)

---

## Command responses

With the exception of the /DISPLAY command, /FORMAT command, and type-2 commands returned through the OM API, responses to IMS commands are prefixed by the letters DFS*nnn*; *nnn* identifies the message.

A response to an IMS command is an acknowledgment from IMS to the originating terminal that the command has been received. Responses that go to the system console have an IMS ID that identifies the IMS system that issued the message. For example, the response to /DBDUMP DATABASE MSDB would be:

```
DFS058I (time stamp) DBDUMP COMMAND IN PROGRESS (IMS id)
```

At system definition, the TIMESTAMP/NOTIMESTP parameter of the COMM macro determines whether the time stamp is present or absent. If the time stamp feature is included, the date and time of the response appear between the response prefix and the text.

The DFS058 COMMAND COMPLETED/IN PROGRESS response indicates whether IMS accepted the command. If some parameters of the command are not accepted, the response includes the EXCEPT phrase and indicates the parameters that were not accepted. If IMS does not have the space for all of the not-accepted parameters, it truncates the EXCEPT phrase and terminates it with ...etc.. Commands that

specify the ALL parameter are most likely to be truncated. Truncated EXCEPT phrases on commands are normally caused by:

- Misspelling a parameter
- Specifying an invalid parameter for the command
- Specifying the ALL parameter for resources when some of them are already in the requested state

**Related reference:**

 XML tags returned as CSL OM responses (System Programming APIs)

## Command responses to OM

When an IMS type-2 command or IMS type-1 command is issued from the Operations Manager (OM) API, OM routes it (through the Structured Call Interface (SCI)) to one or more IMS systems based on the command routing. Each IMS returns the command response to OM. The command responses from each IMS are grouped together by OM and returned to the caller.

For each IMS command that is issued from OM API, the response is encapsulated in XML tags.

The type-2 command responses to OM also include a return code, reason code, and a possible completion code. The return or reason code returned for the command might also indicate an error from a Common Service Layer (CSL) request.

When the following type-1 commands are issued from OM API, the DFS058I COMMAND IN PROGRESS message is not returned. Instead, the messages that result as a part of command processing are returned to OM. Following is the list of commands that do not return the DFS058I COMMAND IN PROGRESS message:

- /DBDUMP DB
- /DBRECOVERY AREA
- /DBRECOVERY DB
- /DBRECOVERY DATAGRP
- /LOCK DB
- /START AREA
- /START DB
- /START DATAGRP
- /STOP ADS
- /STOP AREA
- /STOP DB
- /STOP DATAGRP
- /UNLOCK DB
- /VUNLOAD AREA

**Related concepts:**

 How to interpret CSL request return and reason codes (System Programming APIs)

**Related reference:**

 Return and reason codes for commands to OM (Messages and Codes)

## Responses to LU 6.2 devices

The response to a command from an LU 6.2 device is sent synchronously if the initiating conversation is not deallocated by the remote device. If the conversation is deallocated, the command response is sent asynchronously using the LUNAME of the device and a TPN of DFSCMD. The remote device must allocate separate DFSCMD conversations to receive each asynchronous message.

When the following commands are entered from LU 6.2 devices:

- /DBDUMP DATABASE
- /DBRECOVERY AREA
- /DBRECOVERY DATABASE
- /START AREA
- /START DATABASE
- /START REGION
- /STOP AREA
- /STOP DATABASE
- /STOP REGION

The DFS058 COMMAND COMPLETED message is generated instead of the DFS058 COMMAND IN PROGRESS message. If the LU 6.2 conversation is synchronous (the conversation still exists), the DFS058 COMMAND COMPLETED message is the last message the operator receives. If the LU 6.2 conversation is asynchronous (the conversation is deallocated), the order of delivery of the DFS058 message can vary.

---

## Sending messages to the z/OS system console

Each z/OS system console in the sysplex has a unique LTERM name.

In an XRF environment, the real z/OS system console name is the RSE name specified in the HSB member. For a non-XRF subsystem, the real system console name is the IMS ID. The generic z/OS system console name is always "WTOR". The generic name is translated to the real LTERM name when a message is sent to the z/OS system console.

Because each z/OS system console has a unique LTERM name, a message can be sent to any z/OS system console in the sysplex if the real LTERM name (not the generic name) is used.

---

## Sending messages to the IMS master terminal

Each IMS master terminal (MTO) in the sysplex has a unique LTERM name, and if they are VTAM<sup>®</sup> terminals they also have unique node names.

Each IMS assigns its primary MTO the default LTERM name "DFSPMTO" as a generic name. Each secondary MTO is assigned "DFSSMTO" as the default generic name. You can override either of these default generic names during IMS system definition by using the DFSDCxxx member of IMS.PROCLIB. The generic name is translated to the real LTERM name when a message is sent to the MTO.

**Restriction:** If you use the DFSDCxxx member to override a default generic name, you must not specify a statically-defined node name. Also, the node name you specify must have the same physical characteristics as the default node you are overriding. If you change the DFSDCxxx member, you must perform a cold start of IMS.

In a shared-queues environment, because each MTO has a unique LTERM name, a message can be sent to any MTO in the sysplex if you use the real LTERM name (not the generic name).

**Related reference:**

 DFSDCxxx member of the IMS PROCLIB data set (System Definition)

---

## IMS terminal command examples

Some command examples show the format of commands entered and received from an IMS terminal--entry terminal (ET), master terminal (MT), or response terminal (RT). One or more examples are given for each command.

The examples show:

**Entry ET:**

How the command looks as entered from the entering terminal (ET).

**Response ET:**

The IMS response transmitted to the entering terminal (ET).

**Entry MT:**

How the command looks as entered from the master terminal (MT).

Many IMS commands reference other terminals. Examples of these commands include:

**Response RT:**

The IMS response transmitted to the referenced terminal or terminals (RT).

The ET and RT responses might or might not print or display, depending on what type of terminal is referenced and whether the MSGDEL (message delete) option was specified for the terminal at system definition time.

---

## TSO SPOC command examples

Both type-2 commands and type-1 commands can be entered from a TSO SPOC. /START DB command, which is a type-1 command, and CREATE DB command, a type-2 command, are used as examples to show the input and output from a TSO SPOC.

### Type-1 command example

The input and output of /START DB command from a TSO SPOC is shown as follows:

TSO SPOC input:

```
STA DB BANKATMS BANKTERM BANKLDGR BE3ORDER
```

TSO SPOC output:

```
SYS3 DFS0488I STA COMMAND COMPLETED. DBN= BANKATMS RC=04
SYS3 DFS0488I STA COMMAND COMPLETED. DBN= BANKTERM RC=04
SYS3 DFS0488I STA COMMAND COMPLETED. DBN= BANKLDGR RC=04
```

```
SYS3   DFS0488I  STA COMMAND COMPLETED. DBN= BE3ORDER RC=08
IMS3   DFS0488I  STA COMMAND COMPLETED. DBN= BANKATMS RC=04
IMS3   DFS0488I  STA COMMAND COMPLETED. DBN= BANKTERM RC=04
IMS3   DFS0488I  STA COMMAND COMPLETED. DBN= BANKLDGR RC=04
IMS3   DFS0488I  STA COMMAND COMPLETED. DBN= BE3ORDER RC=08
```

## Type-2 command example

The input and output of CREATE DB command from a TSO SPOC is shown as follows:

TSO SPOC input:

```
CREATE DB NAME(DB000001,DB000002,DB000003) SET(ACCTYPE(UPD),RESIDENT(Y))
```

TSO SPOC output:

DBName	MbrName	CC
DB000001	IMS1	0
DB000002	IMS1	0
DB000003	IMS1	0

---

## OM API command examples

Some command examples show the input and output of commands to and from the OM API. Both type-2 commands and type-1 commands can be entered from the OM API. One or more examples are given for each command.

The examples show:

OM API input:

How the string looks as built by the AOP client.

OM API output:

The response with XML tags sent from OM.

Explanation: An explanation of the command and response.

Some type-2 commands include a section describing the environment of the IMSplex prior to entering the command.

---

## IMS type-1 command format

This section describes the general format for an IMS type-1 command.

Subsections:

- "Command recognition character" on page 17
- "Verb" on page 17
- "Keywords" on page 17
- "Parameters" on page 17
- "Null keywords" on page 20
- "Comments" on page 21
- "Passwords" on page 21

## Command recognition character

The slash (/) is the command recognition character for IMS commands. In a DBCTL environment, another character can be defined as the command recognition character and would replace the slash.

## Verb

The /VERB (such as /STOP) is the first command element and identifies the action to be taken. To abbreviate a verb, use only the slash and the first three characters. Enter the characters using either uppercase or lowercase letters.

## Keywords

IMS keywords are reserved words identifying IMS resources that relate to transaction management, databases, or the IMS system. Many commands require one or more keywords.

Table 80 on page 63 is a list of available keywords and their synonyms. The alphabetic list of keywords after Table 80 on page 63 explains the use of the keyword for each IMS command it affects.

**Restriction:** Keywords and their synonyms are reserved words. Do not use reserved words to name system resources such as transactions, programs, logical terminals, databases, and MFS block names.

Keywords are shown in uppercase. The keyword or an allowable synonym can be entered in uppercase or lowercase.

## Parameters

Parameters refer to installation defined or created resources, such as nodes, databases, or physical terminals, or IMS reserved parameters. Parameters in IMS commands must be replaced with parameter values. Reserved parameters have a unique meaning to IMS and refer to system resources or functions.

The format of the parameters that can be entered is determined by the command and keyword with which they are associated. Use the syntax example provided at the beginning of each command description to determine the valid parameter formats for that command.

When commands are entered, parameters are separated from a keyword by a space or an equal sign (=). Multiple parameters are separated by a comma, a blank, or a comma followed by a blank. The last parameter of one keyword is separated from the next keyword either by a space, a comma, or an equal sign.

In the syntax examples, default parameters appear above the syntax line. Optional parameters appear below the syntax line.

In the syntax examples, a repeat-line is shown as follows:



The repeat-line indicates that multiple parameters can be entered. To save time

while entering multiple parameters for the LINE, PTERM, NODE, and LINK keywords, inclusive parameters can be specified in some commands.

**Restriction:** Use of the ALL parameter, generic parameters, or excessive use of multiple or inclusive parameters in the large terminal network can cause some of the following problems:

- “Lock out” of the operator for the duration of the command processing
- Creation of large multisegment output messages
- Command action for only part of the terminal resources intended
- Temporary storage shortages
- Artificially high storage pools within IMS, VTAM, or both

### *Reserved words*

The resources you define, such as nodes, physical terminals, and databases, may be referenced as parameter values in IMS commands. Do not use reserved words to name these resources, or IMS might not interpret the command as expected. For example, if you define a node called TKOTRA and issue a /DISPLAY NODE TKOTRA command, IMS will list all the nodes that have an attribute of TKOTRA, instead of the specific node named TKOTRA.

### *ALL parameter*

Entering the reserved parameter ALL specifies all defined resources related to that keyword. ALL cannot be used with every command.

ALL=NO or ALL=DIS can be specified on the KEYWD macro to prevent use of the ALL parameter. ALL=NO and ALL=DIS can be specified for all keywords. ALL=NO indicates that the ALL parameter is invalid on all the commands for the keyword. ALL=DIS indicates that the ALL parameter is invalid on all the /DISPLAY commands for the keyword.

### *Inclusive parameters*

Inclusive parameters are used to specify a group of parameters by entering only the first and last parameters of the group. Inclusive parameters can be used for the following resources:

- Line
- Link
- Static node
- PTERM

To use inclusive parameters, enter the first parameter, a hyphen, then the last parameter (parameter1-parameter2).

For the LINE, PTERM, and LINK keywords, the group of parameters specified is assumed to be based on numeric sequence; that is, entering 3-6 specifies parameters 3, 4, 5, and 6. Inclusive parameters are limited to the PTERM keyword when it appears with the LINE keyword in a command. For the NODE keyword, the group of parameters must be static and based on the IMS definition sequence, not numeric or alphabetic sequence.

### *Generic parameters*

You can use generic parameters to specify a group of resources whose names match a 1- to 8-character alphanumeric mask. Generic parameters apply to both type-1 and type-2 commands.

A generic parameter can include one or more of the following wildcard characters:

- \* Matches zero or more characters. When used alone, it matches all the names for the keyword or resource.
- % Matches exactly one character.

The following examples illustrate some uses of wildcard characters:

- C0\*** Matches any string of any length that begins with C0. For instance: C0, C00, C0OP
- %S** Matches any 3-character string that ends with S. For instance: IMS, XYS
- C0\*%1** Matches any string of 5 or more characters that starts with C0, ends with 1, and has at least 2 characters between C0 and 1. For instance: CONV1, CONV21, CONV221, CONV23B1

Specifying an asterisk alone as the generic parameter (such as NAME(\*)) is equivalent to specifying ALL (such as NAME(ALL)). However, depending on the installation, other generic parameters can mean ALL. For example, the generic parameter %%% means ALL to an installation whose resource names are all 4 characters long. Some commands support NAME(\*) and do not support NAME(ALL) for a wildcard name.

When a generic parameter is used, IMS uses a serial search of all resources to match the generic name. Using a serial search of all resources can have performance implications in larger systems.

**Restriction:** For type-1 commands, if you specify ALL=NO on the KEYWD macro, you cannot use an asterisk (\*) or ALL for the keyword. However, if you specify ALL=NO on the KEYWD macro, you can use other generic parameters that are equivalent to ALL. If you are using BASIC EDIT, and you enter a command with a generic parameter containing two asterisks in a row at the end of the command, the command is discarded.

The following table lists some of the type-1 commands and keywords that support generic parameters.

*Table 1. Type-1 commands and keywords that support generic parameters*

Command	Keyword supported
/BROADCAST	LTERM, NODE, USER
/CHANGE	NODE
/CLSDST	NODE
/DISPLAY	LTERM, MSNAME, NODE, TRAN, USER
/DISPLAY ASMT	LTERM, MSNAME, NODE, USER
/DISPLAY CONV	NODE, USER
/IDLE	NODE
/OPNDST	NODE
/PSTOP	LTERM, TRAN
/PURGE	LTERM, MSNAME, TRAN

Table 1. Type-1 commands and keywords that support generic parameters (continued)

Command	Keyword supported
/RSTART	NODE, USER
/START	LTERM, MSNAME, NODE, TRAN, USER
/STOP	LTERM, MSNAME, NODE, TRAN, USER
/TRACE	NODE

**Restriction:** Generic parameters are not supported in NODE, USER, or TRAN CLASS combinations.

### Group parameters

A group parameter allows easy specification of a group of resources, where the resources in the group are defined and managed by the installation. Group parameters apply to both type-1 and type-2 command.

Group parameters are supported with the DATAGRP keyword for the /DBRECOVERY, /START, /STOP, and UPDATE (type-2) commands. A group name is one to eight characters long. It is associated with a list of databases, DEDB areas, or database data sets.

A database group name and its associated databases or DEDB areas is defined in the DBRC RECON data set using the DBRC INIT.DBDSGRP command with parameters GRPNAME and DBGRP.

A database data set (DBDS) group name and its associated DBDSs is defined in the DBRC RECON data set using the DBRC INIT.DBDSGRP command with parameters GRPNAME and MEMBERS.

- Adding resources to a group
  - Use the CHANGE.DBDSGRP command with the GRPNAME and ADDDB parameters to add databases or areas to a database group.
  - Use the CHANGE.DBDSGRP command with the GRPNAME and ADDMEM parameters to add DBDSs to a DBDS group.
- Deleting resources from a group
  - Use the CHANGE.DBDSGRP command with the GRPNAME and DELDB parameters to delete databases or areas from a database group.
  - Use the CHANGE.DBDSGRP command with the GRPNAME and DELMEM parameters to delete DBDSs from a DBDS group.
- Displaying resources in a group
  - Use the LIST.DBDSGRP command or the online /RMLIST command to display the databases or areas in a database group, or the DBDSs in a DBDS group.

Groups defined for commands should contain only databases and areas for use with the DATAGROUP keyword.

### Null keywords

Null keywords are used to help clarify the syntax of the command being entered but have no effect on the command. Null keywords can be used within IMS

commands at any time. However, they cannot be used with type-2 commands. Null keywords are reserved words. Do not use them to name system resources. The following are IMS null keywords:

- AFTER
- FOR
- ON
- SECURITY
- TO

## Comments

You can add comments or notes to the end of a command. When you add comments, identify the end of the command by entering a period after the last parameter. If you do not enter a period, residual data from a 3270 affects command processing.

## Passwords

Depending on the password security arrangement of the defined IMS system and the terminal where the command is entered, a password can be required after the command verb, or after a parameter or a keyword of certain commands.

If a command password is required, it must be entered immediately after /VERB and is normally enclosed in parentheses. No spaces or intervening characters are allowed between the /VERB and the left parenthesis.

When using MFS, if a command password is required, the installation-defined MFS format must contain a defined password field. The password must be entered into the password field defined by the format displayed. If command entry field of the format is defined to accept the command verb only, message DFS1051 is issued when an attempt is made to enter /VERB(password).

None of the IMS-supplied default formats (exception for the master terminal format) have a password field defined.

You must use RACF or an equivalent security product to require a command verb password associated with the user ID signed on to the dynamic terminal.

The /LOCK and /UNLOCK command keywords and parameters support passwords.

The /LOCK and /UNLOCK command parameters also support passwords when defined for users using RACF for static and dynamic resources. The password associated with a signed on user and specified after a command resource parameter will be used to perform a reverification check, if the resource is defined to RACF with 'REVERIFY' specified in the APPLDATA field. If the resource passes the RACF authorization check, and RVFY=Y is specified as an IMS startup parameter, IMS will verify that the password following the parameter is the same as the password entered during signon for the user that entered the command. If 'REVERIFY' is specified for a resource, but a password is not provided, or the wrong password is provided, the command processing for that resource will be rejected. This is also true for the /SET TRANSACTION and /SET CONVERSATION commands.

The use of passwords is not shown in the examples.

**Related reference:**

Chapter 5, "List of reserved words," on page 91

➡ IMS Command Language Modification facility (DFSCKWD0) (Exit Routines)

---

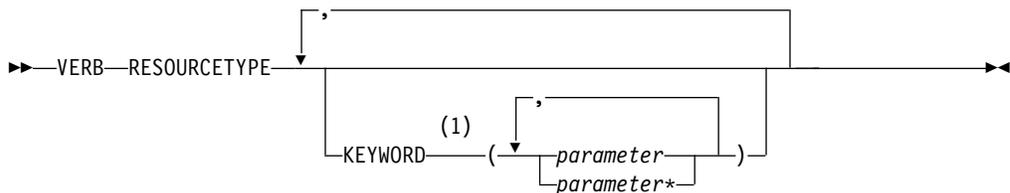
## IMS type-2 command format

IMS type-2 commands use a different format from the format used by IMS type-1 commands.

Type-2 commands can be issued only through the Operations Manager (OM) APIs by an automated operator program (AOP). An example of an AOP is the IBM-supplied TSO single point of control (SPOC) application. These commands cannot be entered from the system console, the MTO, an end-user terminal, or an IMS AOI application program. Type-2 commands are not passed to the IMS AOI user exits and are also not logged to the IMS secondary master.

Almost all commands issued through the OM API (including the IMS type-1 commands) do not require a command recognition character (for example, a slash). If one is supplied, the OM API will ignore it. The only exception to this rule is the type-1 /EXIT command, which requires a slash when entered from a TSO SPOC application.

The command format with common keywords is as follows:



**Notes:**

- 1 For some commands, KEYWORD is required, not optional.

The following is a list of the parameters with a short description of each.

**VERB** Identifies the action to be taken. The command verb can be abbreviated.

**RESOURCE TYPE**

Identifies the type of resource the command acts upon. The following is a list of the resource types:

**AREA** An area resource in an IMS.

**DATAGRP**

The members of a data group in an IMS.

**DB** The database in an IMSplex.

**DBDESC**

The databases descriptor in an IMSplex.

**IMSCON**

IMS Connect resources.

**IMS** Status and attributes of an IMS.

**IMSPLEX**

A group of IMS systems, CSL (SCI, OM, RM) members and CQs.

**LE** Language Environment® runtime parameter definitions.

**MEMBER**

An IMSplex component (For example, an IMS, OM, RM, or SCI).

**MSLINK**

Logical links.

**MSPLINK**

Physical links.

**MSNAME**

Logical link paths.

**ODBM**

Open Database Manager (ODBM), a component of the Common Service Layer (CSL).

**OLC** An online change process.

**OLREORG**

An online reorganization of a HALDB partition.

**OTMADESC**

IMS Open Transaction Manager Access (OTMA) destination descriptor.

**PGM** A program in an IMS.

**PGMDESC**

A program descriptor in an IMS.

**RM** Resource Manager (RM) address space.

**RTC** A Fast Path routing code.

**RTCDESC**

A Fast Path routing code descriptor.

**STRUCTURE**

A resource structure managed by RM.

**TRAN** An IMS resource that is a message destination.

**TRANDESC**

An IMS descriptor that is a message destination.

**KEYWORD()**

Optional command elements, depending upon the specific command.

*parameter*

Identifies an IMS-defined value or an installation-defined or created resource. Most parameters can end with a wildcard (an asterisk - \*). Parameters are listed in parentheses and must be replaced with parameter values. Multiple parameters are separated by a comma.

When a keyword and a keyword parameter are specified in the format **KEYWORD(*parameter*)**, the parameter can include embedded blanks, commas, and parentheses. However, if parentheses are embedded in the parameter, the parentheses must be paired and properly nested. For example, **KEYWORD(ABC(DEF))** is valid, but **KEYWORD(ABC(DEF)** is invalid.

If an optional keyword is specified without any parameters listed within the parentheses, the keyword is assigned a null value and is treated as if the keyword was not specified at all.

## Generic parameters

You can use generic parameters to specify a group of resources whose names match a 1- to 8-character alphanumeric mask. Generic parameters apply to both type-1 and type-2 commands.

A generic parameter can include one or more of the following wildcard characters:

- \* Matches zero or more characters. When used alone, it matches all the names for the keyword or resource.
- % Matches exactly one character.

The following examples illustrate some uses of wildcard characters:

- C0\*** Matches any string of any length that begins with C0. For instance: C0, C00, COOP
- %S** Matches any 3-character string that ends with S. For instance: IMS, XYS
- C0\*%1** Matches any string of 5 or more characters that starts with C0, ends with 1, and has at least 2 characters between C0 and 1. For instance: CONV1, CONV21, CONV221, CONV23B1

Specifying an asterisk alone as the generic parameter (such as NAME(\*)) is equivalent to specifying ALL (such as NAME(ALL)). However, depending on the installation, other generic parameters can mean ALL. For example, the generic parameter %%%% means ALL to an installation whose resource names are all 4 characters long. Some commands support NAME(\*) and do not support NAME(ALL) for a wildcard name.

When a generic parameter is used, IMS uses a serial search of all resources to match the generic name. Using a serial search of all resources can have performance implications in larger systems.

The following table lists some of the type-2 commands and keywords that support generic parameters.

*Table 2. Type-2 commands and keywords that support generic parameters*

Command	Keyword supported
DELETE	NAME with DB, DBDESC, PGM, PGMDESC, RTC, RTCDESC, TRAN, or TRANDESC
EXPORT	NAME with DB, DBDESC, PGM, PGMDESC, RTC, RTCDESC, TRAN, or TRANDESC
IMPORT	NAME with DB, DBDESC, PGM, PGMDESC, RTC, RTCDESC, TRAN, or TRANDESC
QUERY	NAME with AREA, DB, DBDESC, MSLINK, MSNAME, MSPLINK, PGM, PGMDESC, RTC, RTCDESC, TRAN, or TRANDESC
QUERY LE	LTERM, TRAN, USERID, PROGRAM
UPDATE	NAME with AREA, DB, DBDESC, MSLINK, MSNAME, MSPLINK, PGM, PGMDESC, RTC, RTCDESC, TRAN, or TRANDESC

**Restriction:** Generic parameters are not supported in a TRAN CLASS combination.

**Related concepts:**

- ➞ Issuing type-2 IMS commands (System Programming APIs)
- ➞ Type-2 command environment (System Administration)

---

## Sending commands to the IMSplex

After setting up an IMSplex, you can issue commands using the TSO single point of control (SPOC) interface.

You can also write an automated operator program that can issue the same commands. The TSO SPOC provides integrated online help.

The IMS Application Menu provides a common interface to enable you to start applications such as TSO SPOC, Syntax Checker, IVP, and more.

**Important:** SPOC applications (including the TSO SPOC and the Batch SPOC utility) issue the CSLOMCMD request. You therefore need to be aware of the parameters and return and reason codes of the CSLOMCMD request.

If you write automated programs, those programs must include logic to handle the responses from the commands; the automated programs have to parse the XML statements.

Most commands that are issued to an IMSplex are issued to OM. The exceptions are:

- BPE commands, which can be issued directly to CSL members and to CQS.
- The SHUTDOWN command, which can be issued directly to SCI to shut down one or more CSL members.
- Query requests issued by a z/OS master console to the CSL.

In an IMSplex, the format, behavior, and responses to certain IMS commands have changed. Some IMS commands are no longer recoverable; others are not supported in an IMSplex. Some commands are supported only in an IMSplex.

**Related concepts:**

- ➞ Other sample applications verified by the IVP (Installation)
- ➞ Controlling IMS with the TSO SPOC application (Operations and Automation)
  
- ➞ IMS Application Menu (System Administration)

**Related reference:**

- ➞ CSLOMCMD: command request (System Programming APIs)

---

## Command processing in an IMSplex

In an IMSplex environment, IMS commands issued through OM can behave differently than when those same commands are issued to an individual IMS subsystem.

Type-2 commands can be issued only through the OM API. Type-1 IMS commands can be issued through the OM API or to individual IMS systems through end-user terminals, master terminals, system consoles, or AOI applications. The following sections describe some of the behavioral differences.

## Command routing

Commands that are issued to OM are, by default, routed to all the IMSplex components that are active and have registered interest in processing those commands. If you want to route a command to one or more specific IMS systems in the IMSplex, use the ROUTE() parameter on the command request.

OM selects one IMSplex member (that is, IMS or RM) that is registered for the command to be the command master for each command from the OM API. The command master performs global command actions where applicable. An XRF alternate system is not a command master candidate until it takes over.

Type-2 command responses may differ depending on which IMSplex member was the command master. For example, for a QUERY TRAN NAME(traname) QCNT (GT 1) SHOW(ALL) command, only the command master returns the global queue counts, unless it does not have access to the shared queues (for example, the command master is local queues enabled).

If a command is routed to multiple IMS systems and the command fails on some of those systems as a result of environmental reasons (such as /DBR to an XRF alternate system), the overall OM return code will not reflect the environmental error.

If a command fails on all systems to which it is routed because of environmental reasons, IRC\_LIST and IRSN\_LISTNONE are produced. For type-1 commands, any error messages that are associated with the invalid environment errors will also be sent back for display. For type-2 commands, the return and reason code returned by the IMS system are from OM.

## Command behavior and the Resource Manager

Depending on whether an IMSplex is defined with a Resource Manager (and there is a resource structure available to RM), command behavior can be affected. When a resource structure is not defined, resource status must be maintained on local IMS systems in the IMSplex. In this case, commands have only a local effect.

If RM is defined with a resource structure and STM is enabled in the IMSplex, RM maintains global resource information, including resource status, for nodes, LTERMS, and users. In this scenario, resource status is maintained both globally and locally. Usually, if a user signs off or a client shuts down, resources status is maintained globally but deleted locally. If you do not want TM resources to be updated in RM, you can specify STM=NO in the PROCLIB member, DFSDCxxx.

As an alternative to STM=NO, users can choose to disable resource sharing for static ISC resources only. If this option is set, the command status for static ISC resources is always considered local as if there were no resource structure. Commands processed for a static ISC node or subpool only modify local status. Status is not updated in the resource structure. The purpose of the option is to remove the unique name restriction for static ISC-related resources so that static ISC LTERM and subpool names can be active multiple times concurrently in an IMSplex.

Another behavior that is worth noting is how command processing clients process type-1 commands (related to nodes, LTERMS, and users) that are routed to the entire IMSplex. In general, OM chooses one of the command processing clients in the IMSplex to be the “master” to coordinate the processing of the type-1 commands. Whether the master (or a non-master) IMS will process a type-1 command depends on where the command resource status is kept. If the command resource status is kept in a resource structure, the type-1 command will usually be processed by a non-master client where the command resource is active. If the command resource is not active on any of the command processing clients in the IMSplex, OM will still route the type-1 command to all clients in the IMSplex, but only the master client will process the command. If the type-1 command is being routed to all the clients in the IMSplex, command processing clients where the command resource is not active will reject the type-1 command.

**Related reference:**

- [➔ DFSDCxxx member of the IMS PROCLIB data set \(System Definition\)](#)
- [➔ Initialization exit routine \(DFSINTX0\) \(Exit Routines\)](#)

## Command characteristics

Command characteristics and operations (the way commands actually work) are affected by terminal security defaults, command recoverability, and whether commands are logged to the secondary master terminal, supported from an AOI application, mirrored on the XRF alternate, supported from the XRF alternate, and supported from LU 6.2 devices.

## Terminal security defaults for IMS type-1 commands

When neither RACF (RACF=NONE) nor the command security exit (DFSCCMD0) is used, all type-1 commands are allowed from the master terminal, except those that apply to the terminals from which the commands are issued. Only some commands are allowed from a remote terminal in this case.

The following table shows the terminal security defaults for IMS commands from the master terminal and remote terminals when neither RACF (RACF=NONE) nor the command security exit (DFSCCMD0) is used. Only the commands shown in the table are allowed in these environments.

*Table 3. Terminal security defaults for IMS type-1 commands*

Master terminal	Remote terminal
/ACTIVATE	
/ALLOCATE	
/ASSIGN	
/BROADCAST	/BROADCAST
/CANCEL	/CANCEL
/CHANGE	
/CHECKPOINT	
/CLSDST	
/COMPT	
/CQCHKPT	
/CQQUERY	

Table 3. Terminal security defaults for IMS type-1 commands (continued)

Master terminal	Remote terminal
/CQSET	
/DBDUMP	
/DBRECOVERY	
/DELETE	
/DEQUEUE	
/DIAGNOSE	/DIAGNOSE
/DISPLAY	
/END	/END
/ERESTART	
/EXCLUSIVE	/EXCLUSIVE
/EXIT	/EXIT
/FORMAT	/FORMAT
/HOLD	/HOLD
/IDLE	
/INITIATE	
/LOCK	/LOCK
/LOG	/LOG
/LOOPTEST	/LOOPTEST
/MODIFY	
/MONITOR	
/MSASSIGN	
/MSVERIFY	
/NRESTART	
/OPNDST	
/PSTOP	
/PURGE	
/QUIESCE	
/RCLSDST	/RCLSDST
	/RCOMPT
/RDISPLAY	/RDISPLAY
/RECOVER	
/RELEASE	/RELEASE
/RESET	/RESET
/RMxxxxxx	/RMLIST
/RSTART	
/RTAKEOVER	
/SECURE	
/SET	/SET
/SIGN	/SIGN
/SMCOPY	

Table 3. Terminal security defaults for IMS type-1 commands (continued)

Master terminal	Remote terminal
/SSR	
/START	
/STOP	
/SWITCH	
/TERMINATE	
/TEST	/TEST
/TRACE	
/UNLOCK	/UNLOCK
/UPDATE	
/VUNLOAD	

## Commands recovered during emergency restart

Certain commands that successfully alter IMS resources are written to the system log as X'02' or X'22' log records and are reprocessed during emergency restart.

IMS type-1 commands that are recovered during emergency restart write an X'02' log record. Type-2 commands that are recovered during emergency restart write an X'22' log record. The following table lists the commands, along with the exceptions, that are recovered during emergency restart.

Table 4. Commands recovered during emergency restart

Command	Exceptions
/ASSIGN	
/CHANGE	/CHANGE DESCRIPTOR
CREATE	
/DELETE	
DELETE	
/END	
/EXCLUSIVE	
/EXIT	
/HOLD	
/LOCK	/LOCK LTERM, /LOCK NODE, /LOCK PTERM
/LOG	
/MONITOR	
/MSASSIGN	
/PSTOP	/PSTOP LTERM
/PURGE	/PURGE APPC, /PURGE LTERM
/RELEASE	
/RSTART	
/SET	/SET LTERM, /SET TRAN
/SMCOPY	

Table 4. Commands recovered during emergency restart (continued)

Command	Exceptions
/START	/START APPC, /START ISOLOG, /START TRKARCH, /START PROG
/STOP	/STOP APPC
/TEST MFS	/TEST LINE, /TEST NODE, /TEST USER
/UNLOCK	/UNLOCK LTERM, /UNLOCK NODE, /UNLOCK PTERM, /UNLOCK SYSTEM
UPDATE DB	
UPDATE MSLINK	
UPDATE MSNAME	
UPDATE MSPLINK	
UPDATE PGM	
UPDATE RTC	
UPDATE TRAN	UPDATE TRAN START(TRACE), UPDATE TRAN STOP(TRACE)

**Restriction:** If an IMS outage (abend, modify, or cancel of IMS) occurs immediately after a command is entered, the command status might not be carried across an emergency restart or XRF takeover.

The command log records are logged asynchronously (no check write or wait write). If there is no other IMS activity that forces the log buffer to be written to the OLDS or WADS data set, the status set by the command, for restart purposes, did not occur.

There are many events in IMS where log records are check-written to the log. Any one of these events subsequent to the command causes the command log record to be written to the OLDS or WADS data set.

## Command processing in a dynamic resource definition environment

IMS processes commands differently in a dynamic resource definition (DRD) environment than in a non-DRD environment.

If you are running without DRD enabled, the control blocks that manage the resources (DDIRs, PDIRs, SMBs, and RCTEs) are loaded from the MODBLKS data set at cold start. If a type-1 command is issued to change the attribute of a resource (such as the database access type or the transaction class), the internal control blocks are updated and the changes are recovered across a warm or emergency restart. If you perform a cold start, however, the control blocks are reloaded from the MODBLKS data set, so the updated attributes revert to the original values unless you have updated your MODBLKS data set.

If you are running with DRD enabled, any changes that you make to resource definitions by using type-1 or type-2 commands persist across a warm or emergency restart. These changes also persist across a cold start if the updated resource definitions are exported to either a resource definition data set (RDDS) or the IMSRSC repository, and then imported from the RDDS or the repository during cold start. When you export the resource definitions to an RDDS or the repository, all the current attribute values are exported. If you change the value of one of the

attributes by using a type-1 or type-2 command (such as the database access type or the transaction class), the updated attribute is exported. You can either use the type-2 EXPORT DEFN command to export resource definitions explicitly, or use the automatic export function to have IMS export the definitions automatically during IMS normal checkpoints. The updated attribute values are then imported during cold start if automatic import is enabled.

**Related concepts:**

 Considerations for using dynamic resource definition (System Definition)

## IMS type-1 commands logged to the secondary master terminal

The secondary master terminal provides a hardcopy log of system activity, including checkpoint information and error messages.

If the hardcopy logging facility has been selected and not disabled (using the /SMCOPY command), certain IMS commands and responses are logged on the secondary master terminal.

You can select whether only commands issued from the master terminal are logged, only commands entered from other terminals are logged, or both are logged. Only the commands listed in the following table are logged to the secondary master terminal.

*Table 5. IMS type-1 commands logged to the secondary master terminal*

Commands	Commands (continued)
/ACTIVATE	/QUIESCE
/ALLOCATE	/RCLSDST
/ASSIGN	/RCOMPT
/CHECKPOINT	/RECOVER
/CLSDST	/RMCHANGE
/COMPT	/RMDELETE
/CQCHKPT	/RMGENJCL
/CQQUERY	/RMINIT
/CQSET	/RMLIST
/DBDUMP	/RMNOTIFY
/DBRECOVERY	/RSTART
/DELETE	/RTAKEOVER
/DEQUEUE	/SECURE
/DIAGNOSE	/START
/DISPLAY	/STOP
/IDLE	/SWITCH
/INITIATE OLREORG	/TERMINATE OLREORG
/MODIFY	/TRACE
/MONITOR	/UPDATE MSLINK
/MSASSIGN	/UPDATE OLREORG
/OPNDST	/UNLOCK SYSTEM
/PSTOP	/VUNLOAD
/PURGE	

## IMS type-1 commands supported from an AO application

Automated operator (AO) applications are application programs that issue IMS operator commands using DL/I calls.

AO applications can use two different DL/I calls to issue commands: CMD and ICMD. This section lists which IMS commands can be issued using each of these calls and describes command security for AO applications.

AO applications using the CMD and ICMD call can use RACF, DFSCCMD0, or both for security.

The TSO SPOC and REXX SPOC API are IMS applications that use OM API and cannot issue commands through DL/I calls.

## Issuing IMS type-1 commands by using the CMD call

If you are using the CMD call to issue IMS commands, transactions can be defined as AO applications with AOI= on the TRANSACT system definition macro. AO transactions are invoked in the same way any IMS transaction is invoked. AO transactions run as IMS applications with the authority to issue a planned series of IMS commands. For example, an AO application can be called in by an AO transaction after a normal restart of IMS to start IMS resources. The AO application would consist of those commands regularly used by the master terminal operator (MTO) after IMS is active. Fast Path exclusive transactions cannot be defined as AO transactions.

When RACF (or an equivalent product) is used for type-1 AOI security, the AOI= parameter on the TRANSACT macro defines which transactions can issue commands. RACF (or an equivalent product) defines which commands are allowed for each authorized transaction. These definitions in RACF give control to the authorized user who entered the transaction or to the transaction name or the command name itself. The AOI= parameter on the TRANSACT macro contains the controlling specification.

The following table shows the IMS commands that can be issued in an AO application using the CMD call. Each row includes an IMS command, restrictions, and the application environment (MPP or BMP running under DB/DC, or MPP or BMP running under DCCTL).

*Table 6. IMS type-1 commands supported from an AO application using CMD*

IMS command	Restrictions	Application environment	
		MPP or BMP running under DB/DC	MPP or BMP running under DCCTL
/ACTIVATE		Yes	Yes
/ALLOCATE		Yes	Yes
/ASSIGN		Yes	Yes
/BROADCAST		Yes	Yes
/CHANGE		Yes	Yes
/CHECKPOINT	simple form (no keywords), SNAPQ, or STATISTICS	Yes	Yes
/CLSDST		Yes	Yes
/COMPT		Yes	Yes
/CQCHKPT		Yes	Yes
/CQQUERY		Yes	Yes

Table 6. IMS type-1 commands supported from an AO application using CMD (continued)

IMS command	Restrictions	Application environment	
		MPP or BMP running under DB/DC	MPP or BMP running under DCCTL
/CQSET		Yes	Yes
/DBDUMP		Yes	Yes
/DBRECOVERY		Yes	Yes
/DELETE		Yes	Yes
/DEQUEUE		Yes	Yes
/DIAGNOSE		Yes	Yes
/DISPLAY		Yes	Yes
/END	with keywords	Yes	Yes
/EXCLUSIVE	with keywords	Yes	Yes
/EXIT	with LINE and NODE keywords	Yes	Yes
/FORMAT	with LTERM keyword	Yes	Yes
/IDLE		Yes	Yes
/INITIATE	OLREORG	Yes	No
/LOCK	without LTERM, NODE, or PTERM keywords	Yes	Yes
/LOG		Yes	Yes
/LOOPTEST		Yes	Yes
/MONITOR		Yes	Yes
/MSASSIGN		Yes	Yes
/OPNDST		Yes	Yes
/PSTOP		Yes	Yes
/PURGE		Yes	Yes
/QUIESCE		Yes	Yes
/RDISPLAY		Yes	Yes
/RECOVER		Yes	No
/RMCHANGE		Yes	Yes
/RMDELETE		Yes	Yes
/RMGENJCL		Yes	Yes
/RMINIT		Yes	Yes
/RMLIST		Yes	Yes
/RMNOTIFY		Yes	Yes
/RSTART		Yes	Yes
/SECURE		Yes	Yes
/SMCOPY		Yes	Yes
/SSR		Yes	Yes
/START		Yes	Yes

Table 6. IMS type-1 commands supported from an AO application using CMD (continued)

IMS command	Restrictions	Application environment	
		MPP or BMP running under DB/DC	MPP or BMP running under DCCTL
/STOP		Yes	Yes
/SWITCH		Yes	Yes
/TERMINATE	OLREORG	Yes	No
/TEST MFS	with LINE/NODE keywords	Yes	Yes
/TRACE		Yes	Yes
/UNLOCK	without LTERM, NODE, PTERM, or SYSTEM keywords	Yes	Yes
/VUNLOAD		Yes	No

If the 3270 message facility service (MFS) bypass option is selected, the output message specified is MODNAME=DFSEDTN, and the terminal is in preset mode, then the only command recognized by IMS is the /RESET command issued from an unformatted screen. All other commands are passed to the application program.

### Issuing IMS type-1 commands by using the ICMD call

The following table shows by application program type the commands that can be issued in an AO application using the ICMD call. This includes DRA thread running under DBCTL or DB/DC; BMP running under DBCTL; MPP, BMP, IFP running under DB/DC; and MPP, BMP, or IFP running under DCCTL. Commands that cause the IMS control region to terminate, such as /CHE FREEZE, cannot be issued.

You can secure the commands issued by an ICMD call using RACF (or the equivalent) or the Command Authorization user exit, DFSCCMD0. RACF lets you specify which commands the applications can issue. DFSCCMD0 and RACF let you do authorization checking during ICMD processing. The authorization checking can be controlled either through the transaction name, the command name, or the user ID of the signed on user. The AOI= parameter on the TRANSACT macro allows you to specify which authorization checking is to be used.

Table 7. IMS type-1 commands supported from an AO application using ICMD

IMS Command	Application environment			
	DBRA thread running under DBCTL or DB/DC	BMP running under DBCTL	MPP, BMP, or IFP running under DB/DC	MPP, BMP, or IFP running under DCCTL
/ACTIVATE	N/A	N/A	Yes	Yes
/ALLOCATE	N/A	N/A	Yes	Yes
/ASSIGN	N/A	N/A	Yes	Yes
/BROADCAST	N/A	N/A	Yes	Yes
/CANCEL	N/A	N/A	No	No

Table 7. IMS type-1 commands supported from an AO application using ICMD (continued)

IMS Command	Application environment			
	DBRA thread running under DBCTL or DB/DC	BMP running under DBCTL	MPP, BMP, or IFP running under DB/DC	MPP, BMP, or IFP running under DCCTL
/CHANGE	Yes	Yes	Yes	Yes
/CHECKPOINT (simple form)	Yes	Yes	Yes	Yes
/CHECKPOINT SNAPQ	N/A	N/A	Yes	Yes
/CHECKPOINT STATISTICS	Yes	Yes	Yes	Yes
/CLSDST	N/A	N/A	Yes	Yes
/COMPT	N/A	N/A	Yes	Yes
/CQCHKPT	N/A	N/A	Yes	Yes
/CQQUERY	N/A	N/A	Yes	Yes
/CQSET	N/A	N/A	Yes	Yes
/DBDUMP	Yes	Yes	Yes	N/A
/DBRECOVERY	Yes	Yes	Yes	N/A
/DELETE	Yes	Yes	Yes	Yes
/DEQUEUE	Yes	Yes	Yes	Yes
/DIAGNOSE	Yes	Yes	Yes	Yes
/DISPLAY	Yes	Yes	Yes	Yes
/END	N/A	N/A	Yes	Yes
/ERESTART	No	No	No	No
/EXCLUSIVE	N/A	N/A	Yes	Yes
/EXIT	N/A	N/A	Yes	Yes
/FORMAT	N/A	N/A	Yes	Yes
/HOLD	N/A	N/A	No	No
/IDLE	N/A	N/A	Yes	Yes
/INITIATE OLREORG	Yes	Yes	Yes	No
/LOCK DB	Yes	Yes	Yes	N/A
/LOCK PGM	Yes	Yes	Yes	Yes
/LOCK TRAN	N/A	N/A	Yes	Yes
/LOG	Yes	Yes	Yes	Yes
/LOOPTEST	N/A	N/A	Yes	Yes
/MODIFY	No	No	No	No
/MONITOR	N/A	N/A	Yes	Yes
/MSASSIGN	N/A	N/A	Yes	Yes
/MSVERIFY	N/A	N/A	No	No
/NRESTART	No	No	No	No
/OPNDST	N/A	N/A	Yes	Yes

Table 7. IMS type-1 commands supported from an AO application using ICMD (continued)

IMS Command	Application environment			
	DBRA thread running under DBCTL or DB/DC	BMP running under DBCTL	MPP, BMP, or IFP running under DB/DC	MPP, BMP, or IFP running under DCCTL
/PSTOP	Yes	Yes	Yes	Yes
/PURGE	N/A	N/A	Yes	Yes
/QUIESCE	N/A	N/A	Yes	Yes
/RCLSDST	N/A	N/A	No	No
/RCOMPT	N/A	N/A	No	No
/RDISPLAY	N/A	N/A	Yes	Yes
/RECOVER	Yes	Yes	Yes	No
/RELEASE	N/A	N/A	No	No
/RESET	N/A	N/A	No	No
/RMCHANGE	Yes	Yes	Yes	Yes
/RMDELETE	Yes	Yes	Yes	Yes
/RMGENJCL	Yes	Yes	Yes	Yes
/RMINIT	Yes	Yes	Yes	Yes
/RMLIST	Yes	Yes	Yes	Yes
/RMNOTIFY	Yes	Yes	Yes	Yes
/RSTART	N/A	N/A	Yes	Yes
/RTAKEOVER	No	No	No	No
/SECURE	N/A	N/A	Yes	Yes
/SET	N/A	N/A	No	No
/SIGN	N/A	N/A	No	No
/SMCOPY	N/A	N/A	Yes	Yes
/SSR	No	Yes	Yes	Yes
/START	Yes	Yes	Yes	Yes
/STOP	Yes	Yes	Yes	Yes
/SWITCH	Yes	Yes	Yes	Yes
/TERMINATE OLREORG	Yes	Yes	Yes	No
/TEST MFS LINE or NODE or USER	N/A	N/A	Yes	Yes
/TRACE	Yes	Yes	Yes	Yes
/UNLOCK DB	Yes	Yes	Yes	N/A
/UNLOCK PGM	Yes	Yes	Yes	Yes
/UNLOCK TRAN	N/A	N/A	Yes	Yes
/VUNLOAD	Yes	Yes	Yes	N/A

The following commands and keywords are not permitted:

- /CHECKPOINT keywords ABDUMP, DUMPQ, FREEZE, PURGE, and QUIESCE
- The commands /END, /EXIT, and /EXCLUSIVE if they have no keywords

- /LOCK keywords LTERM, NODE, and PTERM
- /UNLOCK keywords LTERM, NODE, PTERM, and SYSTEM

**Related concepts:**

-  Tools for automated operations (Operations and Automation)
-  Security for AO application programs (System Administration)

## Command security when using OM

For IMS command security, you have your choice of OM command security using RACF or equivalent, or IMS command security.

**Recommendation:** Use OM command security instead of IMS security.

By allowing OM to perform the security checks, commands which fail security authorization are not routed to IMS, reducing processing overhead and network traffic. When IMS command security is used, it is up to the user to ensure that all IMS systems use the same security profiles or user exits. If IMS systems in the same IMSplex use different security rules, the results of command security checking may be unpredictable.

**Related reference:**

-  IMS commands, RACF access authorities and resource names table (System Administration)

## Commands mirrored on an XRF alternate

Certain commands that successfully alter IMS resources are written to the system log as X'02' or X'22' log records.

IMS type-1 commands that alter resources write an X'02' log record. Type-2 commands that alter resources write an X'22' log record. In an XRF environment, the X'02' or X'22' log records are read and processed by the alternate system as it tracks the active system. The effect of these commands is mirrored on the alternate system when it takes over and becomes the active system. These commands, along with exceptions, are listed in the following table.

*Table 8. List of IMS commands mirrored on an XRF alternate*

Command	Exceptions
/ASSIGN	/ASSIGN is mirrored only if the assignment does not involve the master terminal.
/CHANGE	
CREATE	
/DELETE	
DELETE	
/END	
/EXCLUSIVE	
/EXIT	
/HOLD	
/LOCK	
/LOG	

Table 8. List of IMS commands mirrored on an XRF alternate (continued)

Command	Exceptions
/MONITOR	
/MSASSIGN	
/PSTOP	/PSTOP REGION
/PURGE	/PURGE APPC
/RELEASE	
/RESET	
/RSTART	
/SECURE	/SECURE APPC
/SET	
/SMCOPY	
/START	/START APPC, /START REGION
/STOP	/STOP APPC, /STOP REGION
/TEST	/TEST is usually not mirrored on an XRF alternate system. However, /TEST MFS is mirrored if the alternate system is tracking the state of the terminal that entered the /TEST MFS command.
/TRACE	/TRACE is usually not mirrored on an XRF alternate system. However, /TRACE SET LINE and /TRACE SET LINK are mirrored for lines and links that can be restarted. /TRACE SET NODE is mirrored for all node types.
/UNLOCK	/UNLOCK SYSTEM
UPDATE DB	
UPDATE DBDESC	
UPDATE MSLINK	
UPDATE MSNAME	
UPDATE MSPLINK	
UPDATE PGM	
UPDATE PGMDESC	
UPDATE RTC	
UPDATE RTCDESC	
UPDATE TRAN	UPDATE TRAN START(TRACE), UPDATE TRAN STOP(TRACE)
UPDATE TRANDESC	

## Commands supported on the XRF alternate

The commands that are valid on the XRF alternate, along with the exceptions, are shown in the following table.

Table 9. List of commands supported on the XRF alternate

Command	Exceptions
/ACTIVATE	
/CANCEL	

Table 9. List of commands supported on the XRF alternate (continued)

Command	Exceptions
/CHANGE	
/CLSDST	
/COMPT	
/DISPLAY	/DISPLAY QCNT
/END	
/ERESTART	
/FORMAT	
/IDLE	
/NRESTART	
/OPNDST	
/PSTOP REGION	
QUERY AREA	
QUERY DB	QUERY DB SHOW(WORK) QUERY DB SHOW(DEFN) QUERY DB SHOW(DEFN,IMSID) QUERY DB SHOW(DEFN,GLOBAL) QUERY DB SHOW(IMSID)
QUERY DBDESC	QUERY DBDESC SHOW(DEFN) QUERY DBDESC SHOW(DEFN,IMSID) QUERY DBDESC SHOW(DEFN,GLOBAL) QUERY DBDESC SHOW(IMSID)
QUERY IMS	
QUERY IMSPLEX	
QUERY LTERM	QUERY LTERM SHOW(GLOBAL)
QUERY MEMBER	
QUERY NODE	QUERY NODE SHOW(GLOBAL)
QUERY OTMADESC	
QUERY OTMATI	
QUERY PGM	QUERY PGM SHOW(WORK) QUERY PGM SHOW(DEFN) QUERY PGM SHOW(DEFN,IMSID) QUERY PGM SHOW(DEFN,GLOBAL) QUERY PGM SHOW(IMSID)
QUERY PGMDESC	QUERY PGMDESC SHOW(DEFN) QUERY PGMDESC SHOW(DEFN,IMSID) QUERY PGMDESC SHOW(DEFN,GLOBAL) QUERY PGMDESC SHOW(IMSID)
QUERY RTC	QUERY RTC SHOW(WORK) QUERY RTC SHOW(DEFN) QUERY RTC SHOW(DEFN,IMSID) QUERY RTC SHOW(DEFN,GLOBAL) QUERY RTC SHOW(IMSID)
QUERY RTCDESC	QUERY RTCDESC SHOW(DEFN) QUERY RTCDESC SHOW(DEFN,IMSID) QUERY RTCDESC SHOW(DEFN,GLOBAL) QUERY RTCDESC SHOW(IMSID)

Table 9. List of commands supported on the XRF alternate (continued)

Command	Exceptions
QUERY TRAN	QUERY TRAN SHOW(WORK) QUERY TRAN SHOW(DEFN) QUERY TRAN SHOW(DEFN,IMSID) QUERY TRAN SHOW(DEFN,GLOBAL) QUERY TRAN SHOW(IMSID)
QUERY TRANDESC	QUERY TRANDESC SHOW(DEFN) QUERY TRANDESC SHOW(DEFN,IMSID) QUERY TRANDESC SHOW(DEFN,GLOBAL) QUERY TRANDESC SHOW(IMSID)
QUERY USER	QUERY USER SHOW(GLOBAL)
/RCLSDST	
/RCOMPT	
/RDISPLAY	
REFRESH USEREXIT	
/RMLIST	
/SECURE	
/SMCOPY	
/START DC	
/START LUNAME	
/START REGION	
/START RTCODE	
/START SURVEILLANCE	
/STOP	
/STOP BACKUP	
/STOP DC	
/STOP LUNAME	
/STOP REGION	
/STOP RTCODE	
/STOP SURVEILLANCE	
/SWITCH	
/TEST	
/TRACE	
UPDATE MSLINK TKOTRC( )	
UPDATE TRAN SET(MAXRGN)	
UPDATE TRAN START   STOP(TRACE)	

## Commands and keywords supported on an RSR tracking subsystem

Only commands and keywords required to manage RSR tracking are supported on an RSR tracking subsystem.

For example, commands and keywords relating to transactions are not supported, because transactions are not allowed on a tracking subsystem. See Table 10 on page 41

41 for the commands and keywords supported on an IMS DB/DC tracking subsystem. The subset of commands and keywords relating to databases are the only ones allowable on a DBCTL tracking subsystem.

Commands entered on an IMS tracking subsystem that are not supported on a tracking subsystem result in message

DFS136I COMMAND xxxxxxxxxxxx INVALID FOR TRACKER

Commands entered on a DCCTL tracking subsystem that are not supported on the tracking subsystem or on DCCTL result in message

DFS136I COMMAND xxxxxxxxxxxx INVALID FOR DCCTL

Keywords entered on an IMS tracking subsystem that are not supported on a tracking subsystem result in message

DFS110 COMMAND KEYWORD yyyyyyyyyyyy INVALID FOR TRACKER

Keywords entered on a DBCTL or DCCTL tracking subsystem that are not supported on a tracking subsystem or with DBCTL or DCCTL result in message

DFS110 KEYWORD yyyyyyyyyyyy INVALID FOR DBCTL|DCCTL

The following table lists the commands and keywords that are allowed on an IMS DB/DC RSR tracking. The table also includes comments.

*Table 10. Commands and keywords allowable on an IMS DB/DC RSR tracking subsystem*

Commands	Keywords	Comments
/ACTIVATE	NODE	
/ASSIGN	COMPONENT, ICOMPONENT, LINE, LTERM, NODE, PTERM, USER	
/BROADCAST	ACTIVE, LINE, LTERM, NODE, PTERM, USER	Non-VTAM lines are not supported.
/CANCEL		
/CHANGE	CPLOG	Specifies the number of system log records between system-generated checkpoints.
/CHECKPOINT	ABDUMP, DUMPQ, FREEZE, SNAPQ, STATISTICS	Simple checkpoint (that is, with no keywords) is also supported.
/CLSDST	NODE, USER	
/COMPT	CNS, CRD, NODE, NOTRDY, PCH, PDS, PRT, RDR, READY, TDS, UDS, USER, VID, WPM1, WPM2, WPM3	
/DBRECOVERY	AREA, DATABASE, DATAGROUP, LOCAL, NOFEOV	Command only valid for a DATABASE level tracking system. Keywords only valid for areas and databases tracked at the DATABASE level.
/DEQUEUE	LINE, LTERM, NODE, PTERM, PURGE, PURGE1, USER	
/DIAGNOSE	ADDRESS, AREA, BLOCK, DB, JOBNAME, LINE, LINK, LTERM, MODULE, NODE, OPTION, PGM, REGION, SET, SHOW, SNAP, TRAN, USER	

Table 10. Commands and keywords allowable on an IMS DB/DC RSR tracking subsystem (continued)

Commands	Keywords	Comments
/DISPLAY	ACTIVE, AREA, ASSIGNMENT, DATABASE, DBD, DC, LINE, LTERM, MODE, MODIFY, MONITOR, NODE, OLDS, POOL, PTERM, SHUTDOWN STATUS, TRACE, TRACKING STATUS, USER, XTRC	<ul style="list-style-type: none"> <li>• For /DISPLAY POOL ALL, only pools needed on the tracking subsystem are displayed.</li> <li>• For /DISPLAY POOL pool, some parameters are not supported.</li> <li>• For /DISPLAY STATUS with no keywords, only resources needed on a tracking subsystem are displayed.</li> <li>• For /DISPLAY TRACE ALL, only resources needed on a tracking subsystem are displayed.</li> </ul>
/END	LINE, NODE, PTERM, USER	
/ERESTART	BUILDQ, CHECKPOINT, COLDBASE, COLDCOMM, COLDSYS, FORMAT, NOPASSWORD, NOTERMINAL, NOUSER, OVERRIDE, PASSWORD, TERMINAL, USER	
/FORMAT	LTERM	
/IDLE	LINE, NODE, NOSHUT	
/LOG		
/MODIFY	ABORT, COMMIT, LTERM, PASSWORD, PREPARE, TERMINAL	The following parameters are supported: ACBLIB, BLDL, FMTLIB, MODBLKS, and RACF.
/NRESTART	BUILDQ, CHECKPOINT, FORMAT, NOBUILDQ, NOPASSWORD, NOTERMINAL, NOUSER, PASSWORD, TERMINAL, USER	
/OPNDST	ID, LOGOND, MODE, NODE, Q, UDATA, USER, USERD	
/PSTOP	LINE, LTERM, PTERM	Non-VTAM lines are not supported.
/PURGE	LINE, LTERM, PTERM	Non-VTAM lines are not supported.
QUERY	AREA, DB, IMS, MEMBER	
/RCLSDST		
/RCOMPT	CNS, CRD, NODE, NOTRDY, PCH, PDS, PRT, RDR, READY, TDS, UDS, USER, VID, WPM1, WPM2, WPM3	
/RDISPLAY		
REFRESH USEREXIT	TYPE, MEMBER	
/RMxxxxxx	LTERM	Only applies to tracking subsystem RECON data set.
/RSTART	LINE, LOPEN, NODE, PTERM, USER	Non-VTAM lines are not supported.
/RTAKEOVER	UNPLAN, NOREVERSE	
/SIGN	ON, USERD	

Table 10. Commands and keywords allowable on an IMS DB/DC RSR tracking subsystem (continued)

Commands	Keywords	Comments
/SMCOPY	ON, TERMINAL	
/START	AREA, AUTOARCH, DATABASE, DATAGROUP, DC, ISOLOG, LINE, LOCAL, LTERM, NODE, OLDS, PTERM, SERVGRP, USER, WADS	/START AREA, /START DATABASE, and /START DATAGROUP are only valid on a DATABASE level tracking subsystem. Non-VTAM lines are not supported.
/STOP	ADS, AUTOARCH, DC, LINE, LTERM, NODE, OLDS, PTERM, SERVGRP, TRKAUTOARCH, USER, WADS	Non-VTAM lines are not supported.
/SWITCH	CHECKPOINT, OLDS	
/TEST MFS	LINE, NODE, PTERM, USER	
/TRACE	LEVEL, LINE, MODULE, NODE, ON, OPTION, SET, TABLE, UNITYPE, USER, VOLUME	No LUMI, RETR, SCHD, or SUBS TABLE tracing is allowed on a tracking subsystem.
UPDATE AREA	START(Access), STOP(Access)	
UPDATE DATAGRP	START(Access), STOP(Access)	
UPDATE DB	START(Access), STOP(Access)	

## IMS-supported terminals in an RSR environment

Since IMS-supported terminals are not supported for an RSR tracking subsystem, some type-1 commands are not supported on an RSR tracking subsystem (that is, they will cause a DFS058 COMMAND COMPLETED EXCEPT ... message to be issued).

These type-1 commands are:

- /BROADCAST LINE
- /BROADCAST PTERM
- /PSTOP LINE [PTERM]
- /PURGE LINE [PTERM]
- /RSTART LINE [PTERM]
- /START LINE [PTERM]
- /STOP LINE [PTERM]

If the parameter ALL is used with one of these commands, the IMS-supported terminals affected are simply skipped.

## IMS type-1 commands supported from LU 6.2 devices and OTMA

Only certain IMS type-1 commands are valid from LU 6.2 devices and OTMA.

The IMS type-1 commands that are supported from LU 6.2 devices and OTMA are listed in the following table:

*Table 11. IMS type-1 commands supported from LU 6.2 devices and OTMA*

Commands	Commands (continued)
/ALLOCATE*	/PURGE
/ASSIGN	/QUIESCE
/BROADCAST*	/RDISPLAY
/CHANGE*	/RMCHANGE*
/CHECKPOINT	/RMDELETE*
/CLSDST	/RMGENJCL*
/COMPT	/RMINIT*
/DBDUMP	/RMLIST*
/DBRECOVERY	/RMNOTIFY*
/DELETE	/RSTART
/DEQUEUE	/SECURE
/DIAGNOSE	/SMCOPY
/DISPLAY	/SSR
/IDLE	/START
/LOG	/STOP
/LOCK	/SWITCH
/MODIFY	/TRACE
/MONITOR	/UNLOCK
/MSASSIGN	
/MSVERIFY	
/OPNDST*	
/PSTOP	

**Note:** \* Multisegment commands are not supported by OTMA.

## Terminal security defaults for LU 6.2 devices and OTMA

Only some IMS type-1 commands from LU 6.2 devices and OTMA are authorized in an environment in which RACF is not used (RACF=NONE) and the command security exit (DFSCCMD0) is also not used.

The following table shows the terminal security defaults for IMS type-1 commands from LU 6.2 devices and OTMA when RACF is not used (RACF=NONE) and the command security exit (DFSCCMD0) is also not used. Only the commands shown in the table will be authorized in these environments.

*Table 12. Terminal security defaults for IMS type-1 commands from LU 6.2 devices and OTMA*

LU 6.2 Devices	OTMA
/BROADCAST	/LOC
/LOC	/LOG

Table 12. Terminal security defaults for IMS type-1 commands from LU 6.2 devices and OTMA (continued)

LU 6.2 Devices	OTMA
/LOG	/RDISPLAY
/RDISPLAY	
/RMLIST	

## Commands and keywords supported by the OM API

Some IMS commands are supported by the OM API.

The following table is a list of IMS command verbs and primary keywords that can be issued through the Operations Manager (OM) API. The command verb can be specified in the long form or short form, as specified in columns one and two. The primary keyword is the first keyword that follows the command verb, as specified in column three.

**Restriction:** The OM API supports only the short form of the keywords that are described in the following table. For example, the keyword ACTIVE (used on the /BROADCAST command) has a short form of ACT. The OM API supports only the ACT keyword, not ACTIVE.

Table 13. Commands supported by the OM API

Command (long form)	Command (short form)	Primary keywords (short form)
/ACTIVATE	/ACT	LINK, NODE
/ALLOCATE	/ALL	LU
/ASSIGN	/ASS	CLASS, CPRI, INPUT, LCT, LPRI, LTERM, NPRI, OUTPUT, PARLIM, PLCT, SEGNO, SEGSZ, TRAN, USER
/BROADCAST	/BRO	ACT, LINE, LTERM, MASTER, NODE, PTERM, USER
/CHANGE	/CHA	APPC, CCTL, CPLOG, DESC, DIR, FDR, LINK, NODE, PSWD, SUBSYS, SURV, TRAN, UOR, USER
/CHECKPOINT	/CHE	DUMPQ, FREEZE, PURGE, STATISTICS
/CLSDST	/CLS	NODE
/CQCHKPT	/CQC	SHAREDQ, SYSTEM
/CQQUERY	/CQQ	STATISTICS
/CQSET	/CQS	SHUTDOWN
CREATE	CRE	DB, DBDESC, IMSCON, LTERM, MSLINK, MSPLINK, MSNAME, OTMADESC, PGM, PGMDESC, RTC, RTCDESC, TRAN, TRANDESC

Table 13. Commands supported by the OM API (continued)

Command (long form)	Command (short form)	Primary keywords (short form)
/DBDUMP	/DBD	DB
/DBRECOVERY	/DBR	AREA, DB, DATAGRP
/DELETE	/DEL	DESC, PSWD, TERMINAL
DELETE	DEL	DB, DBDESC, DEFN, LE, OTMADESC, PGM, PGMDESC, RTC, RTCDESC, TRAN, TRANDESC
/DEQUEUE	/DEQ	AOITKN, LINE, LTERM, LU, MSNAME, NODE, SUSPEND, TMEM, TRAN, USER
/DISPLAY	/DIS	ACT, AFFIN, AOITKN, APPC, AREA, ASMT, CCTL, CONV, CPLOG, CQS, DB, DBD, DESC, FDR, FPV, HSB, HSSP, LINE, LINK, LTERM, LU, MASTER, MODIFY, MSNAME, NODE, OASN, OLDS, OTMA, OVERFLOWQ, PGM, POOL, PSB, PTERM, Q, QCNT, RECOVERY, RTC, SHUTDOWN, STATUS, STRUC, SUBSYS, SYSID, TIMEOVER, TMEM, TRACE, TRACKING, TRAN, UOR, USER
/END	/END	LINE, NODE, USER
/ERESTART	/ERE	BACKUP, COLDBASE, COLDCOMM, COLDSYS, NULL
/EXCLUSIVE	/EXC	LINE, NODE, USER
/EXIT	/EXI	CONV
EXPORT	EXP	DEFN
/IDLE	/IDL	LINE, LINK, NODE
IMPORT	IMP	DEFN
INITIATE	INIT	OLC, OLREORG
/LOCK	/LOC	DB, PGM, TRAN
/LOG	/LOG	
/MODIFY	/MOD	ABORT, COMMIT, PREPARE
/MONITOR	/MON	LINE
/MSASSIGN	/MSA	LINK, MSNAME, SYSID, TRAN
/NRESTART	/NRE	CHKPT, NULL
/OPNDST	/OPN	NODE
/PSTOP	/PST	LINE, LINK, LTERM, MSPLINK, REGION, TRAN

Table 13. Commands supported by the OM API (continued)

Command (long form)	Command (short form)	Primary keywords (short form)
/PURGE	/PUR	APPC, FPPROG, FPRGN, LINE, LTERM, MSNAME, TRAN
QUERY	QRY	AREA, DB, DBDESC, IMS, IMSCON, IMSPLEX, LE, LTERM, MEMBER, MSLINK, MSNAME, MSPLINK, NODE, ODBM, OLC, OLREORG, OTMADESC, OTMATI, PGM, PGMDESC, POOL, RTC, RTCDESC, STRUCTURE, TRAN, TRANDESC, USER, USEREXIT, USERID
QUEUE	QUE	LTERM, TRAN
/QUIESCE	/QUI	NODE
/RDISPLAY	/RDI	MASTER
/RECOVER	/REC	ADD, REMOVE, START, STOP, TERMINATE
REFRESH	—	USEREXIT
/RMCHANGE	/RMC	
/RMDELETE	/RMD	
/RMGENJCL	/RMG	
/RMINIT	/RMI	
/RMLIST	/RML	
/RMNOTIFY	/RMN	
/RSTART	/RST	LINE, LINK, MSPLINK, NODE, USER
/RTAKEOVER	/RTA	DUMPQ, FREEZE, UNPLAN
/SECURE	/SEC	APPC, OTMA
/SMCOPY	/SMC	MASTER, MSG, TERMINAL
/START	/STA	APPC, AREA, AUTOARCH, CLASS, DATAGRP, DB, DC, DESC, ISOLOG, LINE, LTERM, LU, MADSIOT, MSNAME, NODE, OLDS, OTMA, PGM, REGION, RTC, SB, SERVGRP, SLDSREAD, SUBSYS, SURV, THREAD, TMEM, TRAN, TRKARCH, USER, VGR, WADS, XRCTACK

Table 13. Commands supported by the OM API (continued)

Command (long form)	Command (short form)	Primary keywords (short form)
/STOP	/STO	ADS, APPC, AREA, AUTOARCH, BACKUP, CLASS, DATAGRP, DB, DC, LINE, LTERM, LU, MADSIOT, MSNAME, NODE, OLDS, OTMA, PGM, REGION, RTC, SB, SERVGRP, SLDSREAD, SUBSYS, SURV, THREAD, TMEM, TRAN, USER, VGR, WADS, XRCTACK
/SWITCH	/SWI	OLDS, SYSTEM, WADS
TERMINATE	TERM	OLC, OLREORG
/TEST	/TES	MFS
/TRACE	/TRA	SET
/UNLOCK	/UNL	DB, PGM, SYSTEM, TRAN
UPDATE	UPD	AREA, DATAGRP, DB, DBDESC, IMS, IMSCON, LE, MSLINK, MSNAME, MSPLINK, ODBM, OLREORG, OTMADESC, PGM, PGMDESC, POOL, RTC, RTCDESC, TRAN, TRANDESC
/VUNLOAD	/VUN	AREA

## Equivalent IMS type-1 and type-2 commands

Certain IMS type-1 commands and type-2 commands perform similar tasks.

The following table shows the different instances of the IMS type-1 commands and type-2 commands that perform similar tasks.

Table 14. Type-2 equivalents for the /ASSIGN command

Task	/ASSIGN command	Similar IMS type-2 command
Changes the value for the limit count of a transaction.	/ASSIGN LCT <i>new_lmct_number</i> TO TRAN <i>tranname</i>	UPDATE TRAN NAME( <i>tranname</i> ) SET(LCT( <i>new_limit_count</i> ))
Changes the value for the limit priority of a transaction.	/ASSIGN LPRI <i>new_lpri_number</i> TO TRAN <i>tranname</i>	UPDATE TRAN NAME( <i>tranname</i> ) SET(LPRI( <i>new_limit_priority</i> ))
Changes the value for the normal priority of a transaction.	/ASSIGN NPRI <i>new_npri_number</i> TO TRAN <i>tranname</i>	UPDATE TRAN NAME( <i>tranname</i> ) SET(NPRI( <i>new_normal_priority</i> ))
Changes the value for the parallel processing limit count of a transaction.	/ASSIGN PARLIM <i>new_parlim_number</i> TO TRAN <i>tranname</i>	UPDATE TRAN NAME( <i>tranname</i> ) SET(PARLIM( <i>new_parallel_limit</i> ))
Changes the value for the processing limit count of a transaction.	/ASSIGN PLCT <i>new_plmct_number</i> TO TRAN <i>tranname</i>	UPDATE TRAN NAME( <i>tranname</i> ) SET(PLCT( <i>new_processing_limit</i> ))

Table 14. Type-2 equivalents for the /ASSIGN command (continued)

Task	/ASSIGN command	Similar IMS type-2 command
Changes the limit on the number of application program output segments allowed in message queues for each GU call.	/ASSIGN SEGNO <i>new_segno_number</i> TO TRAN <i>traname</i>	UPDATE TRAN NAME( <i>traname</i> ) SET(SEGNO( <i>new_segment_number</i> ))
Creates or changes the limit on the size of application program output segments allowed in message queues for each GU call.	/ASSIGN SEGSZ <i>new_segsize_number</i> TO TRAN <i>traname</i>	UPDATE TRAN NAME( <i>traname</i> ) SET(SEGSZ( <i>new_segment_size</i> ))
Changes the class number of a transaction.	/ASSIGN TRAN <i>traname</i> TO CLS <i>new_class_number</i>	UPDATE TRAN NAME( <i>traname</i> ) SET(CLASS( <i>new_class_number</i> ))

Table 15. Type-2 equivalents for the /DBDUMP command.

Task	/DBDUMP command	Similar IMS type-2 command
Stops updates to a database.	/DBDUMP DB <i>dbname</i>	UPDATE DB NAME( <i>dbname</i> ) STOP(UPDATES) OPTION(FEOV) <sup>1</sup>

<sup>1</sup> This command does not automatically issue checkpoints unless OPTION(FEOV) is specified.

Table 16. Type-2 equivalents for the /DBRECOVERY command.

Task	/DBRECOVERY command	Similar IMS type-2 command
Stops the accessing and updating of an area.	/DBRECOVERY AREA <i>areaname</i>	UPDATE AREA NAME( <i>areaname</i> ) STOP(ACCESS)
Stops the accessing and updating of all areas and databases of the data group.	/DBRECOVERY DATAGRP <i>datagrpname</i>	UPDATE DATAGRP NAME( <i>datagrpname</i> ) STOP(ACCESS)
Stops access to the database and take it offline.	/DBRECOVERY DB <i>dbname</i>	UPDATE DB NAME( <i>dbname</i> ) STOP(ACCESS) OPTION(FEOV) <sup>1</sup>

<sup>1</sup> This command does not automatically issue checkpoints unless OPTION(FEOV) is specified.

Table 17. Type-2 equivalents for the /DISPLAY AFFIN command

Task	/DISPLAY AFFIN command	Similar IMS type-2 command
Displays TCP/IP generic resource affinities.	/DISPLAY AFFIN LINK <i>links</i>	QUERY MSLINK NAME( <i>linkname</i>   *) SHOW(AFFIN)
Displays VTAM generic resource affinities.	/DISPLAY AFFIN NODE <i>node</i>	QUERY NODE SHOW(AFFIN)

Table 18. Type-2 equivalents for the /DISPLAY AREA command

Task	/DISPLAY AREA command	Similar IMS type-2 command
Displays information about an area.	/DISPLAY AREA	QUERY AREA

Table 19. Type-2 equivalents for the /DISPLAY ASMT command.

Task	/DISPLAY ASMT command	Similar IMS type-2 command
Displays input and output components.	/DISPLAY ASMT LTERM <i>ltermname</i>	QUERY LTERM SHOW(COMPONENT)
Displays the node associated with the lterm.	/DISPLAY ASMT LTERM <i>ltermname</i>	QUERY LTERM SHOW(NODE)

Table 19. Type-2 equivalents for the /DISPLAY ASMT command (continued).

Task	/DISPLAY ASMT command	Similar IMS type-2 command
Displays the user associated with the lterm.	/DISPLAY ASMT LTERM <i>ltermname</i>	QUERY LTERM SHOW(USER)
Displays the attributes of an MSC logical link	/DISPLAY ASMT MSPLINK <i>msplinkname</i>	QUERY MSLINK SHOW(ALL)
Displays the attributes of an MSC physical link	/DISPLAY ASMT MSPLINK <i>msplinkname</i>	QUERY MSPLINK SHOW(ALL)
Displays assigned lterms.	/DISPLAY ASMT NODE <i>node</i>	QUERY NODE SHOW(LTERM)
Displays the user associated with the node.	/DISPLAY ASMT NODE <i>node</i>	QUERY NODE SHOW(USER)
Displays the other half-session qualifier ID of the ISC node.	/DISPLAY ASMT USER <i>user</i>	QUERY USER SHOW(ID)
Displays assigned lterms.	/DISPLAY ASMT USER <i>user</i>	QUERY USER SHOW(LTERM)
Displays the node.	/DISPLAY ASMT USER <i>user</i>	QUERY USER SHOW(NODE) QUERY USERID SHOW(NODE)
Displays the user ID.	/DISPLAY ASMT USER <i>user</i>	QUERY USER SHOW(USERID)
Displays the user.	/DISPLAY ASMT USER <i>user</i>	QUERY USERID SHOW(USER)

Table 20. Type-2 equivalents for the /DISPLAY CONV command

Task	/DISPLAY CONV command	Similar IMS type-2 command
Displays nodes with active or held conversations.	/DISPLAY CONV	QUERY NODE STATUS(CONV)
Displays users with active or held conversations.	/DISPLAY CONV	QUERY USER STATUS(CONV)
Displays IMS conversation information for a particular node.	/DISPLAY CONV NODE <i>node</i>	QUERY NODE SHOW(CONV)
Displays IMS conversation information for a particular user.	/DISPLAY CONV USER <i>user</i>	QUERY USER SHOW(CONV)

Table 21. Type-2 equivalents for the /DISPLAY DB command

Task	/DISPLAY DB command	Similar IMS type-2 command
Displays the status of a database.	/DISPLAY DB <i>dbname1 ...dbnamen</i>   ALL	QUERY DB

Table 22. Type-2 equivalents for the /DISPLAY DBD command

Task	/DISPLAY DBD command	Similar IMS type-2 command
Displays the programs being accessed by databases.	/DISPLAY DBD <i>dbname</i>	QUERY DB NAME( <i>dbname</i> ) SHOW(PGM)

Table 23. Type-2 equivalents for the /DISPLAY LTERM command.

Task	/DISPLAY LTERM command	Similar IMS type-2 command
Displays logical link path for remote lterms.	/DISPLAY LTERM <i>ltermname</i>	QUERY LTERM SHOW(MSNAME)
Displays the lterm message queue count.	/DISPLAY LTERM <i>ltermname</i> /DISPLAY LTERM <i>ltermname</i> QCNT	QUERY LTERM SHOW(QCNT)

Table 23. Type-2 equivalents for the /DISPLAY LTERM command (continued).

Task	/DISPLAY LTERM command	Similar IMS type-2 command
Displays status of the lterm.	/DISPLAY LTERM <i>ltermname</i>	QUERY LTERM SHOW(STATUS)
Displays queue count in the EMH queues.	/DISPLAY LTERM <i>ltermname</i> QCNT EMHQ	QUERY LTERM SHOW(EMHQ)

Table 24. Type-2 equivalents for the /DISPLAY MASTER command

Task	/DISPLAY MASTER command	Similar IMS type-2 command
Displays the primary and secondary master terminal.	/DISPLAY MASTER	QUERY LTERM STATUS(MTO,SMTO)

Table 25. Type-2 equivalents for the /DISPLAY MODIFY command.

Task	/DISPLAY MODIFY command	Similar IMS type-2 command
Displays work in progress for runtime resource definitions that would cause a DELETE, /MODIFY COMMIT, INITIATE OLC PHASE(COMMIT), or UPDATE command to change resource definitions to fail.	/DISPLAY MODIFY ALL	QUERY DB NAME( <i>dbname</i> ) SHOW(WORK)
		QUERY PGM NAME( <i>pgmname</i> ) SHOW(WORK)
		QUERY RTC NAME( <i>rtcode</i> ) SHOW(WORK)
		QUERY TRAN NAME( <i>trancode</i> ) SHOW(WORK)

Table 26. Type-2 equivalents for the /DISPLAY NODE command.

Task	/DISPLAY NODE command	Similar IMS type-2 command
Displays VTAM connection identifier (CID).	/DISPLAY NODE <i>node</i>	QUERY NODE SHOW(CID)
Displays send/receive message counts.	/DISPLAY NODE <i>node</i>	QUERY NODE SHOW(COUNT)
Displays preset destination.	/DISPLAY NODE <i>node</i>	QUERY NODE SHOW(PRESET)
Displays queue count.	/DISPLAY NODE <i>node</i>	QUERY NODE SHOW(QCNT)
	/DISPLAY NODE <i>node</i> QCNT	
Displays status for a particular node.	/DISPLAY NODE <i>node</i>	QUERY NODE SHOW(STATUS)
Displays terminal type.	/DISPLAY NODE <i>node</i>	QUERY NODE SHOW(TYPE)
Displays user ID.	/DISPLAY NODE <i>node</i>	QUERY NODE SHOW(USERID)
Displays user.	/DISPLAY NODE <i>node</i>	QUERY NODE SHOW(USER)
Displays VTAM mode table names.	/DISPLAY NODE <i>node</i> MODE	QUERY NODE SHOW(MODETBL)
Displays message queue count in the Expedited Message Handler (EMH) queues.	/DISPLAY NODE <i>node</i> QCNT EMHQ	QUERY NODE SHOW(EMHQ)
Displays owner IMSID in RM resource structure.	/DISPLAY NODE <i>node</i> RECOVERY	QUERY NODE SHOW(OWNER)
Displays status recovery information.	/DISPLAY NODE <i>node</i> RECOVERY	QUERY NODE SHOW(RECOVERY)

Table 27. Type-2 equivalents for the /DISPLAY PGM command.

Task	/DISPLAY PGM command	Similar IMS type-2 command
Displays information about a program.	/DISPLAY PGM <i>pgmname</i>	QRY PGM NAME( <i>pgmname</i> ) SHOW(ALL)
		QUERY PGM NAME( <i>pgmname</i> ) SHOW(TRAN)

Table 28. Type-2 equivalents for the /DISPLAY POOL command

Task	/DISPLAY POOL command	Similar IMS type-2 command
Displays processor storage utilization statistics for OSAM and VSAM database buffer pools.	/DISPLAY POOL DBAS	QUERY POOL TYPE(DBAS) SHOW(STATISTICS)

Table 29. Type-2 equivalents for the /DISPLAY PSB command.

Task	/DISPLAY PSB command	Similar IMS type-2 command
Displays transactions, routing codes, and databases associated with a PSB.	/DISPLAY PSB <i>psbname</i>	QUERY PGM NAME( <i>pgmname</i> ) SHOW(DB)
		QUERY PGM NAME( <i>pgmname</i> ) SHOW(RTC)
		QUERY PGM NAME( <i>pgmname</i> ) SHOW(TRAN)

Table 30. Type-2 equivalents for the /DISPLAY QCNT command

Task	/DISPLAY QCNT command	Similar IMS type-2 command
Displays lterms with messages that are older than the specified age (shared queues).	/DISPLAY QCNT LTERM MSGAGE <i>x</i>	QUERY LTERM MSGAGE( <i>x</i> )

Table 31. Type-2 equivalents for the /DISPLAY RTC command

Task	/DISPLAY RTC command	Similar IMS type-2 command
Displays information about one or more Fast Path routing codes.	/DISPLAY RTC <i>rtcname1...rtcnamen</i>   ALL	QUERY RTC NAME( <i>rtcname1...rtcnamen</i>   *) SHOW(ALL)

Table 32. Type-2 equivalents for the /DISPLAY STATUS command

Task	/DISPLAY STATUS command	Similar IMS type-2 command
Displays information about databases with the specified status.	/DISPLAY STATUS DB	QUERY DB STATUS(ALLOCF, ALLOCS,BACKOUT,EEQE, LOCK,NOTINIT,NOTOPEN, OFR,OLR,OPEN,RECALL,RECOV, RNL,STOSCHD,STOUPDS)
Displays status.	/DISPLAY STATUS LTERM	QUERY LTERM SHOW(STATUS)
Displays logical terminals with specified status.	/DISPLAY STATUS LTERM	QUERY LTERM STATUS( <i>status</i> ) SHOW(STATUS)
Displays nodes with specified status.	/DISPLAY STATUS NODE	QUERY NODE STATUS( <i>status</i> )
Displays all programs that have status and what that status is.	/DISPLAY STATUS PGM	QUERY PGM STATUS(DB-NOTAVL, IOPREV,LOCK,NOTINIT, STOSCHD,TRACE)

Table 32. Type-2 equivalents for the /DISPLAY STATUS command (continued)

Task	/DISPLAY STATUS command	Similar IMS type-2 command
Displays all Fast Path routing codes that have status and what that status is.	/DISPLAY STATUS RTC	QUERY RTC STATUS(ACTIVE,NOTINIT, NOTSCHD,STOQ)
Displays information about transactions with the specified status.	/DISPLAY STATUS TRANSACTION	QUERY TRAN NAME( <i>tranname</i> ) STATUS (IOPREV,LCK,QERR,SUSPEND, STOQ,STOSCHD,USTO)
Displays users with specified status.	/DISPLAY STATUS USER	QUERY USER STATUS( <i>status</i> )

Table 33. Type-2 equivalents for the /DISPLAY TRACE command

Task	/DISPLAY TRACE command	Similar IMS type-2 command
Displays nodes with trace status.	/DISPLAY TRACE NODE	QUERY NODE STATUS(TRACE)

Table 34. Type-2 equivalents for the /DISPLAY TRAN command

Task	/DISPLAY TRAN command	Similar IMS type-2 command
Displays information about a transaction.	/DISPLAY TRAN <i>tranname</i>	QUERY TRAN NAME( <i>tranname</i> ) SHOW(ALL)
Displays all of the transactions.	/DISPLAY TRAN ALL	QUERY TRAN SHOW(ALL)
Displays all of the transactions on the shared queues with a global queue count.	/DISPLAY TRAN <i>tranname</i> QCNT	QUERY TRAN NAME( <i>tranname</i> ) SHOW(QCNT)

Table 35. Type-2 equivalents for the /DISPLAY USER command.

Task	/DISPLAY USER command	Similar IMS type-2 command
Displays preset destination.	/DISPLAY USER <i>user</i>	QUERY USER SHOW(PRESET)
Displays node.	/DISPLAY USER <i>user</i>	QUERY USER SHOW(NODE) QUERY USERID SHOW(NODE)
Displays status for a particular user.	/DISPLAY USER <i>user</i>	QUERY USER SHOW(STATUS)
Displays user ID.	/DISPLAY USER <i>user</i>	QUERY USER SHOW(USERID)
Displays status of a particular user ID.	/DISPLAY USER <i>user</i>	QUERY USERID SHOW(STATUS)
Displays user.	/DISPLAY USER <i>user</i>	QUERY USERID SHOW(USER)
Displays autologon parameters.	/DISPLAY USER <i>user</i> AUTOLOGON	QUERY USER SHOW(AUTOLOGON)
Displays queue count.	/DISPLAY USER <i>user</i> /DISPLAY USER <i>user</i> QCNT	QUERY USER SHOW(QCNT)
Displays message queue count in the Expedited Message Handler (EMH) queues.	/DISPLAY USER <i>user</i> QCNT EMHQ	QUERY USER SHOW(EMHQ)
Displays status recovery information.	/DISPLAY USER <i>user</i> RECOVERY	QUERY USER SHOW(RECOVERY)
Displays owner IMSID in RM resource structure.	/DISPLAY USER <i>user</i> RECOVERY	QUERY USER SHOW(OWNER)

Table 36. Type-2 equivalents for the /LOCK command

Task	/LOCK command	Similar IMS type-2 command
Stops the use of a database.	/LOCK DB <i>dbname</i>	UPDATE DB NAME( <i>dbname</i> ) SET(LOCK(ON))
Locks a program.	/LOCK PGM <i>pgmname</i>	UPDATE PGM NAME( <i>pgmname</i> ) SET(LOCK(ON))
Locks a transaction.	/LOCK TRAN <i>trannname</i>	UPDATE TRAN NAME( <i>trannname</i> ) SET(LOCK(ON))

Table 37. Type-2 equivalents for the /MSASSIGN command

Task	/MSASSIGN command	Similar IMS type-2 command
Changes the transaction so that it is local and runs on the local system.	/MSASSIGN TRAN <i>trannname</i> TO LOCAL	UPDATE TRAN NAME( <i>trannname</i> ) SET(REMOTE(N))
Changes the transaction so that it is remote, and assign it to a specific logical link path.	/MSASSIGN TRAN <i>trannname</i> TO MSNAME <i>msname</i>	UPDATE TRAN NAME( <i>trannname</i> ) SET(MSNAME( <i>name</i> ))

Table 38. Type-2 equivalents for the /PSTOP command

Task	/PSTOP command	Similar IMS type-2 command
Stops the scheduling of transactions.	/PSTOP TRAN <i>trannname</i>	UPDATE TRAN( <i>trannname</i> ) START(Q) STOP(SCHD)
Stops logons to the physical link (only for MSC VTAM links).	/PSTOP MSPLINK <i>mslinkname</i>   ALL	UPDATE MSPLINK NAME( <i>mslinkname</i>   *) STOP(LOGON)
Posts the regions in which the program is scheduled so that a new refreshed copy of the program can be obtained on the next schedule of the program.	/PSTOP REGION TRAN <i>trannname</i>	UPDATE PGM START(REFRESH)

Table 39. Type-2 equivalents for the /PURGE command

Task	/PURGE command	Similar IMS type-2 command
Stops input messages for a particular transaction code.	/PURGE TRAN <i>trannname</i>	UPDATE TRAN NAME( <i>trannname</i> ) START(SCHD) STOP(Q)

Table 40. Type-2 equivalents for the /RDISPLAY command

Task	/RDISPLAY command	Similar IMS type-2 command
Displays the primary and secondary master terminal.	/RDISPLAY MASTER	QUERY LTERM STATUS(MTO,SMTO)

Table 41. Type-2 equivalents for the /RSTART command

Task	/RSTART command	Similar IMS type-2 command
Resets MSC TCP/IP or VTAM links to enable logons.	/RSTART MSPLINK <i>mslinkname</i>   ALL	UPDATE MSPLINK NAME( <i>mslinkname</i>   *) START(LOGON)

Table 42. Type-2 equivalents for the /START AREA command

Task	/START AREA command	Similar IMS type-2 command
Starts the area.	/START AREA <i>areaname</i>	UPDATE AREA NAME( <i>areaname</i> ) START(ACCESS)

Table 43. Type-2 equivalents for the /START DATAGRP command

Task	/START DATAGRP command	Similar IMS type-2 command
Starts the data group.	/START DATAGRP <i>datagrpname</i>	UPDATE DATAGRP NAME( <i>datagrpname</i> ) START(ACCESS)

Table 44. Type-2 equivalents for the /START DB command

Task	/START DB command	Similar IMS type-2 command
Starts a database and change access intent of the database.	/START DB ACCESS	UPDATE DB START(ACCESS) SET(ACCTYPE())
Starts a database.	/START DB <i>dbname</i>	UPDATE DB NAME( <i>dbname</i> ) START(ACCESS)

Table 45. Type-2 equivalents for the /START PGM command

Task	/START PGM command	Similar IMS type-2 command
Starts program scheduling.	/START PGM <i>pgmname</i>	UPDATE PGM NAME( <i>pgmname</i> ) START(SCHD)

Table 46. Type-2 equivalents for the /START RTC command

Task	/START RTC command	Similar IMS type-2 command
Starts queuing to a Fast Path routing code.	/START RTC <i>rtcname</i>	UPDATE RTC NAME( <i>rtcname</i> ) START(Q)

Table 47. Type-2 equivalents for the /START TRAN command

Task	/START TRAN command	Similar IMS type-2 command
Starts a transaction.	/START TRAN	UPDATE TRAN NAME( <i>trannname</i> ) START(Q,SCHD,SUSPEND)

Table 48. Type-2 equivalents for the /STOP AREA command

Task	/STOP AREA command	Similar IMS type-2 command
Stops an area.	/STOP AREA <i>areaname</i>	UPDATE AREA NAME( <i>areaname</i> ) STOP(SCHD)

Table 49. Type-2 equivalents for the /STOP DATAGRP command

Task	/STOP DATAGRP command	Similar IMS type-2 command
Stops a data group.	/STOP DATAGRP <i>datagrpname</i>	UPDATE DATAGRP NAME( <i>datagrpname</i> ) STOP(SCHD)

Table 50. Type-2 equivalents for the /STOP DB command

Task	/STOP DB command	Similar IMS type-2 command
Stops a database.	/STOP DB <i>dbname</i>	UPDATE DB NAME( <i>dbname</i> ) STOP(SCHD)

Table 51. Type-2 equivalents for the /STOP PGM command

Task	/STOP PGM command	Similar IMS type-2 command
Stops program scheduling.	/STOP PGM <i>pgmname</i>	UPDATE PGM NAME( <i>pgmname</i> ) STOP(SCHD)

Table 52. Type-2 equivalents for the /STOP RTC command

Task	/STOP RTC command	Similar IMS type-2 command
Stops queuing to a Fast Path routing code.	/STOP RTC <i>rtcname</i>	UPDATE RTC NAME( <i>rtcname</i> ) STOP(Q)

Table 53. Type-2 equivalents for the /STOP TRAN command

Task	/STOP TRAN command	Similar IMS type-2 command
Stops the queuing and scheduling of messages destined for a transaction.	/STOP TRAN <i>tranname</i>	UPDATE TRAN NAME( <i>tranname</i> ) STOP(Q,SCHD)

Table 54. Type-2 equivalents for the /TRACE PGM command

Task	/TRACE PGM command	Similar IMS type-2 command
Starts the tracing of a program.	/TRACE SET ON PGM <i>pgmname</i>	UPDATE PGM NAME( <i>pgmname</i> ) START(TRACE)
Stops the tracing of a program.	/TRACE SET OFF PGM <i>pgmname</i>	UPDATE PGM NAME( <i>pgmname</i> ) STOP(TRACE)

Table 55. Type-2 equivalents for the /TRACE TRAN command

Task	/TRACE TRAN command	Similar IMS type-2 command
Starts the tracing of a transaction.	/TRACE SET ON TRAN <i>tranname</i>	UPDATE TRAN NAME( <i>tranname</i> ) START(TRACE)
Stops the tracing of a transaction.	/TRACE SET OFF TRAN <i>tranname</i>	UPDATE TRAN NAME( <i>tranname</i> ) STOP(TRACE)

Table 56. Type-2 equivalents for the /UNLOCK DB command

Task	/UNLOCK DB command	Similar IMS type-2 command
Unlocks a database.	/UNLOCK DB <i>dbname</i>	UPDATE DB NAME( <i>dbname</i> ) SET(LOCK(OFF))

Table 57. Type-2 equivalents for the /UNLOCK PGM command

Task	/UNLOCK PGM command	Similar IMS type-2 command
Unlocks a program.	/UNLOCK PGM <i>pgmname</i>	UPDATE PGM NAME( <i>pgmname</i> ) SET(LOCK(OFF))

Table 58. Type-2 equivalents for the /UNLOCK TRAN command

Task	/UNLOCK TRAN command	Similar IMS type-2 command
Unlocks a transaction.	/UNLOCK TRAN <i>tranname</i>	UPDATE TRAN NAME( <i>tranname</i> ) SET(LOCK(OFF))

**Related concepts:**

 [Commands for IMS operations tasks \(Operations and Automation\)](#)

## 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 59. 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   <i>show_parm</i> )	VIEWIA ALL	QUERY ALIAS NAME(*)
QUERY IMSCON TYPE(ALIAS) NAME( <i>alias_name</i> ) SHOW(ALL   <i>show_parm</i> )	VIEWIA <i>alias_name</i>	QUERY ALIAS NAME( <i>aliasName</i> )
QUERY IMSCON TYPE(ALIAS) NAME( <i>alias_name</i> ) ODBM( <i>odbm_name</i> )	VIEWIA <i>alias_name</i> <i>odbm_name</i>	QUERY ALIAS NAME( <i>aliasName</i> ) ODBM( <i>odbmName</i> )

Table 60. 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   <i>show_parm</i> )	VIEWHWS	QUERY MEMBER TYPE(IMSCON) SHOW(ALL)

Table 61. 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   <i>show_parm</i> )	VIEWDS ALL	QUERY DATASTORE NAME(*) SHOW(ALL)
QUERY IMSCON TYPE(DATASTORE) NAME( <i>datastore_name</i> ) SHOW(ALL   <i>show_parm</i> )	VIEWDS <i>datastore_name</i>	QUERY DATASTORE NAME( <i>datastore_name</i> )

Table 62. 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   <i>show_parm</i> )	VIEWIP ALL	QUERY IMSPLEX NAME(*) SHOW(ALL)
QUERY IMSCON TYPE(IMSPLEX) NAME( <i>IMSplex_name</i> ) SHOW(ALL   <i>show_parm</i> )	VIEWIP <i>IMSplex_name</i>	QUERY IMSPLEX NAME( <i>imsplexName</i> ) SHOW(ALL)

Table 63. 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   <i>show_parm</i> )	VIEWMSC ALL	QUERY MSC NAME(*)
QUERY IMSCON TYPE(MSC) NAME( <i>msc_id</i> ) SHOW(ALL   <i>show_parm</i> )	VIEWMSC <i>msc_id</i>	QUERY MSC NAME( <i>msc_id</i> )

Table 64. 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   <i>show_parm</i> )	VIEWPORT ALL	QUERY PORT NAME(*) SHOW(ALL)
QUERY IMSCON TYPE(PORT) NAME( <i>portid</i> ) SHOW(ALL   <i>show_parm</i> )	VIEWPORT <i>port_id</i>	QUERY PORT NAME( <i>port_id</i> ) SHOW(ALL)
QUERY IMSCON TYPE(PORT) NAME(LOCAL) SHOW(ALL   <i>show_parm</i> )	VIEWPORT LOCAL	QUERY PORT NAME(LOCAL) SHOW(ALL)

Table 65. 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   <i>show_parm</i> )	VIEWRMT ALL	QUERY RMTIMSCON NAME(*)
QUERY IMSCON TYPE(RMTIMSCON) NAME( <i>rmtimscon_name</i> ) SHOW(ALL   <i>show_parm</i> )	VIEWRMT <i>rmtimscon_name</i>	QUERY RMTIMSCON NAME( <i>rmtimscon_name</i> )

Table 66. 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   <i>show_parm</i> )	VIEWUOR ALL	QUERY UOR NAME(*) SHOW(ALL)
QUERY IMSCON TYPE(UOR) NAME( <i>uor_id</i> ) SHOW(ALL   <i>show_parm</i> )	VIEWUOR <i>uor_id</i>	QUERY UOR NAME( <i>uor_id</i> ) SHOW(ALL)
QUERY IMSCON TYPE(UOR) STATE( <i>state</i> )	None	None

Table 67. 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
UPDATE IMSCON TYPE(ALIAS) NAME( <i>alias_name</i> ) ODBM( <i>odbm_name</i> ) START(COMM)	STARTIA <i>alias_name</i> <i>odbm_name</i>	UPDATE ALIAS NAME( <i>aliasName</i> ) ODBM( <i>odbmName</i> ) START(ROUTE)
UPDATE IMSCON TYPE(ALIAS) NAME( <i>alias_name</i> ) ODBM( <i>odbm_name</i> ) STOP(COMM)	STOPIA <i>alias_name</i> <i>odbm_name</i>	UPDATE ALIAS NAME( <i>aliasName</i> ) ODBM( <i>odbmName</i> ) STOP(ROUTE)

Table 68. 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( <i>client_name</i> ) PORT( <i>portid</i> ) STOP(COMM)	STOPCLNT <i>portid clientid</i>	DELETE PORT NAME( <i>portName</i> ) CLIENT( <i>clientName</i> )

Table 69. 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 69. 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(RECORDE)	RECORDER OPEN	UPDATE MEMBER TYPE(IMSCON) START(TRACE)
UPDATE IMSCON TYPE(CONFIG) STOP(RECORDE)	RECORDER CLOSE	UPDATE MEMBER TYPE(IMSCON) STOP(TRACE)

Table 70. 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( <i>converter_name</i> ) OPTION(REFRESH)	REFRESH CONVERTER NAME( <i>converter_name</i> )	UPDATE CONVERTER NAME( <i>converter_name</i> ) OPTION(REFRESH)

Table 71. 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( <i>datastore_name</i> ) START(COMM)	OPENDS <i>datastore_id</i> STARTDS <i>datastore_id</i>	UPDATE DATASTORE NAME( <i>datastoreName</i> ) START(COMM)
UPDATE IMSCON TYPE(DATASTORE) NAME( <i>datastore_name</i> ) STOP(COMM)	STOPDS <i>datastore_id</i>	UPDATE DATASTORE NAME( <i>datastoreName</i> ) STOP(COMM)
UPDATE IMSCON TYPE(DATASTORE) NAME( <i>datastore_name</i> ) SET()	None	None

Table 72. 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( <i>imsplex_name</i> ) START(COMM)	OPENIP <i>imsplex_id</i> STARTIP <i>imsplex_id</i>	UPDATE IMSPLEX NAME( <i>imsplex_name</i> ) START(COMM)
UPDATE IMSCON TYPE(IMSPLEX) NAME( <i>imsplex_name</i> ) STOP(COMM)	STOPIP <i>imsplex_id</i>	UPDATE IMSPLEX NAME( <i>imsplex_name</i> ) STOP(COMM)

Table 73. 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( <i>logical_link_name</i> ) STOP(COMM)	STOPLINK <i>logical_link_name</i>	DELETE LINK NAME( <i>linkName</i> )
UPDATE IMSCON TYPE(LINK) NAME( <i>logical_link_name</i> ) MSC( <i>lclplk_id</i> ) STOP(COMM)	STOPLINK <i>logical_link_name lclplk_id</i>	DELETE LINK NAME( <i>linkname</i> ) LCLPLKID( <i>lclPlkid</i> )

Table 74. 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( <i>lclPlkid</i> ) START(COMM)	STARTMSC <i>lclPlkid</i>	UPDATE MSC NAME( <i>lclPlkid</i> ) START(COMM)
UPDATE IMSCON TYPE(MSC) NAME( <i>lclPlkid</i> ) STOP(COMM)	STOPMSC <i>lclPlkid</i>	UPDATE MSC NAME( <i>lclPlkid</i> ) STOP(COMM)

Table 75. 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
UPDATE IMSCON TYPE(ODBM) NAME( <i>odbm_name</i> ) START(COMM)	STARTOD <i>odbm_name</i>	UPDATE ODBM NAME( <i>odbmName</i> ) START(COMM)
UPDATE IMSCON TYPE(ODBM) NAME( <i>odbm_name</i> ) STOP(COMM)	STOPOD <i>odbm_name</i>	UPDATE ODBM NAME( <i>odbmName</i> ) STOP(COMM)

Table 76. 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
UPDATE IMSCON TYPE(PORT) NAME( <i>port_name</i> ) START(COMM)	OPENPORT <i>port_id</i> STARTPT <i>port_id</i>	UPDATE PORT NAME( <i>port_name</i> ) START(COMM)

Table 76. 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
UPDATE IMSCON TYPE(PORT) NAME( <i>port_name</i> ) STOP(COMM)	STOPPORT <i>port_id</i>	UPDATE PORT NAME( <i>port_name</i> ) STOP(COMM)

Table 77. 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( <i>userid</i> ) OPTION(REFRESH)	REFRESH RACFUID NAME( <i>userid</i> )	UPDATE RACFUID NAME( <i>userid</i> ) OPTION(REFRESH)

Table 78. 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( <i>rmtimscon_name</i> ) START(COMM)	STARTRMT <i>rmtimsconName</i>	UPDATE RMTIMSCON NAME( <i>rmtimsconName</i> ) START(COMM)
UPDATE IMSCON TYPE(RMTIMSCON) NAME( <i>rmtimscon_name</i> ) STOP(COMM)	STOPRMT <i>rmtimsconName</i>	UPDATE RMTIMSCON NAME( <i>rmtimsconName</i> ) STOP(COMM)

Table 79. 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( <i>sendclient_name</i> ) RMTIMSCON( <i>rmtimscon_name</i> ) STOP(COMM)	STOPSCLN <i>rmtimscon sendclient</i>	DELETE RMTIMSCON NAME( <i>rmtimsconname</i> ) SENDCLNT( <i>clientid</i> )
UPDATE IMSCON TYPE(SENDCLNT) NAME( <i>sendclient_name</i> ) RMTICICS( <i>rmtcics_name</i> ) STOP(COMM)	STOPSCLN <i>rmtcics sendclient</i>	No equivalent z/OS command

## Chapter 2. Command keywords and their synonyms

All IMS command keywords and their synonyms, as well as keywords supported by the OM API, are listed. The list covers command keywords for the OM command processing clients such as IMS, ODBM, IMSCON, and RM.

IMS ignores misspelled keywords, and issues an error message when it encounters an invalid keyword. In an IMSplex, invalid or misspelled keywords will result in an UNKNOWN POSITIONAL PARAMETER message.

You can modify type-1 command keywords by using the IMS Command Language Modification Facility (DFSCKWD0) with the KEYWD macro.

You can specify ALL=NO or ALL=DIS on the KEYWD macro to prevent the use of the ALL parameter.

- You can specify ALL=NO for all the type-1 command keywords that support the ALL parameter. ALL=NO indicates that the ALL parameter is invalid on all the IMS commands that apply to the keyword being changed.
- You can specify ALL=DIS for all the type-1 /DISPLAY command keywords that support the ALL parameter. ALL=DIS indicates that the ALL parameter is invalid on all the /DISPLAY commands that apply to the keyword being changed.

You can prevent the use of the NAME(\*) parameter on type-2 commands by using the CSL OM input user exit routine.

In the following table, the IMS keyword column is for the OM command processing clients such as IMS, ODBM, IMSCON, and RM. A keyword listed in the OM keyword column indicates that it is the only accepted form supported by the OM API. If no keyword is listed in the OM keyword column, then the IMS keyword or its synonym is acceptable. If no synonyms are listed, none are permitted.

Table 80. Command keywords and their synonyms

IMS keyword	Synonym	OM keyword	Commands that support this keyword
ABDUMP			/CHECKPOINT /STOP /SWITCH
ABORT			/MODIFY
ACBLIB			INITIATE OLC
ACCESS			/START
ACTIVE	A, ACT	ACT	/BROADCAST /DISPLAY /SWITCH
ADAPTER			QUERY OTMADESC
ADD			/RECOVER
ADDRESS			/DIAGNOSE
ADS			/STOP

Table 80. Command keywords and their synonyms (continued)

IMS keyword	Synonym	OM keyword	Commands that support this keyword
AFFINITY	AFFIN, AFF	AFFIN	/DISPLAY /START
ALIAS			QUERY ODBM UPDATE ODBM
ALLENTRIES	ALLENT		/RECOVER
AOITOKEN	AOITKN	AOITKN	/DEQUEUE /DISPLAY /PSTOP
APDB			/TRACE MONITOR
APMQ			/TRACE MONITOR
APPC			/DISPLAY /PURGE /SECURE /START /STOP
AREA			/DBRECOVERY /DIAGNOSE /DISPLAY /RECOVER /START /STOP /VUNLOAD
ASR			QUERY MSLINK
ASSIGNMENT	ASMT	ASMT	/DISPLAY
ATTRIB			QUERY MEMBER
AUTO			/TRACE
AUTOARCH	ARCHIVE, AUTOARC	AUTOARCH	/START /STOP
AUTOLOGON	AUTLGN		/DISPLAY
BACKUP	BU	BACKUP	/ERESTART /STOP /SWITCH
BALGRP	BALG		/DISPLAY
BANDWIDTH			QUERY MSLINK
BKERR			/DISPLAY
BLDL			INITIATE OLC
BLOCK			/DIAGNOSE
BUILDQ	BLDQ, BLDQS, BUILDQS		/ERESTART /NRESTART
CAGROUP	CAGRP		/RECOVER
CANCEL			/STOP
CCTL	CC	CCTL	/DISPLAY
CHECKPOINT	CHECKPT, CHKPOINT, CHKPT	CHKPT	/ERESTART /NRESTART /SWITCH

Table 80. Command keywords and their synonyms (continued)

IMS keyword	Synonym	OM keyword	Commands that support this keyword
CLASS	CLS	CLASS	/ASSIGN CREATE TRAN CREATE TRANDESC /DISPLAY /PSTOP /PURGE QUERY TRAN /START /STOP UPDATE TRAN
CMDAUTH			/ERESTART /NRESTART
CMDAUTHE			/ERESTART /NRESTART
CMTMODE			CREATE TRAN CREATE TRANDESC QUERY OTMATI
CNS			/COMPT /RCOMPT
COLDBASE	COLDB	COLDBASE	/ERESTART
COLDCOMM	COLDC	COLDCOMM	/ERESTART
COLDSESS			/CHANGE NODE UPDATE MSLINK
COLDSYS	COLDS	COLDSYS	/ERESTART
COMMIT			/MODIFY
COMP			/TRACE
COMPONENT	COMPT		/ASSIGN
CONTINUOUS	CONT		/RSTART
CONVERSATION	CONV	CONV	CREATE TRAN CREATE TRANDESC /DISPLAY /EXIT /RELEASE /SET
CONVRTR			QUERY OTMADESC
CPLOG			/CHANGE /DISPLAY
CPRI			/ASSIGN
CQS			/DISPLAY
CRD			/COMPT
DATA			QUEUE LTERM QUEUE TRAN

Table 80. Command keywords and their synonyms (continued)

IMS keyword	Synonym	OM keyword	Commands that support this keyword
DATABASE	DATABASES, DB, DBS	DB	/DBDUMP /DBRECOVERY /DIAGNOSE /DISPLAY /LOCK /RECOVER /START /STOP /UNLOCK
DATAGROUP	DATAGRP	DATAGRP	/DBRECOVERY /START /STOP UPDATE
DATASTORE			QUERY ODBM UPDATE ODBM
DBALLOC			/START
DBD			/DISPLAY
DBDS			/RECOVER
DBDSGRP			/RECOVER
DBS			/DISPLAY MODIFY
DC			/DISPLAY /START /STOP
DCLWA			CREATE TRAN CREATE TRANDESC
DEFAULT			CREATE DBDESC CREATE PGMDESC CREATE TRANDESC QUERY DBDESC QUERY PGMDESC QUERY RTCDESC QUERY TRANDESC UPDATE DBDESC
DEFN			DELETE EXPORT IMPORT
DESCRIPTOR	DESC, L62DESC	DESC	/CHANGE /DELETE /DISPLAY /START /STOP
DIRECTORY	DIR	DIR	/CHANGE
DIRROUTE			CREATE TRAN CREATE TRANDESC
DOPT			CREATE PGM CREATE PGMDESC
DUMPQ	DUMPQS	DUMPQ	/CHECKPOINT /RTAKEOVER
EDITRTN			CREATE TRAN CREATE TRANDESC

Table 80. Command keywords and their synonyms (continued)

IMS keyword	Synonym	OM keyword	Commands that support this keyword
EDITUC			CREATE TRAN CREATE TRANDESC
EMHBSZ			CREATE TRAN CREATE TRANDESC
EMHQ			/DISPLAY
ENDTIME			EXPORT
ERRORABORT			/RECOVER
ERRORCONT			/RECOVER
EXIT			/DISPLAY /TRACE
FDR			/CHANGE /DISPLAY
FMTLIB			INITIATE OLC
FORCE			/CLSDST /PSTOP /SWITCH
FORCSESS	FORC		/CHANGE
FORMAT	FMT		/ERESTART /NRESTART
FP			CREATE PGM CREATE PGMDESC CREATE TRAN CREATE TRANDESC
FPPROG			/PURGE
FPREGION	FPRGN	FPRGN	/PURGE
FPVIRTUAL	FPV	FPV	/DISPLAY
FRCABND			INITIATE OLC
FRCNRML			INITIATE OLC
FREEZE			/CHECKPOINT /RTAKEOVER
GLOBAL			/DBDUMP /DBRECOVERY QUERY TRAN /START /STOP
GPSB			CREATE PGM CREATE PGMDESC
GRPNAME			QUERY OTMATI
GRSNAME	GRSN, GRS		/START
HSB	HOTSTANDBY	HSB	/DISPLAY
HSSP			/DISPLAY
ICOMPONENT	ICOMPT		/ASSIGN
ID			/CHANGE /OPNDST

Table 80. Command keywords and their synonyms (continued)

IMS keyword	Synonym	OM keyword	Commands that support this keyword
IMSCON		IMSCON	CREATE QUERY UPDATE
INDOUBT	IND		/DISPLAY
INPUT			/ASSIGN /DISPLAY /START /STOP /TRACE
INQ			CREATE RTC CREATE RTCDESC CREATE TRAN CREATE TRANDESC
INTERVAL	INT, INTV		/CHANGE /TRACE MONITOR
ISOLOG			/START
JOBNAME	JBN, JOBN		/PSTOP /START /STOP
LA			/TRACE MONITOR
LANG			CREATE PGM CREATE PGMDESC
LCT			CREATE TRAN CREATE TRANDESC
LE			DELETE UPDATE QUERY
LEAVEGR			/CHECKPOINT
LEAVEPLEX			/CHECKPOINT
LEVEL			/TRACE
LIBRARY			QUERY OLC
LIKE			CREATE DB CREATE DBDESC CREATE IMSCON CREATE PGM CREATE PGMDESC CREATE TRAN CREATE TRANDESC

Table 80. Command keywords and their synonyms (continued)

IMS keyword	Synonym	OM keyword	Commands that support this keyword
LINE	LINES	LINE	/ASSIGN /BROADCAST /DEQUEUE /DIAGNOSE /DISPLAY /END /EXCLUSIVE /EXIT /IDLE /LOOPTEST /MONITOR /PSTOP /PURGE /RSTART /START /STOP /TEST /TRACE
LINK			/ACTIVATE /CHANGE /DIAGNOSE /DISPLAY /IDLE /MSASSIGN /PSTOP /RSTART /TRACE
LMCT	LCT	LCT	/ASSIGN
LOCAL			/DBDUMP /DBRECOVERY /MSASSIGN /START /STOP
LOGOND	LGND		/CHANGE /OPNDST
LOPEN			/RSTART
LPRI			/ASSIGN CREATE TRAN CREATE TRANDESC

Table 80. Command keywords and their synonyms (continued)

IMS keyword	Synonym	OM keyword	Commands that support this keyword
LTERM	LTERMS	LTERM	/ASSIGN /BROADCAST DELETE LE /DEQUEUE /DIAGNOSE /DISPLAY /FORMAT /LOCK /MODIFY /PSTOP /PURGE QUERY LE QUERY OTMATI /RMxxxxxx /SET /START /STOP /UNLOCK UPDATE LE
LUNAME	LU, LUN	LU	/ALLOCATE /CHANGE /DEQUEUE /DISPLAY /EXIT /START /STOP /TRACE
MADSIOT			/START /STOP
MASTER			/BROADCAST /DISPLAY /RDISPLAY /SMCOPY
MAXRGN			/CHANGE CREATE TRAN CREATE TRANDESC
MEMBER			UPDATE ODBM
MODBLKS			INITIATE OLC
MODE	MODETABLE, MODETBL		/ALLOCATE /CHANGE /DISPLAY /OPNDST
MODIFY			/DISPLAY
MODNAME			QUERY OTMATI
MODULE			/DIAGNOSE /TRACE
MONITOR	MON		/DISPLAY /TRACE
MSDBLOAD			/NRESTART
MSG	MESSAGE		/SMCOPY /TRACE

Table 80. Command keywords and their synonyms (continued)

IMS keyword	Synonym	OM keyword	Commands that support this keyword
MSGAGE			/DISPLAY QUERY LTERM QUERY OTMATI
MSGTYPE			CREATE
MSLINK			QUERY UPDATE
MSNAME			/BROADCAST CREATE TRAN CREATE TRANDESC /DEQUEUE /DIAGNOSE /DISPLAY /MSASSIGN /MSVERIFY /PURGE /START /STOP
MSPLINK			/DISPLAY /MSASSIGN /PSTOP
MULTSIGN			/ERESTART /NRESTART

Table 80. Command keywords and their synonyms (continued)

IMS keyword	Synonym	OM keyword	Commands that support this keyword
NAME			CREATE DB, CREATE DBDESC, CREATE IMSCON CREATE OTMADESC, CREATE PGM, CREATE PGMDESC, CREATE RTC, CREATE RTCDESC, DELETE DB, DELETE DBDESC, DELETE OTMADESC, DELETE PGM, DELETE PGMDESC, DELETE RTC, DELETE RTCDESC, DELETE TRAN, DELETE TRANDESC, EXPORT, IMPORT INITIATE OLREORG, QUERY AREA, QUERY DB, QUERY DBDESC, QUERY IMSCON, QUERY IMSPLEX, QUERY LTERM, QUERY MSLINK, QUERY MSNAME, QUERY MSPLINK, QUERY NODE, QUERY ODBM, QUERY OTMADESC, QUERY PGM, QUERY PGMDESC, QUERY OLREORG, QUERY RTC, QUERY RTCDESC, QUERY STRUCTURE, QUERY TRAN, QUERY TRANDESC, QUERY USER, QUERY USERID, QUEUE LTERM, QUEUE TRAN, TERMINATE OLREORG, UPDATE AREA, UPDATE DATAGRP, UPDATE DB, UPDATE DBDESC, UPDATE IMSCON, UPDATE MSLINK, UPDATE MSNAME, UPDATE MSPLINK, UPDATE OLREORG, UPDATE OTMADESC, UPDATE PGM, UPDATE PGMDESC, UPDATE RTC, UPDATE RTCDESC, UPDATE TRAN, UPDATE TRANDESC
NOBACKOUT	NOBKO		/START
NOBMP			/ERESTART
NOBUILDQ	NBLDQ, NOBLDQ		/NRESTART
NOCHECK			/RECOVER
NOCMDAUTH			/ERESTART /NRESTART
NOCMDAUTHE			/ERESTART /NRESTART
NOCOMP			/TRACE
NOCQSSHUT			/CHECKPOINT
NODBALLOC			/START

Table 80. Command keywords and their synonyms (continued)

IMS keyword	Synonym	OM keyword	Commands that support this keyword
NODE			/ACTIVATE /ASSIGN /BROADCAST /CHANGE /CLSDST /COMPT /DEQUEUE /DIAGNOSE /DISPLAY /END /EXCLUSIVE /EXIT /IDLE /LOCK /OPNDST /QUIESCE /START /STOP /TEST /TRACE /UNLOCK
NOFEOV			/DBDUMP /DBRECOVERY
NOOPEN			/START
NOPFA			/DBDUMP /DBRECOVERY /STOP
NOREVERSE			/RTAKEOVER
NOSAVE			/ASSIGN /CHANGE
NOSHUT	NOS		/IDLE
NOTRANAUTH			/ERESTART /NRESTART
NOTRDY			/COMPT /RCOMPT
NOUSER			/ERESTART /NRESTART
NPRI			/ASSIGN CREATE TRAN CREATE TRANDESC
OASN			/CHANGE /DISPLAY
ODBM			QUERY UPDATE
OFFLINE			/RECOVER
OLC			INITIATE QUERY TERMINATE

Table 80. Command keywords and their synonyms (continued)

IMS keyword	Synonym	OM keyword	Commands that support this keyword
OLDS			/DISPLAY /START /STOP /SWITCH
OLREORG		OLREORG	INITIATE /INITIATE QUERY /QUERY TERMINATE /TERMINATE UPDATE /UPDATE
OPEN			/START
OPTION		OPTION	DELETE DB DELETE DBDESC DELETE OTMADESC DELETE PGM DELETE PGMDESC DELETE RTC DELETE RTCDESC DELETE TRAN DELETE TRANDESC /DIAGNOSE EXPORT IMPORT INITIATE OLC INITIATE OLREORG QUERY AREA QUERY LTERM QUERY OTMADESC QUEUE LTERM QUEUE TRAN TERMINATE OLREORG /TRACE UPDATE AREA UPDATE DATAGRP UPDATE DB UPDATE DBDESC UPDATE IMS UPDATE MSLINK UPDATE ODBM UPDATE OLREORG UPDATE PGM UPDATE PGMDESC UPDATE RTC UPDATE RTCDESC UPDATE TRAN UPDATE TRANDESC
OSAMGTF			/DISPLAY /TRACE
OTMA			/DISPLAY /SECURE /START /STOP

Table 80. Command keywords and their synonyms (continued)

IMS keyword	Synonym	OM keyword	Commands that support this keyword
OTMADESC			CREATE DELETE QUERY UPDATE
OUTBND	OUTBOUND		/CHANGE
OUTPUT			/ASSIGN /DISPLAY /START /STOP /TRACE
OVERFLOWQ	OFLWQ	OVERFLOWQ	/DISPLAY
OVERRIDE	OVER		/ERESTART
PARLIM			/ASSIGN CREATE TRAN CREATE TRANDESC
PASSWORD	PASSWORDS, PSWD, PSWDS	PSWD	/CHANGE /DELETE /MODIFY PREPARE /NRESTART
PCH			/COMPT /RCOMPT
PDS			/COMPT /RCOMPT
PGMTYPE			CREATE PGM CREATE PGMDESC
PHASE			INITIATE OLC
PI			/DISPLAY /TRACE
PITR			/RECOVER
PLCT			CREATE TRAN CREATE TRANDESC
PLCTTIME			CREATE TRAN CREATE TRANDESC
PLEXPARM			UPDATE IMS
PLMCT	PLCT	PLCT	/ASSIGN
POOL			/DISPLAY
PREPARE			/MODIFY
PRIORITY	PRTY		/DISPLAY

Table 80. Command keywords and their synonyms (continued)

IMS keyword	Synonym	OM keyword	Commands that support this keyword
PROGRAM	PGM, PGMS, PROG, PROGRAMS, PROGS	PGM	CREATE PGM CREATE PGMDESC CREATE TRAN CREATE TRANDESC DELETE LE DELETE PGM DELETE PGMDESC /DIAGNOSE DISPLAY /LOCK QUERY LE QUERY PGM QUERY PGMDESC /START /STOP /TRACE /UNLOCK UPDATE LE UPDATE PGM UPDATE PGMDESC
PRT			/COMPT /RCOMPT
PRTKN	PRK		/CHANGE CCTL
PSB			/DISPLAY QUERY ODBM /TRACE
PSWD			INITIATE OLC
PTERM	PTERMS	PTERM	/ASSIGN /BROADCAST /DEQUEUE /DISPLAY /END /EXCLUSIVE /EXIT /LOCK /LOOPTEST /MONITOR /PSTOP /PURGE /START /STOP /TEST /UNLOCK
PURGE			/CHECKPOINT /DEQUEUE /PSTOP
PURGE1	FIRST, SINGLE, SNGL		/DEQUEUE
Q	QS, QUEUE, QUEUES	Q	/DISPLAY /OPNDST

Table 80. Command keywords and their synonyms (continued)

IMS keyword	Synonym	OM keyword	Commands that support this keyword
QCNT			/DISPLAY QUERY LTERM QUERY MSNAME QUERY TRAN
QMGR			/TRACE TABLE
QUIESCE			/CHECKPOINT
RCVTIME			/RECOVER
RCVTOKEN			/RECOVER
RDDSDSN			EXPORT
RDR			/COMPT /RCOMPT
READNUM			/RECOVER
READY			/COMPT /RCOMPT
RECOVER			CREATE TRAN CREATE TRANDESC
RECOVERY	REC	RECOVERY	/DISPLAY
RECOVGRP	RCVGRP		/RECOVER
REGION	REGIONS, REG, REGS, MSGREG, MSGREGS, MSGREGION, MSREGIONS, THREAD	REGION	/ASSIGN /DIAGNOSE /DISPLAY /PSTOP /START /STOP /TRACE MONITOR
REMOTE			CREATE TRAN CREATE TRANDESC /DISPLAY UPDATE DBDESC
REMOVE	REM	REMOVE	/RECOVER
RESET			/CHANGE
RESIDENT			CREATE DB CREATE DBDESC CREATE PGM CREATE PGMDESC
RESP			CREATE TRAN CREATE TRANDESC
RTCODE	RTC	RTC	/DISPLAY /START /STOP
SAVE			/ASSIGN
SB			/START /STOP
SCOPE			IMPORT UPDATE AREA UPDATE DATAGRP UPDATE DB UPDATE TRAN

Table 80. Command keywords and their synonyms (continued)

IMS keyword	Synonym	OM keyword	Commands that support this keyword
SCHD			/TRACE MONITOR
SCHDTYPE			CREATE PGM CREATE PGMDESC
SCIMEMBER			QUERY ODBM
SEGNO			/ASSIGN CREATE TRAN CREATE TRANDESC
SEGSIZE	SEGSZ	SEGSZ	/ASSIGN CREATE TRAN CREATE TRANDESC
SERIAL			CREATE TRAN CREATE TRANDESC
SERVGRP	SG	SERVGRP	/START /STOP
SET			CREATE DB CREATE DBDESC CREATE IMSCON CREATE OTMADESC CREATE PGM CREATE PGMDESC CREATE RTC CREATE RTCDESC CREATE TRAN CREATE TRANDESC /DIAGNOSE EXPORT INITIATE OLREORG /TRACE UPDATE DB UPDATE DBDESC UPDATE IMSCON, UPDATE LE UPDATE MSLINK UPDATE MSNAME UPDATE MSPLINK UPDATE OLREORG UPDATE OTMADESC UPDATE PGM UPDATE PGMDESC UPDATE RTC UPDATE RTCDESC UPDATE TRAN UPDATE TRANDESC
SHAREDQ	SHRQ	SHAREDQ	/CQCHKPT /CQSET

Table 80. Command keywords and their synonyms (continued)

IMS keyword	Synonym	OM keyword	Commands that support this keyword
SHOW			/DIAGNOSE QUERY AREA QUERY DB QUERY DBDESC QUERY IMS QUERY IMSPLEX QUERY LE QUERY LTERM QUERY MEMBER QUERY MSLINK QUERY MSNAME QUERY MSPLINK QUERY NODE QUERY ODBM QUERY OTMADESC QUERY OTMATI QUERY PGM QUERY PGMDESC QUERY POOL QUERY OLC QUERY OLREORG QUERY RTC QUERY RTCDESC QUERY STRUCTURE QUERY TRAN QUERY TRANDESC QUERY USER QUERY USERID
SHUTDOWN			/CQSET /DISPLAY
SIDE			/CHANGE
SIDL			CREATE TRAN CREATE TRANDESC
SIDR			CREATE TRAN CREATE TRANDESC
SLDSREAD			/START /STOP
SMEM			QUERY OTMADESC
SMSOPTS			/RECOVER
SNAP			/DIAGNOSE
SNAPQ			/CHECKPOINT
SNGLSIGN			/ERESTART /NRESTART
SOURCE			IMPORT
SPASZ			CREATE TRAN CREATE TRANDESC
SPATRUNC			CREATE TRAN CREATE TRANDESC
SQTT			/TRACE TABLE
SSM			/START

Table 80. Command keywords and their synonyms (continued)

IMS keyword	Synonym	OM keyword	Commands that support this keyword
STAGLOBAL			/RECOVER
STALOCAL			/RECOVER
START	STA	START	/RECOVER UPDATE AREA UPDATE DATAGRP UPDATE DB UPDATE MSLINK UPDATE MSNAME UPDATE MSPLINK UPDATE ODBM UPDATE PGM UPDATE RTC UPDATE TRAN
STARTTIME			EXPORT
STATISTICS			/CHECKPOINT /CQQUERY QUERY STRUCTURE
STATUS			/DISPLAY QUERY AREA QUERY DB QUERY IMSPLEX QUERY LTERM QUERY MEMBER QUERY MSLINK QUERY MSNAME QUERY MSPLINK QUERY NODE QUERY ODBM QUERY PGM QUERY OLREORG QUERY RTC QUERY TRAN QUERY USER
STOP	STO	STOP	/RECOVER UPDATE AREA UPDATE DATAGRP UPDATE DB UPDATE MSLINK UPDATE MSNAME UPDATE MSPLINK UPDATE ODBM UPDATE PGM UPDATE RTC UPDATE TRAN
STRUCTURE	STRUC	STRUCTURE	/CQCHKPT /CQQUERY /CQSET /DIAGNOSE /DISPLAY
SUBSYS	SUBSYSTEM, SUBSYSTEMS	SUBSYS	/CHANGE /DISPLAY /START /STOP

Table 80. Command keywords and their synonyms (continued)

IMS keyword	Synonym	OM keyword	Commands that support this keyword
SURVEILLANCE	SUR, SURV, SURVEIL	SURV	/CHANGE /START /STOP
SUSPEND			/DEQUEUE
SYNCLEVEL	SYNCLV		/CHANGE QUERY OTMATI
SYNCSESS	SYNC		/CHANGE
SYSID			/BROADCAST /DISPLAY /MSASSIGN /MSVERIFY
SYSTEM			/CQCHKPT /SWITCH /UNLOCK
TABLE			/DISPLAY /TRACE
TAKEOVER	TKO, TKOVR		/TRACE
TARGET			EXPORT
TCO			/DISPLAY /TRACE
TDS			/COMPT /RCOMPT
TERMINAL	TER, TERM, TERMINALS, TERMS, TERS	TERMINAL	INITIATE OLC /SMCOPY
THREAD			/START /STOP
TIMEOUT	TIMO		/CHANGE /DISPLAY /TRACE
TIMEOVER			/DISPLAY
TMEMBER	TMEM	TMEM	/DEQUEUE /DISPLAY /EXIT QUERY OTMADESC QUERY OTMATI /START /STOP /TRACE
TPIPE	TPI		/DEQUEUE /DISPLAY QUERY OTMADESC QUERY OTMATI /START /STOP /TRACE

Table 80. Command keywords and their synonyms (continued)

IMS keyword	Synonym	OM keyword	Commands that support this keyword
TPNAME	TP, TPN		/ALLOCATE /CHANGE /DEQUEUE /DISPLAY /START /STOP /TRACE
TRACE			/DISPLAY
TRACKING			/DISPLAY
TRACKING STATUS			/DISPLAY
TRANAUTH			/ERESTART /NRESTART
TRANCMD5			INITIATE OLC
TRANSACTION	TRANS, TRAN, TRANSACTIONS, TRANCODE, TRANCODES, TRS	TRAN	/ASSIGN /CHANGE CREATE TRAN CREATE TRANDESC DELETE LE DELETE TRAN DELETE TRANDESC /DEQUEUE /DIAGNOSE /DISPLAY /LOCK /MSASSIGN /PSTOP /PURGE QUERY LE QUERY OTMATI QUERY TRAN QUERY TRANDESC /SET /START /STOP /TRACE /UNLOCK UPDATE LE UPDATE TRAN UPDATE TRANDESC
TRANSTAT			CREATE PGM CREATE PGMDESC CREATE TRAN CREATE TRANDESC
TRAP			/DISPLAY /TRACE
TRKAUTOARCH	TRKARCH	TRKARCH	/START
TRS			/DISPLAY MODIFY

Table 80. Command keywords and their synonyms (continued)

IMS keyword	Synonym	OM keyword	Commands that support this keyword
TYPE			/CHANGE CREATE IMSCON EXPORT IMPORT INITIATE OLC QUERY DB QUERY IMS QUERY IMSCON QUERY IMSPLEX QUERY MEMBER QUERY MSPLINK QUERY ODBM QUERY OTMADESC QUERY POOL QUERY STRUCTURE UPDATE IMSCON UPDATE ODBM
UDATA			/OPNDST
UDS			/COMPT /RCOMPT
UNITYPE			/TRACE
UNPLAN			/RTAKEOVER
UOR			/CHANGE /DISPLAY
USEDDBDS	USEAREA		/RECOVER
USER	SUBPOOL	USER	/ASSIGN /BROADCAST /CHANGE /CLSDST /COMPT /DEQUEUE /DIAGNOSE /DISPLAY /END /ERESTART /EXCLUSIVE /EXIT /NRESTART /OPNDST /QUIESCE /START /STOP /TEST /TRACE
USERD	USRD		/OPNDST
USERID			DELETE LE QUERY LE QUERY OTMATI UPDATE LE
VERIFY			/RECOVER
VGRS	VGR	VGR	/STOP

Table 80. Command keywords and their synonyms (continued)

IMS keyword	Synonym	OM keyword	Commands that support this keyword
VID			/COMPT /RCOMPT
VOLUME			/TRACE
VTAMPOOL	VPL		/ASSIGN
WADS			/STOP /SWITCH
WFI			CREATE TRAN CREATE TRANDESC
WPM1			/COMPT
WPM2			/COMPT
WPM3			/COMPT
XRCTRACK			/START /STOP
XTRC	XTRACE, EXTRACE		/DISPLAY TRACE

**Notes:**

<sup>1</sup> ISOLOG and TRKAUTOARCH are valid only for an RSR tracking subsystem.

**Related reference:**

 [IMS Command Language Modification facility \(DFSCKWD0\) \(Exit Routines\)](#)

## Chapter 3. Commands and keywords valid in DBCTL

Commands and keywords that are valid in the Database Control (DBCTL) environment are listed.

The following table lists commands and keywords valid in the Database Control (DBCTL) environment. All commands and keywords are valid in a DB/DC environment.

Table 81. Commands and keywords valid in DBCTL

Commands	Keywords	Topic
/CHANGE	ABORT, AUTOLOGON, CCTL, COMMIT, FDR, OASN, PASSWORD, PRKTN, RESET, SUBSYS, TIMEOUT	Chapter 12, “/CHANGE commands,” on page 129
/CHECKPOINT	ABDUMP, FREEZE, PURGE, STATISTICS	Chapter 13, “/CHECKPOINT command,” on page 153
CREATE	PGMDESC, PGM	Chapter 19, “CREATE commands,” on page 177
/DBDUMP	DB, GLOBAL, LOCAL, NOFEOV, NOPFA	Chapter 20, “/DBDUMP command,” on page 339
/DBRECOVERY	AREA, DB, DATAGRP, GLOBAL, LOCAL, NOFEOV, NOPFA	Chapter 21, “/DBRECOVERY command,” on page 345
/DELETE	DB, PASSWORD, PGM	Chapter 22, “/DELETE command,” on page 355
DELETE	DB, DBDESC, DEFN, LE, PGM, PGMDESC	Chapter 23, “DELETE commands,” on page 357
/DEQUEUE	AOITOKEN	Chapter 24, “/DEQUEUE command,” on page 471
/DIAGNOSE	AREA, ADDRESS, BLOCK, DB, JOBNAME, MODULE, OPTION, PGM, REGION, SET, SHOW, SNAP	Chapter 25, “/DIAGNOSE commands,” on page 479
/DISPLAY	ACTIVE, AOITOKEN, AREA, BKERR, CCTL, CPLOG, DB, DBD, FDR, FPVIRTUAL, HSSP, INDOUBT, MADSIOT, MODIFY, MONITOR, OASN, OLDS, OSAMGTE, PI, POOL, PGM, PSB, RECOVERY, REGION, SHUTDOWN, STATUS, SUBSYS, TABLE, TCO, TRACE, TRACKING STATUS, XTRC	Chapter 26, “/DISPLAY commands,” on page 521
/ERESTART	CHECKPOINT, COLDBASE, COLDSYS, FORMAT, NOBMP, OVERRIDE	Chapter 28, “/ERESTART command,” on page 769
EXPORT	DEFN, ENDTIME, NAME, OPTION, RDDSDSN, SET, STARTTIME, TARGET, TYPE	Chapter 31, “EXPORT command,” on page 787
IMPORT DEFN SOURCE(CATALOG)	DEFN, FOR, OPTION, SOURCE	“IMPORT DEFN SOURCE(CATALOG) command” on page 827
IMPORT DEFN SOURCE(REPO   RDDSDSN)	DEFN, NAME, OPTION, RDDSDSN, SCOPE, SOURCE, TYPE	“IMPORT DEFN SOURCE(REPO   RDDSDSN) command” on page 848
INITIATE	OLC, OLREORG	Chapter 36, “INITIATE commands,” on page 891

Table 81. Commands and keywords valid in DBCTL (continued)

Commands	Keywords	Topic
/LOCK	DB, PGM	Chapter 37, “/LOCK command,” on page 949
/LOG		Chapter 38, “/LOG command,” on page 955
/MODIFY	ABORT, COMMIT, PASSWORD, PREPARE	Chapter 40, “/MODIFY command,” on page 959
/NRESTART	CHECKPOINT, FORMAT	/NRESTART command (Commands)
/PSTOP	AOITOKEN, JOBNAME, REGION	/PSTOP command (Commands)
QUERY	AREA, DB, DBDESC, IMS, LE, MEMBER, ODBM, OLC, OLREORG, PGM, PGMDESC, POOL, USEREXIT	QUERY commands (Commands)
/RECOVER	ALLENTRIES, AREA, CAGROUP, DB, DBDS, DBDSGRP, ERRORABORT, ERRORCONT, NOCHECK, OFFLINE, PITR, RCVTIME, RCVTOKEN, READNUM, RECOVGRP, REMOVE, SMSOPTS, STAGLOBAL, STALOCAL, START, STOP, USEAREA, USEDDBDS, VERIFY	/RECOVER commands (Commands)
REFRESH	USEREXIT	REFRESH USEREXIT command (Commands)
/RMxxxxxx		/RMxxxxxx commands (Commands)
/RTAKEOVER	DUMPQ, FREEZE, NOREVERSE, UNPLAN	/RTAKEOVER command (Commands)
/SSR		/SSR command (Commands)
/START	ACCESS, AREA, AUTOARCH, DB, DATAGRP, DBALLOC, GLOBAL, ISOLOG, JOBNAME, LOCAL, MADSIOT, NOBACKOUT, NODBALLOC, NOOPEN, OLDS, OPEN, PGM, REGION, SB, SERVGRP, SLDSREAD, SSM, SUBSYS, THREAD, TRKAUTOARCH, WADS, XRCTRACK	/START commands (Commands)
/STOP	ABDUMP, ADS, AREA, AUTOARCH, CANCEL, DB, DATAGRP, GLOBAL, JOBNAME, LOCAL, MADSIOT, NOPFA, OLDS, PGM, REGION, SB, SERVGRP, SLDSREAD, SUBSYS, WADS, XRCTRACK	/STOP commands (Commands)
/SWITCH	CHECKPOINT, OLDS, WADS	/SWITCH command (Commands)
TERMINATE	OLC, OLREORG	TERMINATE OLC command (Commands)
/TRACE	COMP, MONITOR, NOCOMP, OPTION, OSAMGTF, PI, PGM, PSB, SET, TABLE, TCO, VOLUME	/TRACE commands (Commands)
/UNLOCK	DB, PGM	/UNLOCK commands (Commands)
UPDATE	AREA, DB, DBDESC, DATAGRP, IMS, LE, ODBM, OLREORG, PGM, PGMDESC, POOL	UPDATE commands (Commands)
/VUNLOAD	AREA	/VUNLOAD command (Commands)

## Chapter 4. Commands and keywords valid in DCCTL

Commands and keywords that are valid in a Data Communications Control (DCCTL) environment are listed.

The following table lists commands and keywords valid in a Data Communications Control (DCCTL) environment. All commands and keywords are valid in a DB/DC environment.

*Table 82. Commands and keywords valid in DCCTL*

Commands	Keywords	Topic
/ACTIVATE	LINK, NODE	Chapter 7, “/ACTIVATE command,” on page 97
/ALLOCATE	LUNAME, MODE, TPNAME	Chapter 8, “/ALLOCATE command,” on page 99
/ASSIGN	CLASS, COMPONENT, CPRI, COMPONENT, INPUT, LINE, LCT, LPRI, LTERM, NODE, NOSAVE, NPRI, OUTPUT, PARLIM, PLCT, PTERM, REGION, SAVE, SEGNO, SEGSIZE, TRANSACTION, USER, VTAMPOOL	Chapter 9, “/ASSIGN command,” on page 101
/BROADCAST	ACTIVE, LINE, LTERM, MASTER, MSNAME, NODE, PTERM, SYSID, USER	Chapter 10, “/BROADCAST command,” on page 121
/CANCEL		Chapter 11, “/CANCEL command,” on page 127
/CHANGE	APPC, ASR, AUTOLOGON, COLDSESS, CPLOG, DESCRIPTOR, DIRECTORY, FORCSESS, ID, INTERVAL, LINK, LOGOND, LUNAME, MAXRGN, MODE, NODE, NOSAVE, OASN, OUTBND, PASSWORD, RESET, SAVE, SIDE, SUBSYS, SURVEILLANCE, SYNCLEVEL, SYNCSESS, TIMEOUT, TPNAME, TRANSACTION, TYPE, UOR, USER	Chapter 12, “/CHANGE commands,” on page 129
/CHECKPOINT	ABDUMP, DUMPQ, FREEZE, LEAVEPLEX, NOCQSSHUT, PURGE, QUIESCE, SNAPQ, STATISTICS	Chapter 13, “/CHECKPOINT command,” on page 153
/CLSDST	FORCE, NODE, USER	Chapter 14, “/CLSDST command,” on page 161
/COMPT	CNS, CRD, NODE, NOTRDY, PCH, PDS, PRT, RDR, READY, TDS, UDS, USER, VID, WPM1, WPM2, WPM3	Chapter 15, “/COMPT command,” on page 165
/CQCHKPT	SHAREDQ, STRUCTURE, SYSTEM	Chapter 16, “/CQCHKPT command,” on page 169
/CQQUERY	STATISTICS, STRUCTURE	Chapter 17, “/CQQUERY command,” on page 171
/CQSET	SHAREDQ, SHUTDOWN, STRUCTURE	Chapter 18, “/CQSET command,” on page 175
CREATE	OTMADESC, PGM, PGMDESC, RTC, RTCDESC, TRAN, TRANDESC	Chapter 19, “CREATE commands,” on page 177
/DELETE	LINE, LTERM, NODE, PASSWORD, PGM, PTERM, TERMINAL, TRANSACTION	Chapter 22, “/DELETE command,” on page 355

Table 82. Commands and keywords valid in DCCTL (continued)

Commands	Keywords	Topic
DELETE	LE, OTMADESC, PGM, PGMDESC, RTC, RTCDESC, TRAN, TRANDESC	Chapter 23, "DELETE commands," on page 357
/DEQUEUE	AOITOKEN, LINE, LTERM, LUNAME, MSNAME, NODE, PTERM, PURGE, PURGE1, SUSPEND, TMEMBER, TPIPE, TPNAME, TRANSACTION, USER	Chapter 24, "/DEQUEUE command," on page 471
/DIAGNOSE	ADDRESS, AOSLOG, BLOCK, JOBNAME, LINE, LINK, LTERM, MODULE, MSNAME, NODE, OPTION, PGM, REGION, RM, SET, SHOW, SNAP, STRUCTURE, TRAN, USER	Chapter 25, "/DIAGNOSE commands," on page 479
/DISPLAY	ACT, AFFIN, AOITKN, APPC, ASMT, AUTOLOGON, BALGRP, CLASS, CONVERSATION, CPLOG, CQS, DC, DESCRIPTOR, EMHQ, EXIT, HSB, INPUT, LINE, LINK, LTERM, LUNAME, MASTER, MODE, MODIFY, MONITOR, MSGAGE, MSNAME, MSPLINK, NODE, OASN, OLDS, OTMA, OUTPUT, OVERFLOWQ, PGM, POOL, PRIORITY, PGM, PSB, PTERM, Q, QCNT, REGION, REMOTE, RTCODE, SHUTDOWN, STATUS, STRUCTURE, SUBSYS, SYSID, TABLE, TCO, TIMEOUT, TIMEOVER, TMEMBER, TPIPE, TPNAME, TRACE, TRACKING STATUS, TRANSACTION, TRAP, UOR, USER, XTRC	Chapter 26, "/DISPLAY commands," on page 521
/END	LINE, NODE, PTERM, USER	Chapter 27, "/END command," on page 767
/ERESTART	BACKUP, BUILDQ, CHECKPOINT, CMDAUTH, CMDAUTHE, COLDCOMM, COLDSYS, FORMAT, MULTSIGN, NOBMP, NOCMDAUTH, NOCMDAUTHE, NOPASSWORD, NOTERMINAL, NOTRANAUTH, NOTRANCMDs, NOUSER, OVERRIDE, PASSWORD, SNGLSIGN, TERMINAL, TRANAUTH, TRANCMDs, USER	Chapter 28, "/ERESTART command," on page 769
/EXCLUSIVE	LINE, NODE, PTERM, USER	Chapter 29, "/EXCLUSIVE command," on page 781
/EXIT	CONVERSATION, LINE, NODE, PTERM, USER	Chapter 30, "/EXIT command," on page 783
/FORMAT	LTERM	Chapter 32, "/FORMAT command," on page 819
/HOLD		Chapter 33, "/HOLD command," on page 821
/IDLE	LINE, LINK, NODE, NOSHUT	Chapter 34, "/IDLE command," on page 823
IMPORT DEFN SOURCE(REPO   RDDS)	DEFN, NAME, OPTION, RDDSdsn, SCOPE, SOURCE, TYPE	"IMPORT DEFN SOURCE(REPO   RDDS) command" on page 848
INITIATE	OLC, OLREORG	Chapter 36, "INITIATE commands," on page 891
/LOCK	LTERM, NODE, PGM, PTERM, TRANSACTION	Chapter 37, "/LOCK command," on page 949
/LOG		Chapter 38, "/LOG command," on page 955

Table 82. Commands and keywords valid in DCCTL (continued)

Commands	Keywords	Topic
/LOOPTEST	LINE, PTERM	Chapter 39, “/LOOPTEST command,” on page 957
/MODIFY	ABORT, COMMIT, LTERM, PASSWORD, PREPARE, TERMINAL, TRANCMD5	Chapter 40, “/MODIFY command,” on page 959
/MONITOR	LINE, PTERM	Chapter 41, “/MONITOR command,” on page 967
/MSASSIGN	LINK, LOCAL, MSNAME, MSPLINK, SYSID, TRANSACTION	Chapter 42, “/MSASSIGN command,” on page 969
/MSVERIFY	MSNAME, SYSID	Chapter 43, “/MSVERIFY command,” on page 973
/NRESTART	BUILDQ, CHECKPOINT, CMDAUTH, CMDAUTHE, FORMAT, MULTSIGN, NOBUILDQ, NOCMDAUTH, NOCMDAUTHE, NOPASSWORD, NOTERMINAL, NOTRANAUTH, NOTRANCMD5, NOUSER, PASSWORD, SNGLSIGN, TERMINAL, TRANAUTH, TRANCMD5, USER	/NRESTART command (Commands)
/OPNDST	ID, LOGOND, MODE, NODE, Q, UDATA, USER, USERD	/OPNDST command (Commands)
/PSTOP	AOITOKEN, CLASS, FORCE, JOBNAME, LINE, LINK, LTERM, MSPLINK, PTERM, PURGE, REGION, TRANSACTION	/PSTOP command (Commands)
/PURGE	APPC, CLASS, FPPROG, FPREGION, LINE, LTERM, MSNAME, PTERM, TRANSACTION	/PURGE command (Commands)
QUERY	IMS, LE, LTERM, MEMBER, MSLINK, MSNAME, MSPLINK, NODE, OLC, OTMADESC, OTMATI, PGM, PGMDESC, RTC, RTCDESC, TRAN, TRANDESC, USER, USEREXIT, USERID	QUERY commands (Commands)
QUEUE	LTERM, TRAN	QUEUE commands (Commands)
/QUIESCE	NODE, USER	/QUIESCE command (Commands)
/RCLSDST		/RCLSDST command (Commands)
/RCOMPT	CNS, NOTRDY, PCH, PDS, PRT, RDR, READY, TDS, UDS, VID	/RCOMPT command (Commands)
/RDISPLAY	MASTER	/RDISPLAY command (Commands)
REFRESH	USEREXIT	REFRESH USEREXIT command (Commands)
/RELEASE	CONVERSATION	/RELEASE command (Commands)
/RESET		/RESET command (Commands)
/RMxxxxxx	LTERM	/RMxxxxxx commands (Commands)
/RSTART	CONTINUOUS, LINE, LINK, LOPEN, MODE, MSPLINK, NODE, PTERM, USER	/RSTART command (Commands)
/RTAKEOVER	DUMPQ, FREEZE, NOREVERSE, UNPLAN	/RTAKEOVER command (Commands)
/SECURE	APPC, OTMA	/SECURE command (Commands)
/SET	CONVERSATION, LTERM, TRANSACTION	/SET command (Commands)
/SIGN		/SIGN command (Commands)
/SMCOPY	MASTER, TERMINAL	/SMCOPY command (Commands)

Table 82. Commands and keywords valid in DCCTL (continued)

Commands	Keywords	Topic
/SSR		/SSR command (Commands)
/START	AFFINITY, APPC, AUTOARCH, CLASS, DC, DESC, GRSNAME, INPUT, ISOLOG, JOBNAME, LINE, LTERM, LUNAME, MSNAME, NODE, OLDS, OTMA, OUTPUT, PGM, PTERM, REGION, RTCODE, SERVGRP, SLDSREAD, SSM, SUBSYS, SURVEILLANCE, TMEMBER, TPIPE, TPNAME, TRANSACTION, TRKAUTOARCH, USER, VGR, WADS, XRCTRACK	/START commands (Commands)
/STOP	ABDUMP, APPC, AUTOARCH, BACKUP, CANCEL, CLASS, DESC, DC, INPUT, JOBNAME, LINE, LTERM, LUNAME, MSNAME, NODE, OLDS, OTMA, OUTPUT, PGM, PTERM, REGION, RTCODE, SERVGRP, SLDSREAD, SUBSYS, SURVEILLANCE, THREAD, TMEMBER, TPIPE, TPNAME, TRANSACTION, USER, VGR, WADS, XRCTRACK	/STOP commands (Commands)
/SWITCH	ABDUMP, ACTIVE, BACKUP, CHECKPOINT, FORCE, OLDS, SYSTEM, WADS	/SWITCH command (Commands)
TERMINATE	OLC	TERMINATE OLC command (Commands)
/TEST	LINE, NODE, PTERM, USER	/TEST command (Commands)
/TRACE	AUTO, EXIT, INPUT, LEVEL, LINE, LINK, LUNAME, MODULE, MONITOR, MSG, NODE, OPTION, OUTPUT, PGM, SET, TABLE, TAKEOVER, TCO, TIMEOUT, TMEMBER, TPIPE, TPNAME, TRANSACTION, TRAP, UNITYPE, USER, VOLUME	/TRACE commands (Commands)
/UNLOCK	LTERM, NODE, PGM, PTERM, SYSTEM, TRANSACTION	/UNLOCK commands (Commands)
UPDATE	IMS, LE, MSLINK, MSNAME, MSPLINK, OTMADESC, PGM, PGMDESC, POOL, RTC, RTCDESC, TRAN, TRANDESC	UPDATE commands (Commands)

## Chapter 5. List of reserved words

Some words are used and reserved only for IMS commands.

The following table is a list of words that cannot be used to name resources such as transactions or databases. The words listed in this table are used and reserved only for IMS commands.

*Table 83. Words used and reserved only for IMS commands*

A	ABDUMP	ABORT	ACCESS
ACT	ACTIV	ACTIVATE	ACTIVE
ADDS	ADS	AFF	AFFIN
AFFINITY	AFTER	AOITKN	APDB
APMQ	AOITOKEN	ALL	ALLENTRIES
ALLENT	ALLOC	ALLOCATE	ALLOCF
ALLOCS	APPC	ARCHIVE	AREA
ASMT	ASR	ASSIGN	ASSIGNMENT
AUTO	AUTOARC	AUTOARCH	AUTOLOGON
AUTOLGN	AUTOSR	BACKOUT	BACKUP
BALG	BALGRP	BKERR	BLDQ
BLDQS	BROADCAST	BU	BUILDQ
BUILDQS	C1INOP	C2INOP	C3INOP
C4INOP	CAGROUP	CAGRP	CANCEL
CC	CCTL	CHANGE	CHECKPOINT
CHECKPT	CHKPOINT	CHKPT	CHNGS
CLASS	CLS	CLSDST	CMDAUTH
CMDAUTHE	CNS	COLDB	COLDBASE
COLDC	COLDCOMM	COLDS	COLDSESS
COLDSYS	COMMIT	COMP	COMPINOP
COMPONENT	COMPT	CON	CONT
CONTINUOUS	CONV	CONVACT	CONVERSATION
CONVHLD	CPRI	CQCHKPT	CQQUERY
CPLOG	CQC	CQQ	CQS
CQSET	CRD	DATABASE	DATABASES
DATAGROUP	DATAGRP	DB	DBALLOC
DBD	DBDS	DBDSGRP	DBDUMP
DBR	DBRECOVERY	DBS	DC
DEACT	DEADQ	DELETE	DELS
DEQUEUE	DESC	DESCRIPTOR	DL/I
DLOG	DIS	DISP	DISPLAY
DIR	DIRECTORY	DMS	DONE
DUMPQ	DUMPQS	EEQE	EMHQ

Table 83. Words used and reserved only for IMS commands (continued)

END	ERESTART	ERRORCONT	EXCL
EXCLUSIVE	EXIT	EXTRACE	FAST
FDR	FIRST	FMS	FMT
FOR	FORC	FORCE	FORCES
FORCSESS	FORMAT	FPPROG	FPREGION
FPRGN	FPV	FPVIRTUAL	FREEZE
GLOBAL	GRS	GRSN	GRSNAME
HOLD	HOTSTANDBY	HSB	HSSP
IAM	IC	ICOMPONENT	ICOMPT
ID	IDCO	IDLE	IND
INDOUBT	INOP	INPUT	INQONLY
INT	INTERVAL	INTV	IOVF
ISOLOG	JBN	JOB	JOBNAME
KEY	LA	LATC	LCT
LEAVEGR	LEVEL	LGND	LINE
LINES	LINK	LMCT	LOC
LOCAL	LOCK	LOG	LOGOND
LOOPTEST	LOPEN	LOST	LPRI
LRTT	LTERM	LTERMS	LU
LUMI	LUN	LUNAME	MADSIOT
MASTER	MAXRGN	MESSAGE	MFST
MFSTEST	MODE	MODETABLE	MODETBL
MODIFY	MODS	MODULE	MON
MONITOR	MSASSIGN	MSDB	MSDBLOAD
MSG	MSGAGE	MSGREG	MSGREGION
MSGREGIONS	MSGREGS	MSNAME	MSPLINK
MSVERIFY	MULTSIGN	NBLDQ	NOBACKOUT
NOBKO	NOBLDQ	NOBMP	NOBUILDQ
NOCHECK	NOCMDAUTH	NOCMDAUTHE	NOCOMP
NODBALLOC	NODE	NOFEOV	NOIN
NONE	NOOUT	NOPASSWORD	NOPFA
NOPSWD	NOQUEUE	NOREVERSE	NOS
NOSAVE	NOSHUT	NOTER	NOTERM
NOTERMINAL	NOTINIT	NOTOPEN	NOTRANAUTH
NOTRANCMD	NOTRDY	NOUSER	NPRI
NRESTART	OASN	OFF	OFFLINE
OFR	OLDS	OLREORG	ON
OPNDST	OPTION	OSAMGTF	OTMA
OTMT	OUTPUT	OVER	OVERFLOWQ
OVERRIDE	OVFLWQ	PAGE	PARLIM
PASSWORD	PASSWORDS	PCH	PDS

Table 83. Words used and reserved only for IMS commands (continued)

PGM	PGMS	PI	PITR
PLCT	PLMCT	POOL	PREL
PREO	PREPARE	PRI	PRIMARY
PRIORITY	PRK	PROG	PROGRAM
PROGRAMS	PROGS	PRST	PRT
PRTKN	PRTY	PSB	PSS
PSTOP	PSTOPPED	PSWD	PSWDS
PTERM	PTERMS	PUR	PURGE
PURGE1	Q	QCNT	QMGR
QRTT	QS	QUEUE	QUEUES
QUI	QUIESCE	RCLSDST	RCOMPT
RCS	RCVTIME	RCVTOKEN	RDISPLAY
RDR	READNUM	READY	RECOVERY
RECOVGRP	REG	REGION	REGIONS
REGS	RELEASE	RELREQ	REMOTE
REMOVE	RESET	RESP	RESPINP
RESYNC	RETR	RMCHANGE	RMDELETE
RMGENJCL	RMINIT	RMLIST	RMNOTIFY
RSTART	RTAKEOVER	RTC	RTCODE
SAVE	SB	SCHD	SEC
SECURE	SECURITY	SEGNO	SEGSIZE
SEGSZ	SERVGRP	SET	SG
SHAREDQ	SHRQ	SHUT	SHUTDOWN
SIDE	SIGN	SIMLOGON	SINGLE
SMCOPY	SNAPQ	SNGL	SNGLSIGN
SQTT	SSM	SSR	STAGLOBAL
STALOCAL	STA	START	STATIC
STATISTICS	STATUS	STO	STOP
STOPPED	STRG	STRUC	STRUCTURE
SUB	SUBS	SUBPOOL	SUBSYS
SUBSYSTEMEMBER	SUBSYSTEM	SUBSYSTEMS	SUR
SURV	SURVEIL	SURVEILLANCE	SUSPEND
SWITCH	SYNC	SYNLEVEL	SYNCLV
SYNCSESS	SYSID	SYSTEM	TABLE
TAKEOVER	TCO	TDS	TER
TERM	TERMINAL	TERMINALS	TERMINATE
TERMS	TERS	TEST	THREAD
TIMEOUT	TIMEOVER	TIMO	TKO
TKOTRA	TKOVR	TMEM	TMEMBER
TO	TP	TPI	TPIPE
TPN	TPNAME	TRA	TRACE

Table 83. Words used and reserved only for IMS commands (continued)

TRACKING	TRACKING STATUS	TRAN	TRANAUTH
TRANCMD5	TRANCODE	TRANCOD5	TRANS
TRANSACTION	TRANSACTIONS	TRAP	TRKARCH
TRKAUTOARCH	TRS	TYPE	UDATA
UDS	UNITYPE	UNL	UNLOCK
UNPLAN	UOR	USED5DS	USER
USERD	USR5	VERIFY	VGR
VGR5	VID	VIR	VOLUME
VPL	VTAMP5OL	VUNLOAD	WADS
WPM1	WPM2	WPM3	XKEY
XTRC	XTRACE		

## Chapter 6. IMS type-1 commands that are valid in ETO

The commands that are valid for lterms, nodes, and users in ETO are listed.

The following table lists the type-1 commands that are valid for lterms, nodes, and users in ETO.

Table 84. IMS type-1 commands valid in ETO

Commands	Dynamic lterms that are valid	Dynamic nodes that are valid	Dynamic users that are valid
/ACTIVATE		X	
/ASSIGN	X <sup>1</sup>		X <sup>2</sup>
/BROADCAST	X	X	X
/CHANGE		X	X
/CLSDST		X	X
/COMPT		X	X
/DEQUEUE	X	X	X
/DISPLAY	X	X	X
/END			X
/EXCLUSIVE			X
/EXIT		X	X
/FORMAT	X		
/IDLE		X	
/LOCK	X	X	
/MODIFY	X		
/OPNDST		X	
/PSTOP	X		
/PURGE	X		
/QUIESCE		X	X
/RMxxxxxx	X		
/RSTART		X	X
/SET	X		
/SIGN			X
/START	X	X	X
/STOP	X	X	X
/TEST			X
/TRACE		X	
/UNLOCK	X	X	

**Notes:**

1. Only valid for /ASSIGN LTERM USER.
2. Only valid for /ASSIGN USER.



---

## Chapter 7. /ACTIVATE command

The /ACTIVATE command is used to activate VTAM nodes or MSC links with a VTAM "VARY NET, ACTIVE" command.

The /ACTIVATE command can also be used to undo the conditions set by the /IDLE command.

Subsections:

- "Environment"
- "Syntax"
- "Keywords"
- "Usage notes"
- "Examples" on page 98

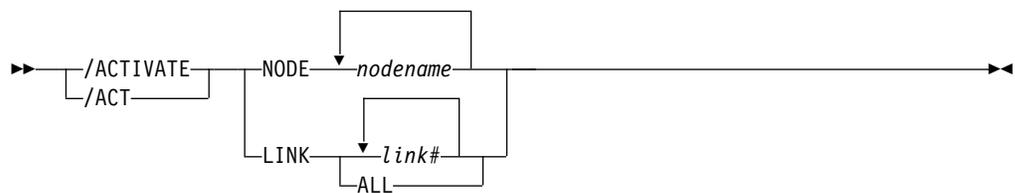
### Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

Table 85. Valid environments for the /ACTIVATE command and keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
/ACTIVATE	X		X
LINK	X		X
NODE	X		X

### Syntax



### Keywords

The following keywords apply to the /ACTIVATE command:

#### LINK

Specifies the MSC links to be activated.

#### NODE

Specifies the VTAM nodes to be activated.

### Usage notes

The /ACTIVATE command is used with the IMS VTAM I/O Timeout Detection Facility.

This command can be issued to an IMSplex using the Batch SPOC utility.

## **Examples**

The following are examples of the /ACTIVATE command:

### *Example 1 for /ACTIVATE command*

Entry ET:

```
/ACTIVATE LINK 1
```

Response ET:

```
DFS058I  ACTIVATE COMMAND COMPLETED
```

Explanation: Link 1 has been activated.

### *Example 2 for /ACTIVATE command*

Entry ET:

```
/ACTIVATE NODE ABC
```

Response ET:

```
DFS058I  ACTIVATE COMMAND COMPLETED
```

Explanation: Node ABC has been activated.

---

## Chapter 8. /ALLOCATE command

Use the /ALLOCATE command for delivering asynchronous output when the LU 6.2 device is available. The command can be originated by the remote LU 6.2 device for this purpose.

/ALLOCATE is a multisegment command similar to the /OPNDST command for VTAM terminals. It causes IMS to allocate a conversation to the specified LUNAME and TPNAME if any output is queued in IMS for that destination. If no such output exists, the command has no effect and (in a non-shared-queues environment) an error message is issued.

Subsections:

- "Environment"
- "Syntax"
- "Keywords"
- "Usage notes" on page 100

### Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

Table 86. Valid environments for the /ALLOCATE command and keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
/ALLOCATE	X		X
LUNAME	X		X
MODE	X		X
TPNAME	X		X

### Syntax

►► — /ALLOCATE — LUNAME — *luname* — TPNAME — *tpname* — MODE — *modename* — NONE — ◀◀

### Keywords

The following keywords apply to the ALLOCATE command:

#### LUNAME

Specifies the LU name of the LU 6.2 application program that is to be allocated. A network-qualified LU name is optional for the LUNAME keyword

#### MODE

Identifies the LOGON MODE table entry that VTAM uses to determine the session operating characteristics. NONE, which can be used as a MODE keyword parameter, resets the MODE field to its null state.

**TPNAME**

Specifies the TP name of the LU 6.2 application program that is to be allocated.

**Usage notes**

/ALLOCATE requires an end-of-message (EOM) indicator. An end-of-segment (EOS) indicator must be included for all segments preceding the last segment.

After the syntax checking of the /ALLOCATE command is successfully completed, the DFS058 ALLOCATE COMMAND COMPLETED message is issued and processing of the command continues asynchronously.

This command can be issued to an IMSplex using the Batch SPOC utility.

**Related reference:**

Chapter 2, "Command keywords and their synonyms," on page 63

---

## Chapter 9. /ASSIGN command

The /ASSIGN command alters the relationships between various IMS resources and resets certain values specified during IMS system definition.

Subsections:

- “Environment”
- “Syntax” on page 102
- “Keywords” on page 103
- “Usage notes” on page 111
- “/ASSIGN in an IMSplex” on page 112
- “Equivalent IMS type-2 commands” on page 114
- “Examples” on page 114

### Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

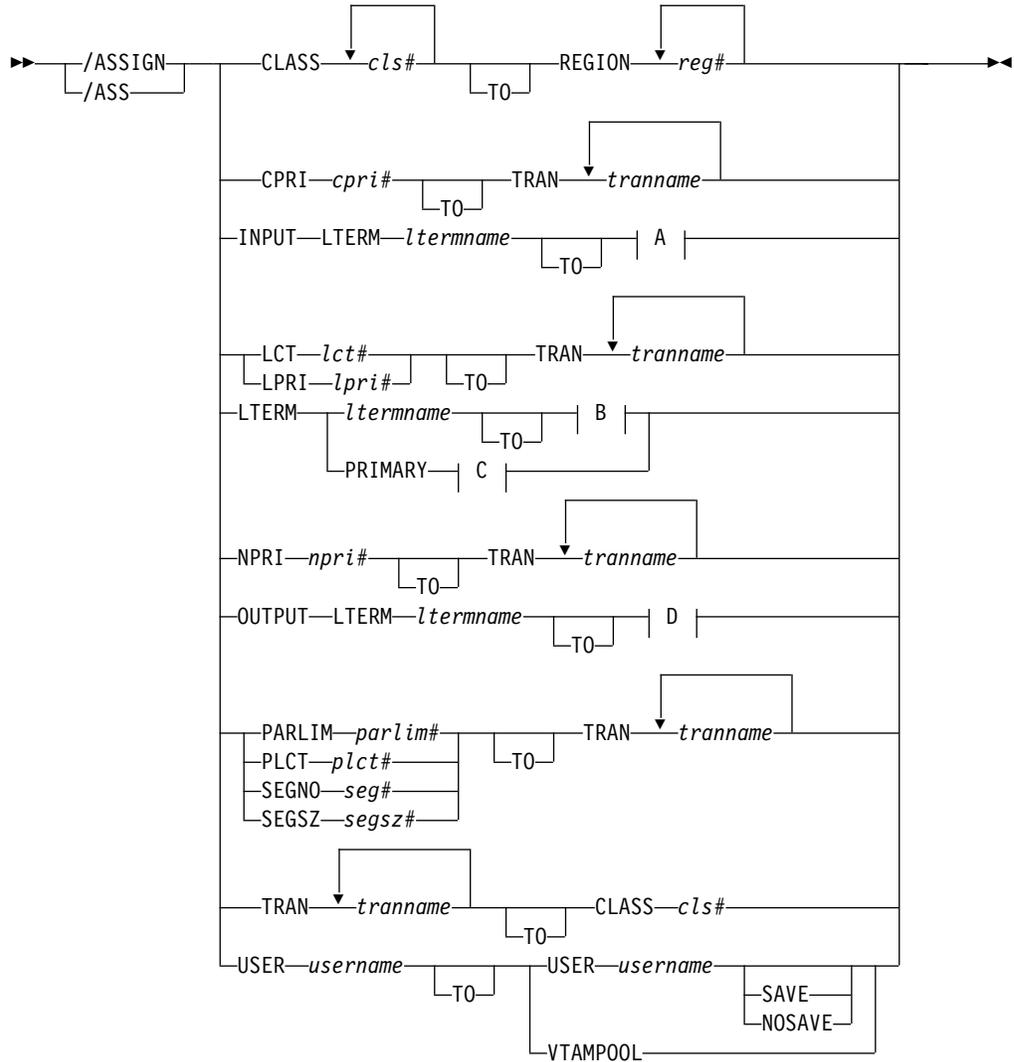
*Table 87. Valid environments for the /ASSIGN command and keywords*

Command / Keywords	DB/DC	DBCTL	DCCTL
/ASSIGN	X		X
CLASS	X		X
COMPONENT	X		X
CPRI	X		X
ICOMPONENT	X		X
INPUT	X		X
LINE	X		X
LCT	X		X
LPRI	X		X
LTERM	X		X
NODE	X		X
NOSAVE	X		X
NPRI	X		X
OUTPUT	X		X
PARLIM	X		X
PLCT	X		X
PTERM	X		X
REGION	X		X
SAVE	X		X
SEGNO	X		X
SEGSZ	X		X
TRAN	X		X

Table 87. Valid environments for the /ASSIGN command and keywords (continued)

Command / Keywords	DB/DC	DBCTL	DCCTL
USER	X		X
VTAMPOOL	X		X

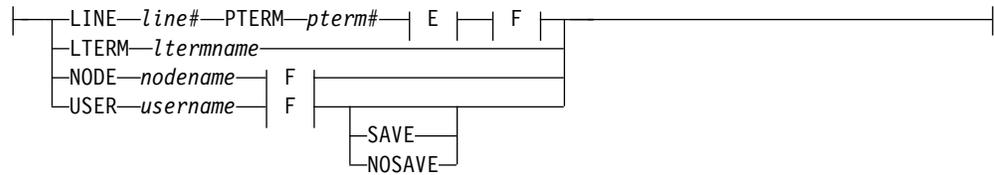
## Syntax



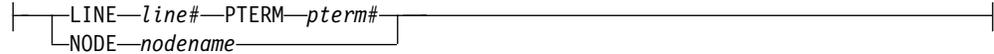
**A:**



**B:**



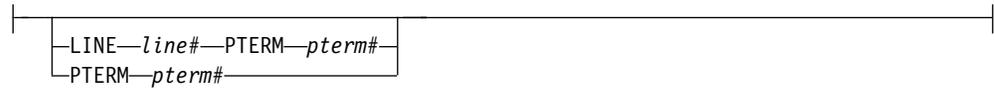
**C:**



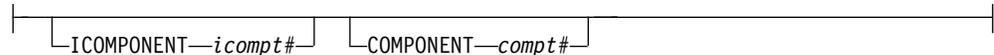
**D:**



**E:**



**F:**



## Keywords

The following keywords can be specified for the /ASSIGN command:

### CLASS

Specifies either the classes of transactions that a message processing region can schedule or the message processing class of a transaction.

The /ASSIGN CLASS cls# (TO) REGION reg# command can include up to four class parameters and replaces the class values that existed previously. Valid class parameters are numeric values from 1 to 999.

For static transactions, the CLASS parameter (cls#) is initially set by system definition using the TRANSACT macro statement. For CPI communications driven transactions, the initial value is obtained from the TP profile.

### COMPONENT

Allows output for a logical terminal to be directed to a specific component of a physical terminal. The COMPONENT parameter, which can have values 1, 2, 3 or 4, indicates the terminal component that should receive the output.

Parameter values other than 1 are valid for the 3275 and SLU 1 terminals, as well as type 1 and type P secondary logical units. When assigning a COMPONENT value from a terminal with components to a terminal without components, give the compt# a value of 1.

The following table shows the relationships between COMPONENT values and terminals. The first column lists the terminal, the second and third columns list the COMPONENT parameters that can have values of 1, 2, 3, or 4.

*Table 88. Relationships between COMPONENT/ICOMPONENT values and terminals*

<b>Terminal</b>	<b>1</b>	<b>2/3/4</b>
3275	Video	Printer
3770	Console/Printer	Defined in TERMINAL macro on logon descriptor during system definition.
SLU 1	Defined in TERMINAL macro on logon descriptor during system definition.	
SLU P terminal	Defined in TERMINAL macro on logon descriptor during system definition.	
SLU 4	Defined in TERMINAL macro during system definition.	

#### **CPRI**

Specifies a new value for the current priority of a transaction. The CPRI keyword is not allowed for BMP transactions, since BMP transactions should always have a priority of 0. The new CPRI value takes effect the next time the transaction is scheduled. Valid CPRI parameters are numeric values from 0 to 14.

#### **ICOMPONENT**

Specifies the input component of a logical terminal to be directed to a specific component of a physical terminal. The ICOMPONENT parameter, which can have values 1, 2, 3, or 4, indicates the terminal component that should receive the input. Parameter values other than 1 are valid for the 3275 and SLU 1 terminals, as well as type 1 and type P secondary logical units. When assigning from a terminal with components to a terminal without components, give icompt# a value of 1. See Table 88 for relationships between ICOMPONENT values and terminal components.

#### **INPUT**

Specifies that the input capability of a logical terminal is to be assigned to a given nonswitched physical terminal without affecting the output assignment. More than one physical terminal can be assigned as input for a logical terminal, but the logical terminal used must be the first logical terminal in a "chain."

#### **LINE**

Specifies the non-VTAM devices to which the input or output capabilities (or both) of a logical terminal are to be assigned. The physical terminal must not be on a switched line, in conversational, response, or preset destination modes.

#### **LCT**

Specifies a new value for the limit count of a transaction. The new LCT value takes effect during execution of the transaction. Valid LCT parameters are numeric values from 1 to 65535.

#### **LPRI**

Specifies a new value for the limit priority of a transaction. The LPRI keyword is not allowed for BMP transactions, since BMP transactions should always have a priority of 0. The new LPRI value takes effect the next time the transaction is scheduled. Valid LPRI parameters are numeric values from 0 to 14.

## LTERM

LTERM assigns the input and output capabilities of a local logical terminal to the given nonswitched physical terminal or terminals. The /ASSIGN LTERM command does not apply to Multiple Systems Coupling (MSC) remote logical terminals.

- If you specify /ASSIGN LTERM with LINE/PTERM selected, the first physical terminal (LINE/PTERM specification) is the input assignment. The second physical terminal (LINE/PTERM specification) is the output assignment. If there is no second physical terminal specification, then the first physical terminal specification is also the output assignment.
- If you specify /ASSIGN LTERM [TO] NODE, the node specification is the input and output assignment.

No response is sent to a terminal referenced in an /ASSIGN command. The logical terminal must not be an inquiry logical terminal and must not have queuing or dequeuing of messages in progress.

LTERMs created dynamically can be assigned to dynamically created users. Static logical terminals can be assigned to other static terminals. Dynamic logical terminals cannot be assigned to lines, static nodes, dynamic nodes, or static users. Static logical terminals cannot be assigned to dynamic nodes or users. /ASSIGN commands that split the logical terminal so that the input LTERM is associated with one node and the output is associated with another node are not allowed for any ACF/VTAM terminals.

**Physical terminal considerations:** The physical terminal named in the command or indirectly referred to by being associated with a logical terminal in the command must not be on a switched line, in conversational, response, or preset destination modes. If a user logical terminal is specified, no logical terminal in the subpool can be currently signed on, and the line and physical terminal for input and output must be the same. If message processing programs are dependent upon a particular LTERM-to-component assignment, this relationship must be maintained by the operator.

MSGDEL processing for an LTERM is based on how the MSGDEL option was defined:

- For static LTERMs defined during system definition as part of the VTAMPOOL for ISC support and for dynamic LTERMs, the MSGDEL options of the users must match when moving LTERMs between the users.
- For all other static LTERMs, the MSGDEL option is defined in the TERMINAL macro for the associated physical terminal.
- If a terminal's MSGDEL capabilities are defined in the TERMINAL macro as MSGDEL=NONIOPCB, assignment of an LTERM can take place only if the LTERM's message queues are empty, except when the LTERM is assigned to a terminal also defined as MSGDEL=NONIOPCB.
- If a terminal's MSGDEL capabilities are defined as MSGDEL=SYSINFO or NOTERM, assignment of an LTERM can take place only if the LTERM's system message queue is empty, except when the LTERM is assigned to a terminal defined as MSGDEL=SYSINFO or NOTERM, or MSGDEL=NONIOPCB.

**Master terminal considerations:** The primary and secondary master terminals cannot be assigned to a user, an input-only or output-only device, a terminal in response mode, an unattended type 1 secondary logical unit terminal, an ISC node, or NTO terminal, or dynamic node. The LINE/PTERM or NODE to

which the master terminal is to be assigned must be started and in an operable state. The line and physical terminal for input and output must be the same.

When a 3270 is designated as the master terminal during IMS system definition, two master logical terminals are generated. One, referred to as the primary master terminal, must be assigned to a 3270 display (3275/3276/3278/3279) for all input and output messages. The other, referred to as the secondary master terminal, must be assigned to a 3270 printer (3284/3286/3287/3288/3289) for certain IMS-selected output messages.

To move a single master logical terminal from one physical terminal to another physical terminal (for example, the secondary master from one 3284 to another 3284, or the primary master from one 3277 to another 3277) without moving the other, use the formats of /ASSIGN LTERM command with one LINE/PTERM or NODE, and specify the LTERM name of the master terminal.

You can use either of the following commands:

- /ASSIGN LTERM ltermname (T0) NODE nodename
- /ASSIGN LTERM ltermname (T0). LINE line# PTERM pterm#

**PRIMARY parameter:** The reserved parameter PRIMARY can be used with the LTERM keyword to change both the primary and secondary master terminals assignments to other physical terminals concurrently. The command formats used with the PRIMARY parameter are:

- /ASSIGN LTERM PRIMARY (T0) LINE line#1 PTERM pterm#1/NODE nodename#1

This command format assigns both the input and output of the primary master terminal to LINE line#1 PTERM pterm#1, and both the input and output of the secondary master terminal to NODE nodename.

- /ASSIGN LTERM PRIMARY (T0) (LINE line#1 PTERM pterm#1/NODE nodename1)(LINE line#2 PTERM pterm#2/NODE nodename2)

This command format assigns the input and output of the primary master terminal to the LINE line#1 PTERM pterm#1/NODE nodename #1 and assigns the input and output of the secondary terminal to LINE line#2 PTERM#2/NODE nodename#2.

These two formats can reassign:

- The 3270 primary and 3270 secondary master terminals to different 3270 physical terminals, where pterm#1 or nodename1 is a 3270 display and pterm#2 or nodename2 is a 3270 printer.
- A non-3270 master terminal to a 3270 display and 3270 printer, where pterm#1 or nodename1 is a 3270 display and pterm#2 or nodename2 is a 3270 printer.

The first LINE/PTERM or NODE specifies the display device to which the primary master logical terminal is to be assigned. The second LINE/PTERM or NODE specifies the printer device to which both the input and output capabilities of the secondary are to be assigned. If only one LINE/PTERM or NODE is specified, then the input and output capabilities of the primary master logical terminal and the secondary master logical terminal are assigned to the same LINE/PTERM or NODE.

**Data considerations:** IMS does not edit data destined for a logical terminal when an /ASSIGN LTERM or /ASSIGN USER command is executed that affects physical terminal to logical terminal relationships. Ensure that data that is to be sent to a given physical terminal is suitable for transmission to a different physical terminal. If the Message Format Service (MFS) is used and the FMT definition included the second device, the change of physical terminal from one MFS-supported device to another MFS-supported device will produce correct output.

The command format used with the /ASSIGN LTERM *ltermname* command is:

- /ASSIGN LTERM *ltermname* [TO] LINE *line#1* PTERM *pterm#1*

This command format assigns the input and output capabilities of the logical terminal to LINE *line#1* PTERM *pterm#1*. This command is not valid if the LTERM is dynamic.

```
/ASSIGN LTERM ltermname [TO] LINE line#1  
PTERM pterm#1 LINE line#2  
PTERM pterm#2
```

This command format assigns the input capability of the logical terminal to the first LINE/PTERM pair and assigns the output capability of the logical terminal to the second LINE/PTERM pair. This command is not valid if the LTERM is dynamic.

- /ASSIGN LTERM *ltermname* [TO] LINE *line#1* PTERM *pterm#1* PTERM *pterm#2*

This command format assigns the input capability of the logical terminal to LINE *line#1* PTERM *pterm#1* and assigns the output capability of the logical terminal to the same LINE, *line#1*, but to a different PTERM, *pterm#2*. This command is not valid if the LTERM is dynamic.

- /ASSIGN LTERM *ltermname* [TO] NODE *nodenamez*

This command format assigns the input and output capabilities of the logical terminal to NODE *nodename*. This command is not valid if the LTERM or NODE is dynamic.

- /ASSIGN LTERM *ltermname1* [TO] LTERM *ltermname2*

This command format allows an input chain to be moved to a symbolic location, rather than to a specific line and physical terminal. The [TO] line and terminal are determined by the current assignment of the specified [TO] logical terminal. With this format, the output assignment is not affected. This form of the /ASSIGN command is invalid for ISC nodes defined for parallel sessions, and it is not supported for LTERMs associated with VTAM terminals.

- /ASSIGN LTERM *ltermname* [TO] USER *username*

This command format assigns the logical terminal to the given user. When moving logical terminals between users, the MSGDEL options of the users as defined during system definition or on user descriptor must match. The user related to LTERM *ltermname* must not be signed on in conversation, in response mode, or in preset mode. For 3600/FINANCE, SLU P, and ISC, the user (*username*) can remain allocated because of terminal message resynchronization requirements. However, the user cannot be signed on because the associated terminal session must be stopped and idle. The user can be allocated, but, if allocated, must be associated with a session that is stopped and idle. This command will turn off the DEADQ status for the user associated with LTERM *ltermname*.

**NODE**

Specifies the VTAM terminal to which the input and output capabilities of a logical terminal are to be assigned. The node must not be dynamic, or in conversational, response, or preset destination modes. If a session with a VTAM terminal is terminated after a message is sent but before the response has been received, message resynchronization is necessary for this terminal. The output message for which no response was received must remain associated with this terminal until message resynchronization determines when the terminal received the message. If you use the /ASSIGN command to move the message to a different terminal, message resynchronization is no longer possible.

**NOSAVE**

Indicates that the assignment changes specified by the /ASSIGN command are lost when control blocks are deleted by the system when they are no longer needed. Omitting SAVE and NOSAVE causes the system-wide default to be used, as specified by the ASSNCHANGE parameter in the DFSDCxxx PROCLIB member.

**NPRI**

Specifies a new value for the normal priority of a transaction. The NPRI keyword is not allowed for BMP transactions, because BMP transactions should always have a priority of 0. The new NPRI value takes effect the next time the transaction is scheduled. Valid NPRI parameters are numeric values from 0 to 14.

**OUTPUT**

Specifies that the output capability of a logical terminal is to be assigned to a given nonswitched physical terminal, without affecting the input assignment. OUTPUT is invalid for ISC nodes defined for parallel sessions.

**PARLIM**

Specifies a new value for the parallel processing limit count of a transaction. parlim# is the maximum number of messages that can currently be queued, but not yet processed, by each active message region currently scheduled for this transaction. An additional region will be scheduled whenever the transaction queue count (for shared-queues environments, the successful consecutive GU count is used instead of the queue count) exceeds the PARLIM value multiplied by the number of regions currently scheduled for this transaction. Valid PARLIM parameters are numeric values from 0 to 32767 and 65535, where 65535 disables transaction load balancing.

A PARLIM of 65535 is the only valid value allowed for a transaction that is shown as eligible for load balancing but has an application program defined as SCHDTYPE=SERIAL. For example, if you dynamically change a parallel transaction to a serial transaction through online change, the only valid value for PARLIM is 65535.

/ASSIGN PARLIM is not valid for CPI Communications driven transactions.

**Note:** In a shared-queues environment, the PARLIM value behaves differently than it does in a non-shared-queues environment. In a non-shared-queues environment, the queue depth (the number of messages that are currently enqueued) for the transaction is used as the value that is compared with the PARLIM value to determine when to schedule another region. IMS responds to a growing queue of input transactions by scheduling more regions as the queue grows.

In a shared-queues environment, each individual IMS does not know the depth of the queue, because the queue is in the shared-queues coupling facility structure that is managed by Common Queue Server (CQS). The transaction queue might be added to by many different IMS systems. IMS is notified only when the first message is put in a queue (that is, when the queue becomes *not empty*). IMS is not notified for every subsequent message that is placed on the queue after that first message. In a shared-queues environment, the PARLIM comparison is done against a counter that each IMS keeps of the number of successful consecutive GU calls for the transaction by that IMS, rather than queue depth. IMS schedules more regions when it consistently gets messages from CQS when it asks for them. Thus, in a shared-queues environment, IMS infers the depth of the queue of messages based on processing activity, but it does not know the actual depth of the queue.

A PARLIM value of 0 in a shared-queues environment is the most responsive setting. PARLIM(0) ensures that message regions are scheduled until all messages are processed from the transaction queue, or until the maximum region value (MAXRGN) limit is reached. PARLIM(0) might, however, result in many unnecessary schedules (or *false schedules*). A false schedule occurs when a message region is scheduled and finds no more messages on the queue. This occurs particularly with PARLIM(0) because after each successful get unique (GU), IMS must always schedule an additional region to try to read the queue to see if there are more messages. This process continues for each successful GU until the queue becomes empty, at which time the successive GU count is reset to 0. This is a consequence of IMS not knowing how many messages are queued on the transaction queue.

Setting the PARLIM to a value greater than 0 can reduce the number of false schedules, because IMS then schedules a new message region only after a number of messages have been obtained consecutively without the queue becoming empty. Setting the PARLIM to a value of 2 or greater is useful for reducing false schedules for transactions that are low-volume and that run relatively quickly (such that the queue depth is typically 1), because it avoids scheduling a second region until the first region has obtained at least two messages in a row. However, be aware that while a PARLIM value greater than 0 can reduce unnecessary schedules, it is also less responsive. If a transaction is long running, or if its processing is delayed (for example, because of locking contention), the consecutive GU count does not change while the transaction is executing, and no additional message regions are scheduled. This can result in delayed processing of other messages for this same transaction until a currently-scheduled message completes. This delay can occur even if message regions are available to process the transaction.

#### **PLCT**

Specifies a new value for the processing limit count of a transaction. The PLCT is the number of messages of this transaction code that a program can process in a single scheduling. The new PLCT values take effect the next time the transaction is scheduled. Valid PLCT parameters are numeric values from 0 to 65535.

/ASSIGN PLCT is not valid for CPI Communications driven transaction programs.

**PTERM**

Specifies the non-VTAM device to which the input or output capabilities (or both) of a logical terminal are to be assigned.

**REGION**

Specifies the message processing region being assigned new classes of transactions that the region can schedule.

**SAVE**

Prevents deletion of user and LTERM control blocks across session and IMS restarts. Control blocks will be retained until the NOSAVE keyword is used with the /ASSIGN command. Omitting SAVE and NOSAVE causes the system-wide default to be used, as specified by the ASSNCHANGE parameter in the DFSDCxxx PROCLIB member.

**SEGNO**

Creates or changes the limit on the number of application program output segments allowed in message queues for each GU call. Segment limits will be established by transaction code, thereby allowing specification of more than one value for each application program. The new SEGNO value takes effect during execution of the transaction. Valid SEGNO parameters are numeric values from 0 to 65535.

The /ASSIGN SEGNO command is not valid for CPI Communications driven transaction programs.

**SEGSZ**

Creates or changes the limit on the size of application program output segments allowed in message queues for each GU call. The new SEGSZ value takes effect during execution of the transaction. Valid SEGSZ parameters are numeric values from 0 to 65535.

The /ASSIGN SEGSZ command is not valid for CPI Communications driven transaction programs.

**TRAN**

Specifies the transaction being assigned to a message processing class or being assigned a value. The new class takes effect the next time the transaction is scheduled.

**USER**

Assigns logical terminals to ISC half-sessions or to dynamic users. When moving logical terminals between users, the MSGDEL options of the users must match. The MSGDEL option for the static ISC users is defined in the SUBPOOL macro during system definition. The MSGDEL option for dynamic terminals is defined in the OPTIONS keyword of the ETO USER descriptor. /ASSIGN USER to VTAMP00L is not valid if USER is not ISC.

/ASSIGN LTERM ltermname (TO) USER username assigns the logical terminal to another USER. The user associated with LTERM ltermname must not be allocated. The USER username can be allocated, but it must be associated with a session that is stopped and idle, if allocated.

**Note:** Assigning an LTERM is not a permanent action. When the destination structure is deleted (for example, with /SIGN OFF) IMS discards information about the assignment. When the original dynamic user who owned that dynamic LTERM signs on again or gets a message switch, the dynamic LTERM is recreated for the original dynamic user.

If SAVE is specified, the dynamic user is not deleted unless another /ASSIGN command with the NOSAVE keyword is issued.

/ASSIGN USER username1 (TO) USER username2 assigns a string of logical terminals to another USER. The first user (username1) in the command must contain at least one logical terminal and must not be signed on, in conversation, in response mode, or in preset mode. For 3600/FINANCE, SLU P, and ISC, the second user (username2) can remain allocated due to terminal message resynchronization requirements. However, the user cannot be signed on and associated terminal sessions must be stopped and idle. This command turns off the DEADQ status for USER username1.

In addition to the ISC rules and restrictions, following dynamic terminal restrictions also apply. The /ASSIGN USER TO USER and /ASSIGN LTERM TO USER commands are rejected if the source or destination dynamic user is in conversation mode, response mode, or preset mode. The second USER can be allocated, but it must be associated with a session that is stopped and idle, if allocated.

When existing dynamic LTERMs or dynamic users are assigned to a destination user that does not exist, the destination user is created unless it is rejected by the DFSINSX0 user exit.

#### **VTAMPOOL**

Is valid only for VTAM ISC sessions. It is used to force a cold start of an ISC session that cannot perform a successful restart.

/ASSIGN VTAMPOOL deallocates an entire string of logical terminals allocated to a given USER. The user must not be signed on, in conversation mode, response mode, or preset mode, and the terminal must be stopped and idle. This command is valid for static and dynamic ISC users.

## **Usage notes**

Most changes made by the /ASSIGN command remain in effect until changed with another /ASSIGN command, or an IMS cold start (/NRESTART CHECKPOINT 0) is performed.

This command can be issued to an IMSplex using the Batch SPOC utility.

When dynamic LTERMs are assigned to dynamic users with the /ASSIGN LTERM TO USER command, or dynamic users are assigned to dynamic users with the /ASSIGN USER TO USER command, changes remain in effect only if the following conditions are true:

- The SAVE keyword is used.
- SAVE and NOSAVE keywords are omitted, but ASSNCHANGE=SAVE is specified in the DFSDCxxx PROCLIB member.

Otherwise, changes are lost when the destination user is deleted during user sign-off or IMS checkpoint.

Changes that affect regions, such as CLASS, are only in effect until the region terminates.

**Recommendation:** Use the /ASSIGN command consistently on all IMS subsystems that are members of a VTAM generic resources group or share IMS queues. For

example, assign an LTERM to a specific node for all IMS subsystems; assigning one LTERM to different nodes on different IMS subsystems could cause unpredictable results.

The /ASSIGN command is mirrored on an XRF alternate environment unless the assignment involves the primary or secondary master terminal, or the assignment of a CLASS to a REGION.

/ASSIGN commands attempting to assign LTERMs or users between static and dynamic terminals receive a unique error message. An error message is issued if /ASSIGN attempts any of the following

- To assign a dynamic LTERM to a static USER.
- To assign a static USER to a dynamic USER.
- To assign a dynamic USER to a static USER.
- To assign a dynamic LTERM to a LINE or NODE.
- To assign a static LTERM to a dynamic USER.

**Restriction:** You cannot use the /ASSIGN command to assign a user structure to a node that does not exist. Use the /OPN NODE nodename USER username command instead.

## **/ASSIGN in an IMSplex**

In an IMSplex, the /ASSIGN LTERM and /ASSIGN USER commands alter the relationships between one or more logical terminals and a physical terminal or user.

### *Static terminals*

For static terminals in an IMSplex, the control blocks should exist in all the IMS systems where a user could possibly log on to that terminal. If /ASSIGN LTERM TO NODE is specified in an IMSplex, it should be issued from the Single Point Of Control (SPOC) application. If ROUTE is specified, it should be specified as ROUTE(\*). Otherwise, the /ASSIGN command may not be processed consistently across the IMSplex and could cause unpredictable results. If one of the /ASSIGN commands fails, that IMS will send back a completion code to the Operations Manager (OM) indicating failure. It will be up to the IMS installation to correct that problem.

/ASSIGN LTERM TO NODE assigns the input and output capabilities of a logical terminal to a static VTAM node. The logical terminal may not be in conversational, response, or preset destination mode. The source and destination terminals do not need to be stopped or logged off.

If global resource information is not kept in the Resource Manager (RM) and the node exists locally, this command applies locally. If global resource information is kept in RM, this command applies globally, if the control blocks exist in RM. However, the assignment changes are not considered significant, and if the resource is deleted, the assignment changes are deleted as well. An LTERM that exists in the RM could be deleted if the static node to which it is assigned does not exist in the RM. If NODEA and LTERMA exist in the RM because the node is in exclusive mode, but static NODEB is not in RM, an /ASSIGN LTERM LTERMA NODE NODEB command would result in LTERMA being deleted in RM.

The IMS where the LTERM and node are active, or the command master if the resources are not active, will make the changes in the RM. In this case, "active" includes the case where the terminal is logged off, but has an RM affinity to a particular IMS. For /ASSIGN LTERM TO NODE, the LTERM and node might not be active in different IMS systems.

When the log on of a static node attempts to write its LTERM names to the RM, but one or more of the LTERM names are already in use, IMS will reject the log on. This error should occur only if there are inconsistent definitions, or there have been inconsistent /ASSIGN commands issued in the IMSplex.

### *Dynamic terminals*

/ASSIGN LTERM TO USER assigns the input and output capabilities of an ETO logical terminal to an ETO user or a static ISC logical terminal to a static ISC subpool. The logical terminal may not be in conversational, response, or preset destination mode. The source user associated with the ETO LTERM might not be signed on. The destination ETO user does not have to be stopped. If global resource information is not kept in RM and the user exists locally, this command applies locally. If global resource information is kept in RM, this command applies globally (if the control blocks exist in RM). However, the assignment changes are not considered significant, and if the resource is deleted, the assignment changes are deleted as well. The IMS where the LTERM and node are active, or the command master if the resources are not active, will make the changes in the RM.

Use /ASSIGN LTERM TO USER with the SAVE keyword in an RM environment, to cause the assignment to persist and to apply the change to future IMS systems that join the IMSplex. The SAVE keyword is valid only for dynamic terminals, so this persistence function is not available for static terminals. Use /ASSIGN LTERM TO USER with the NOSAVE keyword in an RM environment, to stop applying the change to future IMS systems that join the IMSplex.

/ASSIGN USER TO USER assigns all the logical terminals associated with the source user to the destination user. The destination user, if dynamic, does not have to be stopped. If global resource information is not kept in RM and the user exists locally, this command applies locally. If global resource information is kept in RM, this command applies globally. Use /ASSIGN USER TO USER with the SAVE keyword in an RM environment to cause the assignment to persist and to apply the change to future IMS systems that join the IMSplex. The SAVE keyword is valid only for dynamic terminals, so this persistence function is not available for static terminals. Use /ASSIGN USER TO USER with the NOSAVE keyword in an RM environment to stop applying the change to future IMS systems that join the IMSplex.

The /ASSIGN command is allowed when the destination user does not exist. However, when resource information is being kept in RM, unless the SAVE keyword is used, this command will not be allowed. Assignments are not considered significant without the SAVE keyword, and if the user does not already exist because of some other significant status, there is no reason to create the user.

IMS will dynamically create the source LTERM or user (if it does not exist) to enable the /ASSIGN command. If the destination also does not exist, the SAVE keyword must be specified on the /ASSIGN command. Otherwise, the SAVE keyword is optional.

### *Non-VTAM devices and VTAM*

IMS supports the assignment of an LTERM between non-VTAM devices and VTAM terminals. However, IMS will not save any non-VTAM status. For example, if an LTERM is assigned from VTAM to a non-VTAM device, the LTERM will be deleted from the RM, and any status that might have been associated with that LTERM will no longer be recoverable.

## Equivalent IMS type-2 commands

The following table shows variations of the /ASSIGN command and the IMS type-2 commands that perform similar functions.

Table 89. Type-2 equivalents for the /ASSIGN command

Task	/ASSIGN command	Similar IMS type-2 command
Changes the value for the limit count of a transaction.	/ASSIGN LCT <i>new_lmct_number</i> TO TRAN <i>traname</i>	UPDATE TRAN NAME( <i>traname</i> ) SET(LCT( <i>new_limit_count</i> ))
Changes the value for the limit priority of a transaction.	/ASSIGN LPRI <i>new_lpri_number</i> TO TRAN <i>traname</i>	UPDATE TRAN NAME( <i>traname</i> ) SET(LPRI( <i>new_limit_priority</i> ))
Changes the value for the normal priority of a transaction.	/ASSIGN NPRI <i>new_npri_number</i> TO TRAN <i>traname</i>	UPDATE TRAN NAME( <i>traname</i> ) SET(NPRI( <i>new_normal_priority</i> ))
Changes the value for the parallel processing limit count of a transaction.	/ASSIGN PARLIM <i>new_parlim_number</i> TO TRAN <i>traname</i>	UPDATE TRAN NAME( <i>traname</i> ) SET(PARLIM( <i>new_parallel_limit</i> ))
Changes the value for the processing limit count of a transaction.	/ASSIGN PLCT <i>new_plmct_number</i> TO TRAN <i>traname</i>	UPDATE TRAN NAME( <i>traname</i> ) SET(PLCT( <i>new_processing_limit</i> ))
Changes the limit on the number of application program output segments allowed in message queues for each GU call.	/ASSIGN SEGNO <i>new_segno_number</i> TO TRAN <i>traname</i>	UPDATE TRAN NAME( <i>traname</i> ) SET(SEGNO( <i>new_segment_number</i> ))
Creates or changes the limit on the size of application program output segments allowed in message queues for each GU call.	/ASSIGN SEGSZ <i>new_segsize_number</i> TO TRAN <i>traname</i>	UPDATE TRAN NAME( <i>traname</i> ) SET(SEGSZ( <i>new_segment_size</i> ))
Changes the class number of a transaction.	/ASSIGN TRAN <i>traname</i> TO CLS <i>new_class_number</i>	UPDATE TRAN NAME( <i>traname</i> ) SET(CLASS( <i>new_class_number</i> ))

## Examples

The following are examples of the /ASSIGN command:

### Example 1 for /ASSIGN command

Entry ET:

```
/ASSIGN CLASS 5 TO REGION 3
```

Response ET:

```
DFS058I ASSIGN COMMAND COMPLETED
```

Explanation: Class 5 is assigned to region 3. Class 5 transactions are scheduled into region 3. This command resets any previous class assignments to this region.

### Example 2 for /ASSIGN command

Entry ET:

```
/ASSIGN CLASS 4 6 2A TO REGION 5
```

Response ET:

```
DFS058I ASSIGN COMMAND COMPLETED EXCEPT CLASS 2A
```

Explanation: The requested assignment is complete except for 2A, which is an invalid class number.

*Example 3 for /ASSIGN command*

Entry ET:

```
/ASSIGN CPRI 8 TO TRAN PIT, SEED
```

Response ET:

```
DFS058I ASSIGN COMMAND COMPLETED
```

Explanation: A current priority of 8 is set for the transactions named PIT and SEED.

*Example 4 for /ASSIGN command*

Entry ET:

```
/ASSIGN INPUT LTERM JONES TO LINE 4 PTERM 3
```

Response ET:

```
DFS058I ASSIGN COMMAND COMPLETED
```

Explanation: Logical terminal JONES is assigned to LINE 4 PTERM 3 for input identification and security. It associates the chain of logical terminals, of which LTERM JONES will be chained in first, with LINE 4 PTERM 3 for input. LTERM JONES must not be in the interior of an input chain (no other logical terminal can point to it). A physical terminal can point (for input only) to the first logical terminal of any chain. The output physical terminal for LTERM JONES is not changed.

*Example 5 for /ASSIGN command*

Entry ET:

```
/ASSIGN LTERM APPLE TO LINE 5 PTERM 1
```

Response ET:

```
DFS058I ASSIGN COMMAND COMPLETED
```

Explanation: Logical terminal APPLE has both its input and output capabilities assigned to LINE 5 PTERM 1. The components present on LINE 5 PTERM 1 must be compatible with the physical terminal previously related to logical terminal APPLE.

*Example 6 for /ASSIGN command*

Entry ET:

```
/ASSIGN LTERM APPLE TO NODE JONES
```

Response ET:

```
DFS058I ASSIGN COMMAND COMPLETED
```

Explanation: Logical terminal APPLE has both its input and output capabilities assigned to node JONES. The components present on node JONES must be compatible with the physical terminal previously related to logical terminal APPLE.

*Example 7 for /ASSIGN command*

Entry ET:

```
/ASSIGN LTERM SMITH TO LINE 4 PTERM 6 PTERM 7 COMPONENT 2
```

Response ET:

```
DFS058I ASSIGN COMMAND COMPLETED
```

Explanation: Logical terminal SMITH is assigned to LINE 4 PTERM 6 for input and LINE 4 PTERM 7 for output. Output for LINE 4 PTERM 7 is directed to COMPONENT 2.

*Example 8 for /ASSIGN command*

Entry ET:

```
/ASSIGN LTERM X TO LINE 5 PTERM 7 PTERM 6 COMPONENT 4  
ICOMPONENT 3
```

Response ET:

```
DFS058I ASSIGN COMMAND COMPLETED
```

Explanation: Logical terminal X is assigned to LINE 5 PTERM 7 for input and to LINE 5 PTERM 6 for output. Input is only received from input component 3, while output is directed to component 4.

*Example 9 for /ASSIGN command*

Entry ET:

```
/ASSIGN LTERM JONES TO LINE 4 PTERM 6 LINE 9 PTERM 1
```

Response ET:

```
DFS058I ASSIGN COMMAND COMPLETED
```

Explanation: Logical terminal JONES is assigned to LINE 4 PTERM 6 for input capability and LINE 9 PTERM 1 for output capability. The component assignment is unaffected.

*Example 10 for /ASSIGN command*

Entry ET:

```
/ASSIGN LTERM SMITH NODE JONES
```

Response ET:

```
DFS058I ASSIGN COMMAND COMPLETED
```

Explanation: Logical terminal SMITH is assigned to node JONES for both input and output.

*Example 11 for /ASSIGN command*

Entry ET:

```
/ASSIGN LTERM BROWN TO LTERM WHITE
```

Response ET:

```
DFS058I ASSIGN COMMAND COMPLETED
```

Explanation: Logical terminal BROWN is removed from its present input chain (if one exists) of logical terminals and added to the end of the input chain (if one exists) of logical terminal WHITE. The output physical terminal for LTERM BROWN is not changed.

*Example 12 for /ASSIGN command*

Entry ET:

```
/ASSIGN LTERM LAX USER ILL ICOMPONENT 1 COMPONENT 2
```

Response ET:

```
DFS058I ASSIGN COMMAND COMPLETED
```

Explanation: Logical terminal LAX is reassigned from its existing user to the user ILL. The user associated with LAX cannot be allocated to an active session.

*Example 13 for /ASSIGN command*

Entry ET:

```
/ASSIGN LTERM PRIMARY TO LINE 4 PTERM 3
```

Response ET:

```
DFS058I ASSIGN COMMAND COMPLETED
```

Explanation: PTERM 3 on LINE 4 becomes the master terminal. PTERM 3 must be a 3270 display.

*Example 14 for /ASSIGN command*

Entry ET:

```
/ASSIGN LTERM PRIMARY TO NODE BOSS
```

Response ET:

```
DFS058I ASSIGN COMMAND COMPLETED
```

Explanation: The node, BOSS, becomes the primary master terminal and the secondary master terminal.

*Example 15 for /ASSIGN command*

Entry ET:

```
/ASSIGN LTERM PRIMARY TO LINE 1 PTERM 2 LINE 2 PTERM 4
```

Response ET:

```
DFS058I ASSIGN COMMAND COMPLETED
```

Explanation: PTERM 2 on LINE 1 becomes the primary master terminal and PTERM 4 on LINE 2 becomes the secondary master terminal. PTERM 2 is a 3270 display station and PTERM 4 is a 3270 printer.

*Example 16 for /ASSIGN command*

Entry ET:

```
/ASSIGN SEGNO 50 TO TRAN APPLE
```

Response ET:

```
DFS058I ASSIGN COMMAND COMPLETED
```

Explanation: A limit of 50 output segments is set for the transaction APPLE.

*Example 17 for /ASSIGN command*

Entry ET:

```
/ASSIGN SEGSZ 1000 TO TRAN APPLE
```

Response ET:

```
DFS058I ASSIGN COMMAND COMPLETED
```

Explanation: A maximum size of 1000 bytes is set for any one output segment of transaction APPLE.

*Example 18 for /ASSIGN command*

Entry ET:

```
/ASSIGN USER ILL TO USER CAL
```

Response ET:

```
DFS058I ASSIGN COMMAND COMPLETED
```

Explanation: All the logical terminals of user ILL are appended to the existing LTERM string of user CAL. Use of this form of the /ASSIGN command leaves ILL with no LTERMs and therefore unavailable for allocation to a session. Both user ILL and CAL cannot be allocated to a session.

*Example 19 for /ASSIGN command*

Entry ET:

```
/ASSIGN USER CAL TO VTAMPOOL
```

Response ET:

```
DFS058I ASSIGN COMMAND COMPLETED
```

Explanation: The entire allocated LTERM string defined for user CAL is reassigned to the VTAM pool. The node to which user CAL is allocated must not be in session and must be stopped and idle. User CAL is then available for allocation to any ISC session.

*Example 20 for /ASSIGN command*

Entry ET:

```
/ASSIGN TRAN APPLE TO CLASS 5
```

Response ET:

```
DFS058I  ASSIGN COMMAND COMPLETED
```

Explanation: The transaction named APPLE is assigned to class 5.

**Related concepts:**

 [CPI Communications driven application programs \(Communications and Connections\)](#)

**Related reference:**

 [UPDATE TRAN command \(Commands\)](#)



---

## Chapter 10. /BROADCAST command

The /BROADCAST command is a multisegment command used to send a message to terminals in one or more IMS systems.

Subsections:

- “Environment”
- “Syntax”
- “Keywords” on page 122
- “Usage notes” on page 123
- “Examples” on page 124

### Environment

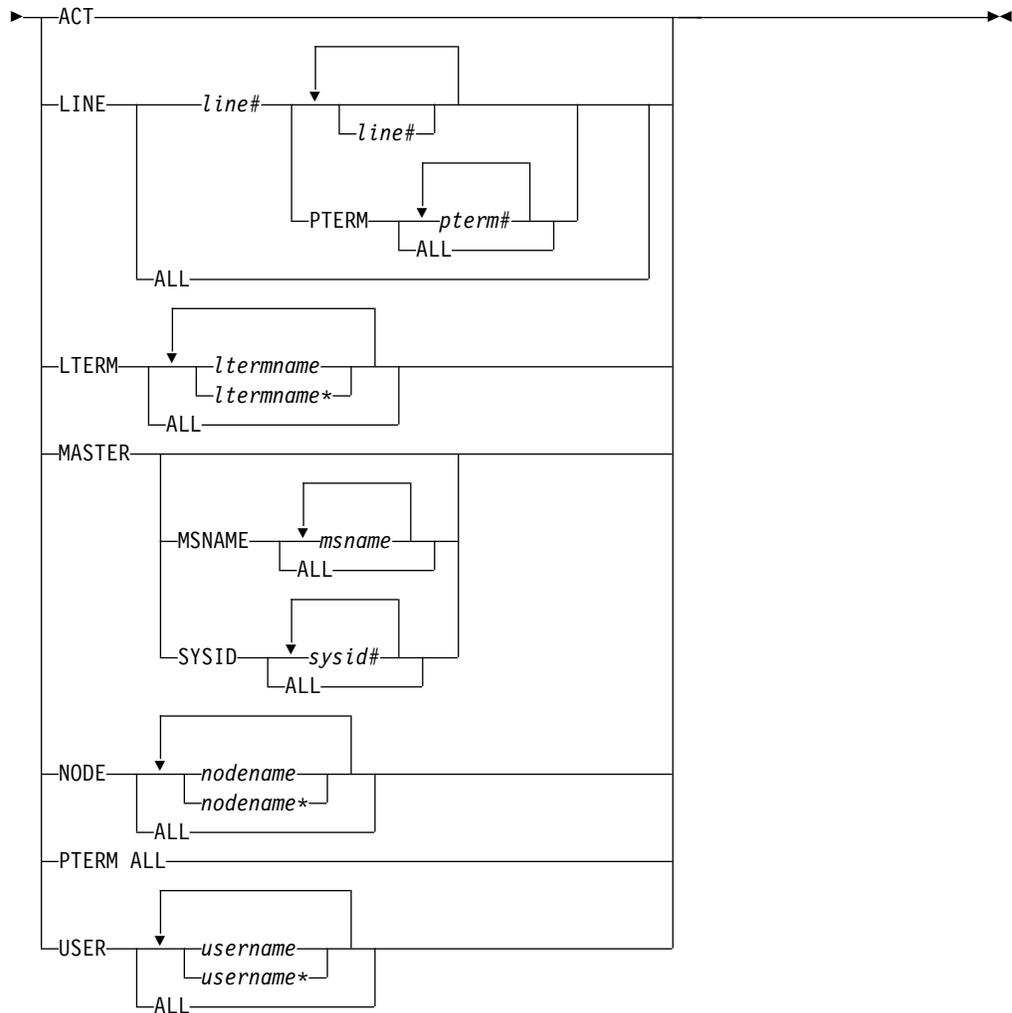
The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

*Table 90. Valid environments for the /BROADCAST command and keywords*

Command / Keywords	DB/DC	DBCTL	DCCTL
/BROADCAST	X		X
ACT	X		X
LINE	X		X
LTERM	X		X
MASTER	X		X
MSNAME	X		X
NODE	X		X
PTERM	X		X
SYSID	X		X
USER	X		X

### Syntax





## Keywords

The following keywords can be specified for the /BROADCAST command:

### ACT

Specifies that the supplied message is queued to the first LTERM allocated to each active node.

When operating on a dynamic terminal, the /BROADCAST ACT command only succeeds if a signed on user exists. A signed on user must exist, otherwise there is no destination to which to send the message.

### LINE

Specifies that a message is to be sent to all terminals associated with the specified line. PTERM specifies that the message is to be sent to specific terminals on the associated line.

### LTERM

Specifies that a message is queued to each named LTERM in the local system or any remote system. The remote logical terminal must be defined in the input system.

Where a LTERM does not exist, IMS attempts to create the LTERM and associated user structure if ETO is active. LTERM parameters can be generic, where the generic parameter specifies logical terminals that already exist.

When /BROADCAST LTERM ALL is specified, one copy of the message is queued for each logical terminal in the local system. In a multiple systems configuration, the message is not sent to remote systems when the ALL parameter is used. When more than one logical terminal is assigned to a physical terminal for output purposes, multiple copies of the message will result.

When the LTERM keyword specifies a logical terminal assigned to the VTAM pool, broadcast messages are queued for the first logical terminal in a subpool.

#### **MASTER**

Specifies that a message is to be sent to the IMS master terminal and to any specified secondary master terminal. Keywords SYSID and MSNAME can be used to further qualify the reserved parameter MASTER.

#### **MSNAME**

Specifies the logical link path in a multiple systems configuration.

#### **NODE**

Specifies that the supplied message is queued to the first output LTERM allocated to a terminal. If a terminal has no signed on user, no message can be queued. NODE parameters can be generic, where the generic parameter specifies nodes that already exist.

When operating on a dynamic terminal, the /BROADCAST NODE command only succeeds if a signed on user exists. A signed on user must exist, otherwise there is no destination to which to send the message.

#### **PTERM**

Specifies the physical terminal to which a message is to be sent.

#### **SYSID**

Specifies the system identification of a system in a multiple system configuration.

#### **USER**

Specifies that the supplied message is queued to the first LTERM associated with the dynamic user. The USER parameter can be generic. The /BROADCAST USER command applies only to existing dynamic users.

In an IMSplex, /BROADCAST USER queues the supplied message to the first LTERM associated with a dynamic user, if the user is signed on locally. /BROADCAST USER cannot be used to queue a message to an LTERM associated with a user signed on to another IMS in the IMSplex, or not signed on at all.

### **Usage notes**

For /BROADCAST commands entered by the master terminal operator, the multisegment input from this command is combined into 79-character segments for output transmission. The first input segment contains only the broadcast destination. The second and subsequent input segments must contain the data to be broadcast. Messages that are broadcast are sent even if the line, terminal, or both, are stopped at the time the broadcast is issued.

This command can be issued to an IMSplex using the Batch SPOC utility.

All /BROADCAST formats require an EOM indication to denote end-of-message; an EOS indication must be included for all segments that precede the last segment.

**Requirement:** When the /BROADCAST command is issued from an MCS/E-MCS console, OM API, or an AOI application, a period must appear as a delimiter between the command and the message text.

Unlike message switches, broadcast messages are sent to a terminal even when the terminal, associated line, or both, are not available (stopped, process stopped, or locked). However, broadcast messages are never sent to terminals if the MSGDEL parameter of the TERMINAL macro is specified MSGDEL=NONIOPCB.

Broadcast messages are always queued for logical terminals. When the ACTIVE, LINE, NODE, PTERM, or USER keywords are used, IMS queues the message for the first logical terminal found that is associated for output purposes with the specified line or physical terminal.

## Examples

The following are examples of the /BROADCAST command:

### *Example 1 for /BROADCAST command*

Entry ET:

```
/BROADCAST ACTIVE (EOS)
SYSTEM WILL BE SHUTDOWN FOR PM (EOS)
IN 5 MINUTES (EOM)
```

Response ET:

```
DFS058I BROADCAST COMMAND COMPLETED
```

Response RT:

```
SYSTEM WILL BE SHUTDOWN FOR PM IN 5 MINUTES
```

Explanation: The entered message is transmitted to all active terminals.

### *Example 2 for /BROADCAST command*

Entry ET:

```
/BROADCAST TO LTERM APPLE, TREE (EOS)
DON'T USE TRANSACTION GREENTRE UNTIL FURTHER (EOS)
NOTICE. (EOM)
```

Response ET:

```
DFS058I BROADCAST COMMAND COMPLETED
```

Response RT:

```
DON'T USE TRANSACTION GREENTRE UNTIL FURTHER
NOTICE.
```

Explanation: The entered message is transmitted to the logical terminals named APPLE and TREE.

### *Example 3 for /BROADCAST command*

Entry ET:

```
/BROADCAST TO LINE ALL (EOS)
SYSTEM WILL SHUTDOWN AT 5PM (EOM)
```

Response ET:

```
DFS058I BROADCAST COMMAND COMPLETED
```

Response RT:

```
SYSTEM WILL SHUTDOWN AT 5PM
```

Explanation: The entered message is transmitted to all physical terminals.

*Example 4 for /BROADCAST command*

Entry ET:

```
/BROADCAST TO LINE 13 PTERM ALL (EOS)
EXPECT DEMO YOUR LINE AT 9PM (EOM)
```

Response ET:

```
DFS058I BROADCAST COMMAND COMPLETED
```

Response RT:

```
EXPECT DEMO YOUR LINE AT 9PM
```

Explanation: The entered message is transmitted to all physical terminals on line 13.

*Example 5 for /BROADCAST command*

Entry ET:

```
/BROADCAST MASTER SYSID 2 (EOS)
SYSTEM WILL SHUTDOWN AT 5:00 PM (EOM)
```

Response ET:

```
DFS058I BROADCAST COMMAND COMPLETED
```

Response RT:

```
SYSTEM WILL SHUTDOWN AT 5:00 PM
```

Explanation: The message is transmitted to the master terminal of the system specified by the SYSID 2.

*Example 6 for /BROADCAST command*

Entry ET:

```
/BROADCAST MASTER MSNAME BOSTON, CHICAGO (EOS)
SYSTEM WILL SHUTDOWN AT 5:00 PM (EOM)
```

Response ET:

```
DFS058I BROADCAST COMMAND COMPLETED
```

Response RT:

```
SYSTEM WILL SHUTDOWN AT 5:00 PM
```

Explanation: The message is transmitted to the master terminals. of the remote systems specified by the MSNAMEs BOSTON and CHICAGO.

*Example 7 for /BROADCAST command*

Remote Terminal entry:

```
/FORMAT DFSMO4  
/BRO LTERM WTOR (eos)  
this is segment 1 (eos)  
this is segment 2 (eos)  
this is segment 3 (eom)
```

Response ET:

```
DFS058I BROADCAST COMMAND COMPLETED
```

Response RT:

```
THIS IS SEGMENT 1  
THIS IS SEGMENT 2  
THIS IS SEGMENT 3
```

Explanation: The remote terminal is first formatted by the /FORMAT command, where default format DFSMO4 supports the input of four segments. This is followed by /BROADCAST with four segments.

**Related reference:**

“Multisegment command input” on page 6

---

## Chapter 11. /CANCEL command

The /CANCEL command cancels all segments of a multisegment input message.

It must be entered prior to the end-of-message (EOM) indicator from the terminal that was entering the message.

Subsections:

- "Environment"
- "Syntax"
- "Usage notes"
- "Example"

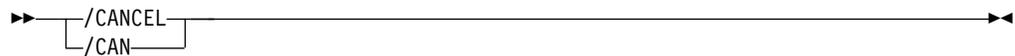
### Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) from which the command can be issued.

Table 91. Valid environments for the /CANCEL command

Command	DB/DC	DBCTL	DCCTL
/CANCEL	X		X

### Syntax



### Usage notes

Because a /CANCEL command must comprise a segment, it cannot be used to cancel a single-segment message.

On a non-3270 device, you can cancel a single-segment message by entering two asterisks (\*\*), followed immediately by an end-of-segment (EOS) indicator. When Message Format Service (MFS) is used, you can define delete characters other than (\*\*) to cancel other segments besides the first.

/CANCEL command has no meaning on display terminals where it is not possible to have some segments of a message already received by IMS while receiving subsequent segments.

### Example

Entry ET:

```
/BROADCAST TO ACTIVE (EOS)  
SYSTEM WILL BE AVAILABLE (EOS)
```

Entry ET:

```
/CANCEL
```

Response ET:

DFS058I CANCEL COMMAND COMPLETED

Explanation: All previously entered segments of the current message are discarded.

---

## Chapter 12. /CHANGE commands

The /CHANGE commands are multisegment commands used to change or delete internal resources within IMS.

All /CHANGE formats require an EOM indication to denote end-of-message; an EOS indication must be included for all segments that precede the last segment.

These commands can be issued to an IMSplex using the Batch SPOC utility.

Subsections:

- “/CHANGE APPC command”
- “/CHANGE CCTL command” on page 130
- “/CHANGE CPLOG command” on page 132
- “/CHANGE DESC command” on page 132
- “/CHANGE DIR MFS command” on page 134
- “/CHANGE FDR command” on page 135
- “/CHANGE LINK command” on page 136
- “/CHANGE NODE command” on page 140
- “/CHANGE SUBSYS command” on page 144
- “/CHANGE SURV command” on page 145
- “/CHANGE TRAN command” on page 148
- “/CHANGE UOR command” on page 149
- “/CHANGE USER command” on page 151

**Related reference:**

“Multisegment command input” on page 6

---

### /CHANGE APPC command

Use the /CHANGE APPC command to specify a change to the timeout value for APPC/MVS.

This value is set in the DFSDCxxx member of IMS.PROCLIB. IMS passes this value to APPC/z/OS for every implicit APPC/MVS conversation.

Subsections:

- “Environment”
- “Syntax” on page 130
- “Keywords” on page 130
- “Usage notes” on page 130

#### Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

Table 92. Valid environments for the /CHANGE APPC command and keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
/CHANGE	X	X	X
APPC	X		X
OUTBND	X	X	X
TIMEOUT	X	X	X

## Syntax



## Keywords

The following keywords are valid for the /CHANGE APPC command:

### OUTBND

Specifies a different outbound LU. The specified LU must be one of the APPC LUs defined in the APPCPMxx member of the SYS1.PARMLIB library. The default outbound LU is BASE LU. Message DFS182 is issued if the specified *luname* is not defined as a SCHEDULER in APPC/MVS for this IMS (in the APPCPMxx SYS1.PARMLIB member or through the SETAPPC command).

### TIMEOUT

Specifies the timeout value in minutes and seconds (*mmmm:ss*). *mmmm* can be a number between 0 and 1440. *ss* can be a number between 0 and 59. If the timeout value is 0, APPC/MVS timeout will be deactivated.

## Usage notes

A DFS34091 or DFS3491I message is issued after the /CHANGE APPC TIMEOUT command is issued to inform the operator of the new timeout value.

## /CHANGE CCTL command

Use the /CHANGE CCTL command to specify the coordinator control subsystem. The recovery elements are resolved by IMS.

Subsections:

- "Environment"
- "Syntax" on page 131
- "Keywords" on page 131
- "Examples" on page 131

## Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

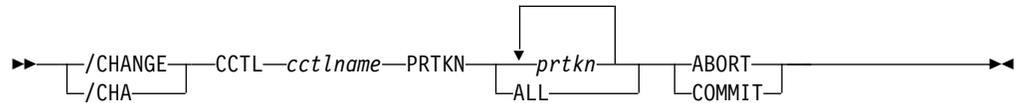
Table 93. Valid environments for the /CHANGE CCTL command and keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
/CHANGE	X	X	X

Table 93. Valid environments for the /CHANGE CCTL command and keywords (continued)

Command / Keywords	DB/DC	DBCTL	DCCTL
ABORT	X	X	
CCTL	X	X	
COMMIT	X	X	
PRTKN	X	X	

## Syntax



## Keywords

The following keywords are valid for the /CHANGE CCTL command:

*cctlname*

Specifies CCTL subsystem ID.

*prtkn*

Specifies the pseudorecovery token, which designates the unit of recovery to be aborted or committed. Use the /DISPLAY CCTL command to determine the name of the pseudorecovery token.

**ABORT**

Backs out changes for a unit of recovery. After completion of backout, the recoverable indoubt structure (RIS) is removed.

**COMMIT**

Commits changes for a unit of recovery. After the process is complete, the RIS is removed.

## Examples

This set of examples shows that an INDOUBT unit of recovery can be aborted if the INDOUBT status cannot be resolved. The /CHANGE ... PRTKN command backs out changes made to the database.

Entry ET:

```
/DISPLAY CCTL CICS1 IND
```

Response ET:

```

CCTL      PSEUDO-RTKN  RECOVERY-TOKEN  REGID  PSBNAME  STATUS
CICS1
           000100C0   9FFA956B7AE24E00  BMP255  INDOUBT
           00010040   9FFA9568FF594301  BMP255  INDOUBT
*90067/113446*
  
```

Entry ET:

```
/CHANGE CCTL CICS1 PRTKN 000100C0 ABORT (EOM)
```

Response ET:

DFS058I CHANGE COMMAND COMPLETED  
DBS0699I REYNC ABORT COMPLETE FOR PSB BMP255...

Explanation: The INDOUBT unit of recovery whose pseudorecovery token (PRTKN) is 000100C0 has been aborted.

---

## /CHANGE CPLOG command

Use the /CHANGE CPLOG command to change the value of the IMS execution parameter, CPLOG.

The IMS CPLOG execution parameter specifies the number of system log records between system-generated checkpoints.

Subsections:

- "Environment"
- "Syntax"
- "Keywords"

### Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

Table 94. Valid environments for the /CHANGE CPLOG command and keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
/CHANGE	X	X	X
CPLOG	X	X	X

### Syntax

►► /CHANGE CPLOG *cp\_log* ◀◀  
    └─/CHA─┘

### Keywords

The following keyword is valid for the /CHANGE CPLOG command:

*cp\_log*

This value must be specified as one or more numeric characters followed by either K or M. Values can range from 1K (1 KB) to 16M (16 MB).

---

## /CHANGE DESC command

Use the /CHANGE DESC command to specify the name of an LU 6.2 descriptor that will be updated with new values.

An error message is issued if the specified descriptor is not found.

Subsections:

- "Environment" on page 133
- "Syntax" on page 133
- "Keywords" on page 133

- “Usage notes” on page 134

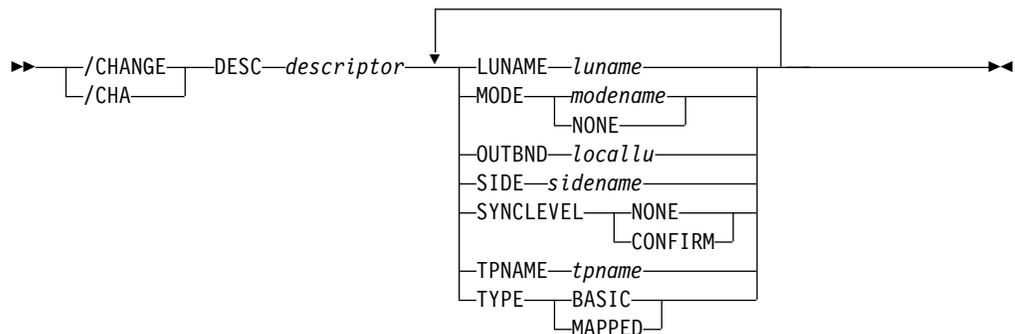
## Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

Table 95. Valid environments for the /CHANGE DESC command and keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
/CHANGE	X	X	X
DESC	X		X
LUNAME	X		X
MODE	X		X
OUTBND	X		X
SIDE	X		X
SYNCLEVEL	X		X
TPNAME	X		X
TYPE	X		X

## Syntax



## Keywords

The following keywords are valid for the /CHANGE DESC command:

### LUNAME

Specifies the LU name that is updated in the LU 6.2 descriptor.

The luname value in the descriptor is set to blanks unless the LUNAME keyword is also specified. A network-qualified LU name is optional for the LUNAME keyword.

### MODE

Specifies the VTAM mode table entry name that is updated in the LU 6.2 descriptor. The NONE parameter resets the mode field to its null state.

The MODE value in the descriptor is set to blanks unless the MODE keyword is also specified.

### OUTBND

Identifies a local LU to be used for outbound message processing. If you do

not specify a local LU, no LU name is displayed in the OUTBNDLU column of the /DISPLAY DESC command, and IMS uses the default LU.

**SIDE**

Specifies the APPC/z/OS side information that is set in the LU 6.2 descriptor. The side information contains default values for APPC conversation attributes such as LUNAME, TPNAME, and MODE.

**SYNCLEVEL**

Specifies the APPC sync level that is updated in the LU 6.2 descriptor. One of the following must be specified:

**CONFIRM**

IMS sync point processing continues.

**NONE**

IMS sync processing continues despite a session failure.

**TPNAME**

Specifies the tpname that is updated in the LU 6.2 descriptor. Message DFS182 is issued if the TPNAME parameter specified is DFSSIDE.

The TPNAME value in the descriptor is set to blanks unless the TPNAME keyword is also specified.

**TYPE**

Specifies the APPC conversation type that is updated in the LU 6.2 descriptor.

The conversation types are:

**BASIC**

Specifies that the data is to be formatted by the transaction programs, using the pattern "LL,data,LL,data".

**MAPPED**

Specifies that the data is to be formatted by APPC.

**Usage notes**

The /CHANGE DESC command changes the destination for future messages only. It does not change the destination for existing messages. The existing output messages are delivered only to the luname and tpname that had been previously specified for that message. This restriction is required by security requirements so message delivery to the intended destination only occurs at the time the messages are created.

---

**/CHANGE DIR MFS command**

Use the /CHANGE DIR MFS command to specify that the entries in the MFS dynamic directory are to be deleted.

This restores the dynamic directory to the original state that it was in just after IMS was initialized.

Subsections:

- "Environment" on page 135
- "Syntax" on page 135
- "Usage notes" on page 135

## Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

Table 96. Valid environments for the /CHANGE DIR MFS command and keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
/CHANGE	X	X	X
DIR	X		X

## Syntax

►► — [ /CHANGE — ] — DIR MFS —►►  
      [ /CHA — ]

## Usage notes

When IMS is initialized, IMS creates an MFS block primary directory based on the contents of all the \$\$IMSDIR members found. At the same time, IMS issues GETMAINS to acquire additional storage that is large enough to hold approximately 10% of the members in the active format library. While IMS is running, entries are only added to the dynamic directory, they are not deleted. Therefore, the dynamic directory continues to get larger until it runs out of space. When this happens, IMS can extend the amount of space in the dynamic directory to hold another 10%. IMS can extend the size of the dynamic directory a maximum of 9 times.

If you want to delete the in-storage (or index) entries from the dynamic directory without restarting IMS or performing an online change for the format library, use the /CHANGE DIR MFS command.

---

## /CHANGE FDR command

Use the /CHANGE FDR command to specify a change to the timeout value for IMS Fast Database Recovery surveillance.

This value is set in the DFSFDRxx member of IMS.PROCLIB. IMS uses this value to determine how long to wait before initiating a Fast Database Recovery takeover.

Subsections:

- “Environment”
- “Syntax” on page 136
- “Usage notes” on page 136

## Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

Table 97. Valid environments for the /CHANGE FDR command and keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
/CHANGE	X	X	X

Table 97. Valid environments for the /CHANGE FDR command and keywords (continued)

Command / Keywords	DB/DC	DBCTL	DCCTL
FDR	X	X	
TIMEOUT	X	X	X

## Syntax

►► /CHANGE FDR TIMEOUT #second  
 [ /CHA ]

## Usage notes

The timeout value (*#seconds*) must be at least 3 but no greater than 999.

IMS rejects this command if the active IMS subsystem is not connected to a Fast Database Recovery region.

## /CHANGE LINK command

Use the /CHANGE LINK command with FORCSESS, SYNCSESS, COLDSESS to override the system definition option defined for forcing resynchronization until the next /CHANGE LINK command or IMS cold start.

The /CHANGE LINK command applies only to IMS systems that are linked by Multiple Systems Coupling (MSC) when TCP/IP or VTAM is used.

Subsections:

- "Environment"
- "Syntax" on page 137
- "Keywords" on page 137
- Examples

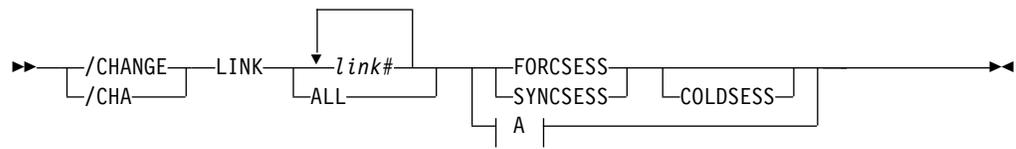
## Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

Table 98. Valid environments for the /CHANGE LINK keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
/CHANGE LINK	X		X
ASR	X		X
COLDSESS	X		X
FORCSESS	X		X
MODE	X		X
SYNCSESS	X		X

## Syntax



**A:**



## Keywords

The following keywords are valid for the /CHANGE LINK command:

### FORCSESS

Defines the link as capable of being warm started from an ERE state, even if the message sequence numbers on each side of the link are not synchronized.

**Attention:** The use of FORCSESS can cause messages to be lost. Specify FORCSESS for a link only if the loss of messages during a restart can be tolerated.

### SYNCSESS

Defines the link as incapable of being warm started from an ERE state, unless the message sequence numbers on each side of the link are synchronized.

### COLDSESS

Resets a link that is in either ERE or NRE state to COLD state, the state of a link when IMS comes up with a cold start.

The FORCSESS or the SYNCSESS keyword must be specified with the COLDSESS keyword. The specification of FORCSESS or SYNCSESS keyword with COLDSESS overrides the previous specification of FORCSESS or SYNCSESS for the link.

**Attention:** Use of the COLDSESS keyword can cause messages to be lost. Use the COLDSESS keyword only if one of the systems associated with an MSC link goes down, and the only way to bring up the session is to cold start it.

The COLDSESS keyword can be specified to terminate control blocks associated with the specified link or links and reset the link to COLD. COLDSESS should be used only after the /PSTOP command is issued and completes against the link or links and the links appear to be hung. If COLDSESS is specified, it should be used on both sides of the link or links. Gather documentation to determine why the link or links did not come down normally.

### ASR

Changes the automatic keyword session restart designation of a VTAM link. The default parameter for ASR is ON.

Automatic session restart is not necessarily activated for a link just because a status of ASR is displayed for that link. You must also have coded SONSCIP=YES on the APPL definition statement for VTAM when defining the network for the VTAM.

ASR applies only to VTAM MSC links. Attempting to turn ASR ON for any other type of MSC link is invalid. If non-VTAM MSC links are specified on the /CHANGE LINK ASR command, the command response indicates that the command processing completed with exceptions.

#### MODE

Changes the default mode table name of a VTAM link. This default is usually established by system definition. Parameter NONE resets this field to its null state (as if no mode table name was specified at system definition).

MODE applies only to VTAM MSC links. Changing the mode table name for any other type of MSC link is invalid. If non-VTAM MSC links are specified on the /CHANGE LINK MODE command, the command response indicates that the command processing completed with exceptions.

### Examples: FORCSESS, SYNCSESS, and COLDSESS

The following series of command examples illustrate the use the /CHANGE LINK command with the FORCSESS, SYNCSESS, and COLDSESS keywords. In all of the examples in this series, LINK 22 is an MSC TCP/IP link.

In the following example, LINK 22 is displayed. The link has not been used yet and is in a PSTOPPED IDLE COLD state.

```
/DISPLAY LINK 22
```

```
DFS000I LINK PARTNER RECD ENQCT DEQCT QCT SENT  
DFS000I 22 TA 0 0 0 0 PSTOPPED IDLE COLD
```

In the following example, /CHANGE LINK 22 FORCSESS is issued to define the link as capable of being restarted when the message sequence numbers are not synchronized.

```
/CHANGE LINK 22 FORCSESS.
```

```
DFS058I 16:58:59 CHANGE COMMAND COMPLETED
```

In the following example, the /DISPLAY LINK command shows that the status of the link now includes FORCE, which indicates that the FORCSESS attribute is set on the link.

```
/DISPLAY LINK 22
```

```
DFS000I LINK PARTNER RECD ENQCT DEQCT QCT SENT  
DFS000I 22 TA 0 0 0 0 PSTOPPED IDLE COLD FORCE
```

In the following example, the /RSTART command starts the link and the /DISPLAY command shows that the status of the link is now IDLE ACTV PRI FORCE.

```
/RSTART LINK 22
```

```
DFS058I 17:02:03 RSTART COMMAND COMPLETED
```

```
DFS2168I 17:02:05 CONNECTION ESTABLISHED ON LINK 22
```

```
/DISPLAY LINK 22
```

```
DFS000I LINK PARTNER RECD ENQCT DEQCT QCT SENT
DFS000I 22 TA          0    0    0    0    0 IDLE ACTV PRI FORCE
```

In the following example, the /PSTOP command is issued with the FORCE keyword, which stops the link. Because LINK 22 is a TCP/IP link, the link does not need to be shut down before issuing the /PSTOP command with the FORCE keyword.

After the /PSTOP command is processed, the reason code and LOSTSESS value in message DFS3177E indicate why the link stopped.

```
/PSTOP LINK 22 FORCE
```

```
DFS058I 17:08:01 PSTOP COMMAND COMPLETED
```

```
DFS3177E 17:08:01 MSC DETECTED AN ERROR,
DFS000I SENDMSG , RETCODE = 00000000, RSNCODE = 00000070, LOSTSESS = FORCESTO
DFS000I MODULE = DFSTC7C0, LINK = 022, LNK12T01
```

```
DFS2169I 17:08:01 DISCONNECTION COMPLETED ON LINK 0022
```

After the disconnection of LINK 22 is complete, a /DISPLAY LINK command shows the status of LINK 22 as PSTOPPED IDLE ERE FORCE IMS1. In the link status,

- ERE indicates that the link is in a warm state and that when the link is restarted, emergency restart synchronization will be performed.
- FORCE indicates that the link will start even if the message sequence numbers are not synchronized.
- IMS1 identifies the IMS system in which the link was stopped.

```
/DISPLAY LINK 22
```

```
DFS000I LINK PARTNER RECD ENQCT DEQCT QCT SENT
DFS000I 22 TA          0    0    0    0    0 PSTOPPED IDLE ERE FORCE IMS1
```

In the following example, the /CHANGE LINK command is issued with the SYNCSESS COLDSESS keywords. The COLDSESS keyword resets the link status from ERE to COLD. The SYNCSESS keyword changes FORCSESS attribute of the link to the SYNCSESS attribute, which redefines the link as requiring the message sequence numbers on each side of the link to be synchronized before the link can start.

```
/CHANGE LINK 22 SYNCSESS COLDSESS.
```

```
DFS058I 17:08:45 CHANGE COMMAND COMPLETED
```

In the following final example of this series, the /DISPLAY LINK command shows that the link status no longer includes FORCE, which indicates that the FORCSESS attribute is no longer set and that the message sequence numbers must be synchronized before the link will start from a warm state.

```
/DISPLAY LINK 22
```

```
DFS000I LINK PARTNER RECD ENQCT DEQCT QCT SENT
DFS000I 22 TA          0    0    0    0    0 PSTOPPED IDLE COLD
```

## Examples: ASR

This set of commands illustrates how the ASR setting of a link can be modified by the /CHANGE command:

Entry ET:

/DISPLAY LINK 6

Response ET:

LINK PARTNER	RECD	ENQCT	DEQCT	QCT	SENT				
6 AL	0	0	0	0	0	ASR	PSTOPPED	IDLE	COLD
*90179/102004*	IMSA								

Explanation: Automatic Session Restart was defined for link 6 by system definition.

Entry ET:

/CHANGE LINK 6 ASR OFF (EOM)

Response ET:

DFS058I CHANGE COMMAND COMPLETED

Entry ET:

/DISPLAY LINK 6

Response ET:

LINK PARTNER	RECD	ENQCT	DEQCT	QCT	SENT				
6 AL	0	0	0	0	0	PSTOPPED	IDLE	COLD	
*90179/102126*									

Explanation: Automatic Session Restart is not available for link 6

Entry ET:

/CHANGE LINK 6 ASR (EOM)

Response ET:

DFS058I CHANGE COMMAND COMPLETED

Explanation: The ASR parameter will default to ON.

Entry ET:

/DISPLAY LINK 6

Response ET:

LINK PARTNER	RECD	ENQCT	DEQCT	QCT	SENT				
6 AL	0	0	0	0	0	ASR	PSTOPPED	IDLE	COLD
*90179/102300*									

Explanation: Automatic Session Restart is in effect again as a result of the second /CHANGE command.

---

## /CHANGE NODE command

Use the /CHANGE NODE command to specify a VTAM node to be changed. The NODE parameter can be generic if the USER keyword is not present. The generic parameter specifies nodes that already exist.

Subsections:

- "Environment" on page 141
- "Syntax" on page 141
- "Keywords" on page 141

- “Examples” on page 143

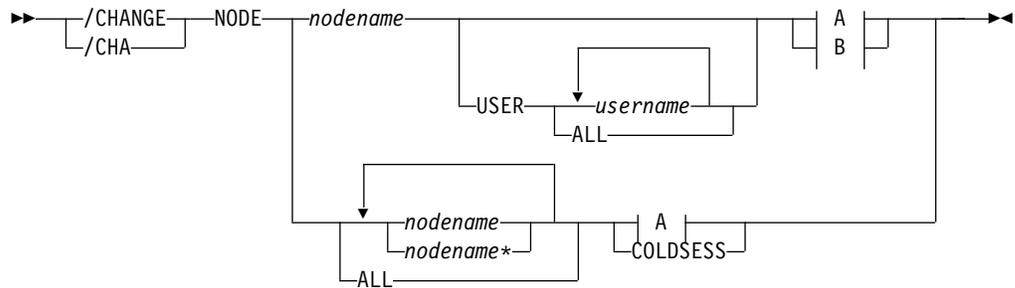
## Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

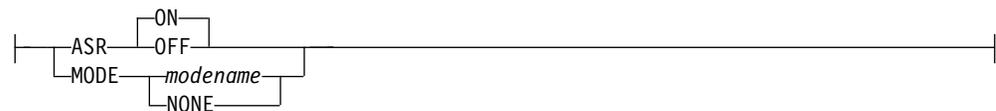
Table 99. Valid environments for the /CHANGE NODE command and keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
/CHANGE	X	X	X
ASR	X		X
COLDSESS	X		X
FORCSESS	X		X
MODE	X		X
NODE	X		X
SYNCSESS	X		X
USER	X		X

## Syntax



**A:**



**B:**



## Keywords

The following keywords are valid for the /CHANGE NODE command:

### ASR

The ASR keyword allows you to change the automatic session restart designation of a node. The default parameter for ASR is ON.

Automatic session restart is not necessarily activated for a node just because a status of ASR is displayed for that node. You must also have coded SONCSIP=YES on the APPL definition statement for VTAM when defining your network.

### **COLDSESS**

When COLDSESS is used with the NODE keyword, it sets up the SLU P or FINANCE session so that the session can be cold started. The COLDSESS keyword should be used if the SLU P or FINANCE session has experienced problems and attempts to warm start the session fail. The COLDSESS keyword terminates terminal and user control blocks associated with the specified node or nodes, and resets the node status to COLD.

In an IMSplex, if global resource information is not kept in Resource Manager (RM), the change is applied locally. If global resource information is kept in RM, the change is applied globally.

A DFS0581 COMMAND COMPLETE EXCEPT message may be received if the node is temporarily in use by another task even if the criteria (node is terminated and idle) for successful completion of the command is met.

### **FORCSESS, SYNCSESS**

The FORCSESS and SYNCSESS keywords are valid only for ISC nodes. Specify FORCSESS and SYNCSESS to override the system definition or logon descriptor option defined to force or not force synchronization of sessions. This override is effective until the next /CHANGE command is issued or an IMS cold start is effected.

### **MODE**

Changes the default mode table name of a node. This default is usually established by system definition or logon descriptor. MODE resets this field to its null state (as if no mode table name had been specified at system definition).

Changing ASR or the mode table name for VTAM 3270 nodes is invalid. If nodes of this type are referred to specifically in the /CHANGE ASR or /CHANGE MODE commands, they are marked in error.

### **USER**

Indicates the ISC user that is allocated to the node or the dynamic user that is signed on to the node.

/CHANGE USER AUTOLOGON SAVE changes the autologon information previously specified by a user descriptor, Destination Creation exit routine (DFSINSX0), or signon exit routine (DFSSGNX0). The autologon change is saved. The autologon information includes the node name, the mode table, the logon descriptor, and the ID. If the Resource Manager is active, the user will be dynamically created if it does not already exist. The SAVE keyword must be specified. If SAVE is not specified, the command is rejected with message DFS1199.

#### **Restrictions for using NODE and USER parameters together:**

- Commands with the NODE USER keyword pair are valid only if:
  - The USER is signed on to the NODE
  - In an ISC environment, the USER is allocated to the NODE
  - The nodes and users already exist
- /CHANGE NODE USER commands with ASR or MODETABLE keyword are valid for ISC and non-ISC nodes and users.

- /CHANGE NODE USER commands with FORCSESS or SYNCSESS keyword are valid for ISC nodes only.

## Examples

The following are examples of the /CHANGE NODE command:

### Example 1

Entry ET:

```
/DISPLAY NODE LUTYPEP1 MODE
```

Response ET:

```
NODE-USR  TYPE  DEF MODETBL  ACT MODETBL
LUTYPEP1  SLUP  DEFRESP
*90179/100630*
```

Explanation: DEFRESP is the mode table name defined for node LUTYPEP1 at system definition or logon descriptor or resource creation. The session is not active so the active mode table field (ACT MODETBL) is blank.

Entry ET:

```
/CHANGE NODE LUTYPEP1 MODE XXXXXXXX (EOM)
```

Response ET:

```
DFS058I CHANGE COMMAND COMPLETED
```

Entry ET:

```
/DISPLAY NODE LUTYPEP1 MODE
```

Response ET:

```
NODE-USR TYPE  DEF MODETBL  ACT  MODETBL
LUTYPEP1 SLUP  XXXXXXXX
*84179/100733*
```

Explanation: The default mode table name has been changed to XXXXXXXX by the previous /CHANGE command. The session is still not active so the active mode table field (ACT MODETBL) is blank.

### Example 2

Entry ET:

```
/DISPLAY NODE LUTYPEP
```

Response ET:

```
NODE-USR TYPE  CID      RECD  ENQCT  DEQCT  QCT  SENT
LUTYPEP  SLUP  00000000  37    37    37    0    37  IDLE
*98276/153630*
```

Explanation: Node LUTYPEP is terminated warm after session received and processed 37 messages.

Entry ET:

```
/CHANGE NODE LUTYPEP COLDSESS (EOM)
```

Response ET:  
DFS058I CHANGE COMMAND COMPLETED

Entry ET:  
/DISPLAY NODE LUTYPEP

Response ET:  

NODE-USR	TYPE	CID	RECD	ENQCT	DEQCT	QCT	SENT		
LUTYPEP	SLUP	00000000	0	0	0	0	0	IDLE	COLD

 \*98279/153630\*

Explanation: Terminal and user blocks associated with node LUTYPEP have been cleared and the status has been reset to COLD.

## /CHANGE SUBSYS command

Use the /CHANGE SUBSYS command to specify the subsystem name from which IMS recovery elements are to be deleted.

Subsections:

- "Environment"
- "Syntax"
- "Keywords"
- "Examples" on page 145

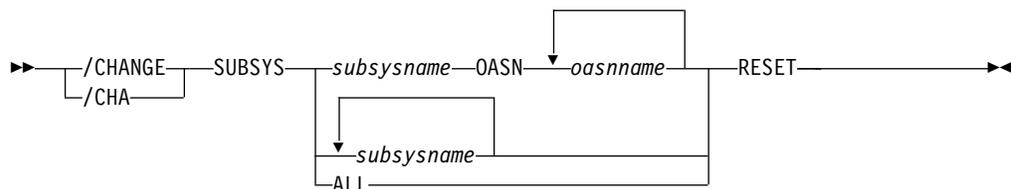
### Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

Table 100. Valid environments for the /CHANGE SUBSYS command and keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
/CHANGE	X	X	X
OASN	X	X	X
RESET	X	X	X
SUBSYS	X	X	X

### Syntax



### Keywords

The following keywords are valid for the /CHANGE SUBSYS command:

**OASN**

Specifies the outstanding recovery element to be deleted.

**RESET**

Causes an incomplete unit of work assigned to an external subsystem (not a CCTL subsystem) to be deleted.

**Examples**

The following are examples of the /CHANGE SUBSYS command:

*Example 1*

Entry ET:

```
/CHANGE SUBSYS DSN RESET (EOM)
```

Response ET:

```
DFS058I CHANGE COMMAND COMPLETED
```

Explanation: Reset all IN-DOUBT recovery units for subsystem DSN.

*Example 2*

Entry ET:

```
/CHANGE SUBSYS ALL RESET (EOM)
```

Response ET:

```
DFS058I CHANGE COMMAND COMPLETED
```

Explanation: Reset all IN-DOUBT recovery units for all subsystems.

*Example 3*

Entry ET:

```
/CHANGE SUBSYS ABCD OASN 99 685 2920 RESET (EOM)
```

Response ET:

```
DFS058I CHANGE COMMAND COMPLETED
```

Explanation: Reset IN-DOUBT recovery units identified by OASN numbers 99, 685, 2920 for subsystem ABCD.

---

**/CHANGE SURV command**

Use the /CHANGE SURV command to specify that the interval or timeout value on the active or alternate system in an XRF environment is changed.

Subsections:

- "Environment" on page 146
- "Syntax" on page 146
- "Keywords" on page 146
- "Usage notes" on page 146
- "Examples" on page 147

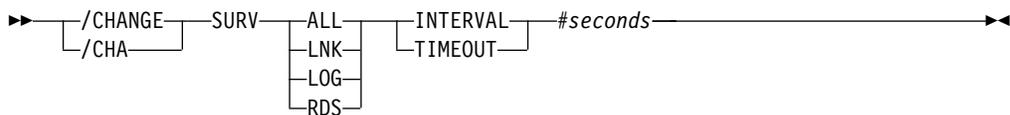
## Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

Table 101. Valid environments for the /CHANGE SURV command and keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
/CHANGE	X	X	X
INTERVAL	X		X
SURV	X		X
TIMEOUT	X	X	X

## Syntax



## Keywords

The following keywords are valid for the /CHANGE SURV command:

### ALL

Same as specifying LNK, LOG, and RDS.

### INTERVAL

Specifies the new interval on the active or alternate system in an XRF environment. The INTERVAL keyword identifies the new value in seconds and must be in the range 1 through 99.

### LNK

IMS ISC link.

### LOG

IMS system log.

### RDS

IMS restart data set.

### TIMEOUT

Specifies the new timeout value on the active or alternate system in an XRF environment. The TIMEOUT keyword identifies the new value in seconds and must be in the range 1 through 99.

## Usage notes

When /CHANGE SURV is entered on the active system, it becomes effective on both the active and alternate systems if the alternate system is up. If entered on the alternate system, only the alternate system is changed.

**Recommendation:** Change surveillance on the active system and allow the alternate system to take affect.

Additional considerations when changing the TIMEOUT values are shown in the following table. Listed in the table are the TIMEOUT value requirements for certain systems where the command is entered on and the action taken for exceptions.

*Table 102. Changing the TIMEOUT value*

<b>System the command is entered on</b>	<b>Requirement</b>	<b>Action taken for exceptions</b>
Active	New timeout value must be greater than or equal to twice the active interval value.	DFS3832 issued. Active timeout value forced to twice the active interval value.
Alternate	Alternate interval value must be greater than or equal to the active interval value.	DFS3812 issued. Alternate interval value forced to active interval value.
Alternate	New timeout value must be greater than or equal to twice the alternate interval value.	DFS3832 issued. Alternate timeout value forced to twice the alternate interval value.

Additional considerations when changing the INTERVAL values are shown in the following table. Listed in the table are the INTERVAL value requirements for certain systems where the command is entered and the action taken for exceptions.

*Table 103. Changing the INTERVAL value*

<b>System the command is entered on</b>	<b>Requirement</b>	<b>Action taken for exceptions</b>
Active	Twice the new interval value must be less than or equal to the active timeout value.	DFS3832 issued. Active timeout value forced to twice the new interval value.
Alternate	Alternate interval value must be greater than or equal to the active interval value.	DFS3812 issued. Alternate interval value forced to active interval value.
Alternate	New alternate interval value must be greater than or equal to the alternate LOG interval value (LNK and RDS).	DFS3833 issued. Alternate interval value forced to alternate log interval value.
Alternate	New alternate interval value must be less than or equal to the alternate RDS and LNK interval value (LOG only).	DFS3833 issued. Alternate interval value forced to alternate RDS then LNK interval value.
Alternate	Twice the new interval value must be less than or equal to the alternate timeout value.	DFS3832 issued. Alternate timeout value forced to twice the new interval value.

## Examples

This set of commands illustrates how surveillance can be modified by the /CHANGE command.

Entry ET (Alternate system):

```
/DISPLAY HSB
```

Response ET (Alternate system):

```

RSENAME      STATUS  PHASE  IMS-ID  VTAM UVAR  ACT-ID  LOG-TIME
DFSRSENM     BACKUP  TRK    IMSB    USERVAR  IMSA    10:35:17
              BACKUP SYSTEM
SURVEILLANCE INTERVAL TIMEOUT STATUS  INTERVAL  TIMEOUT  STATUS
LOG           2          99  INACTIVE          2          99  INACTIVE
LNK           4          99  INACTIVE          4          99  INACTIVE
RDS           3          99  INACTIVE          3          99  INACTIVE
TAKEOVER CONDITIONS - ALARM AUTO
RDS LINK LOG              *RDS LINK
VTAM                    *IRLM
*91226/103517*

```

Entry ET (Alternate system):

```
/CHANGE SURV LNK INTERVAL 3 (EOM)
```

Response ET (Alternate system):

```

DFS058I CHANGE COMMAND COMPLETED
DFS3812I BACKUP LNK INTERVAL VALUE OF 3 HAS BEEN FORCED TO ACTIVE VALUE OF 4
DFS3811I LNK SURVEILLANCE INACTIVE: INTERVAL VALUE CHANGED FROM 4 TO 4

```

Explanation: An attempt to change the interval value was made by entering /CHANGE on the alternate system. The new alternate interval value was not greater than or equal to the active interval value, so IMS forced the interval to the active value and issued asynchronous message DFS3812. Asynchronous message DFS3811 indicates the status of the change that follows the above action by IMS.

---

## /CHANGE TRAN command

Use the /CHANGE TRAN command with the MAXRGN keyword to change the maximum number of regions that can be simultaneously scheduled for a given transaction.

Subsections:

- "Environment"
- "Syntax"
- "Keywords" on page 149
- "Examples" on page 149

### Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

Table 104. Valid environments for the /CHANGE TRAN command and keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
/CHANGE	X	X	X
MAXRGN	X		X
TRAN	X		X

### Syntax

```

➔ /CHANGE—TRAN—traname—MAXRGN—#regions—➔
  /CHA

```

## Keywords

The following keywords are valid for the /CHANGE TRAN command:

### MAXRGN

Specifies the maximum number of regions that can be simultaneously scheduled for a given transaction.

The transaction must be eligible for parallel scheduling (load balancing). For static transactions, the MAXRGN parameter (*#regions*) is initially set by system definition using the TRANSACT macro statement. For CPI Communications driven transactions, the initial value is obtained from the TP profile. The value of *#regions* must be between 0 and the number specified on the MAXPST=*region* parameter.

## Examples

The following commands illustrate how to change the maximum number of regions that can be simultaneously scheduled for a given transaction.

Entry ET:

```
/DISPLAY TRAN SKS7
```

Response ET:

```
TRAN  CLS  ENQCT  QCT  LCT  PLCT  CP  NP  LP  SEGSZ  SEGNO  PARLM  RC
SKS7      7    0    0 65535 65535  8  8  8    0    0    1  0
      PSBNAME: DFSDDL7
      STATUS: BAL( 2)
*90226/134816*
```

Explanation: The status of transaction SKS7 indicates it is eligible for load balancing (BAL) and that two regions can be simultaneously scheduled.

Entry ET:

```
/CHANGE TRANSACTION SKS7 MAXRGN 4 (EOM)
```

Response ET:

```
DFS058I CHANGE COMMAND COMPLETED
```

Entry ET:

```
/DISPLAY TRAN SKS7
```

Response ET:

```
TRAN  CLS  ENQCT  QCT  LCT  PLCT  CP  NP  LP  SEGSZ  SEGNO  PARLM  RC
SKS7      7    0    0 65535 65535  8  8  8    0    0    1  0
      PSBNAME: DFSDDL7
      STATUS: BAL( 4)
*90226/134845*
```

Explanation: The maximum number of regions that can be simultaneously scheduled for transaction SKS7 has been changed from 2 to 4.

---

## /CHANGE UOR command

Use the /CHANGE UOR command to specify that IMS should resolve units of recovery (UORs) for protected resources on the z/OS Resource Recovery Services (RRS) platform.

**Recommendation:** Use the /CHANGE UOR command only when you are certain that no other resource managers would be adversely affected after IMS resolves the UOR.

Subsections:

- "Environment"
- "Syntax"
- "Keywords"
- "Examples"

## Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

Table 105. Valid environments for the /CHANGE UOR command and keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
/CHANGE	X	X	X
ABORT	X	X	
COMMIT	X	X	
UOR	X	X	

## Syntax



## Keywords

The following keywords are valid for the /CHANGE UOR command:

*prtkn*

Specifies the pseudorecovery token that designates the UOW to be committed. Use the /DISPLAY UOR command to obtain the pseudorecovery token (*prtkn*).

If you specify ALL, the /CHANGE UOR command affects all units of recovery.

**ABORT**

Specifies that IMS back out changes for the protected resources.

**COMMIT**

Specifies that IMS make changes permanent for the protected resources.

## Examples

The following commands illustrate the changing of a unit of recovery.

Entry ET:

```
/CHANGE UOR 010040 ABORT
```

Response ET:

```

DFS058I CHANGE COMMAND COMPLETED
DFS0699I RESYNC ABORT COMPLETE FOR PSB STLDL1, PRTKN=00010040,
      TOKEN IMS2 0000000100000000 IMS2
*97226/134816*

```

Explanation: IMS backs out changes for pseudorecovery token 010040.

Entry ET:

```
/CHANGE UOR 010040 COMMIT
```

Response ET:

```

DFS058I CHANGE COMMAND COMPLETED
DFS0699I RESYNC COMMIT COMPLETE FOR PSB STLDL1, PRTKN=00010040,
      TOKEN IMS2 0000000100000000 IMS2
*97226/134816*

```

Explanation: IMS makes changes for pseudorecovery token 010040 permanent.

---

## /CHANGE USER command

Use the /CHANGE USER command to specify that an ETO user will change. The user parameter cannot be generic.

Subsections:

- "Environment"
- "Syntax"
- "Keywords" on page 152

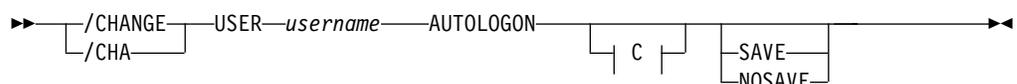
### Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

Table 106. Valid environments for the /CHANGE USER command and keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
/CHANGE	X	X	X
AUTOLOGON	X		X
ID	X		X
LOGOND	X		X
MODE	X		X
NOSAVE	X		X
SAVE	X		X
USER	X		X

### Syntax



**C:**



## Keywords

The following keywords are valid for the /CHANGE USER command:

### **AUTOLOGON**

Specifies that the autologon information previously specified by a user descriptor, the Destination Creation exit routine (DFSINSX0), or the signon exit routine (DFSSGNX0) is being updated dynamically.

#### *nodename*

Specifies the autologon terminal session for the specified user. Omitting the *nodename* clears all autologon information.

### **MODE**

Specifies the VTAM mode table entry name.

### **LOGOND**

Specifies the logon descriptor used to build the terminal control blocks.

**ID** Specifies the ISC partner's half-session qualifier (if the terminal is ISC).

### **NOSAVE**

Indicates that the changed autologon information should not be retained. Deletion of the user occurs when an IMS checkpoint is taken, at session termination, IMS restart, or XRF takeover. If **SAVE** and **NOSAVE** are omitted, IMS uses the system-wide default as specified in the DFSDCxxx PROCLIB member.

### **SAVE**

Indicates that the changed autologon information should be retained. This keyword prevents deletion of the user and remains in effect across a restart or XRF takeover until another /CHANGE command with the **NOSAVE** option is issued. If **SAVE** and **NOSAVE** are omitted, IMS uses the system-wide default as specified in the DFSDCxxx PROCLIB member.

If global resource information is kept in Resource Manager, the change is applied globally when the **SAVE** keyword is specified. If **SAVE** is not specified in this environment, the command is rejected.

---

## Chapter 13. /CHECKPOINT command

The /CHECKPOINT command records control and status information about the system log.

Subsections:

- “Environment”
- “Syntax”
- “Keywords” on page 154
- “Usage notes” on page 157
- “Examples” on page 157

### Environment

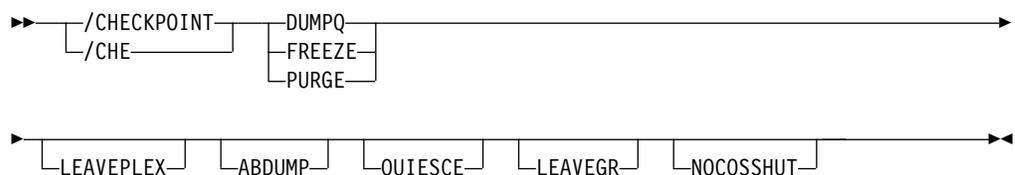
The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

Table 107. Valid environments for the /CHECKPOINT command and keywords

Command / Keyword	DB/DC	DBCTL	DCCTL
/CHECKPOINT	X	X	X
ABDUMP	X	X	X
DUMPQ	X		X
FREEZE	X	X	X
LEAVEGR	X		X
LEAVEPLEX	X	X	X
NOCQSSHUT	X		X
PURGE	X	X	X
QUIESCE	X		X
SNAPQ	X		X
STATISTICS	X	X	X

### Syntax

#### Shutdown checkpoint



**Attention:** This command shuts down your IMS system. Be sure that you understand the consequences of shutting down the system before you issue this command.

## Simple checkpoint



## Statistics checkpoint



## Keywords

The following keywords are valid for the `/CHECKPOINT` command:

### (blank)

Requests a simple checkpoint. Simple checkpoints are also invoked by IMS based on the number of entries to the system log. The number of log entries between simple checkpoints is specified during system definition. In an ETO environment, any dynamic nodes, LTERMs or users with no messages queued or status are deleted.

### ABDUMP

Requests an abnormal termination dump of the IMS control region in addition to the shutdown option selected.

In a DBCTL environment, when `/CHECKPOINT FREEZE ABDUMP` is entered, all of the DBCTL address spaces are dumped.

### FREEZE | DUMPQ | PURGE

Requests a shutdown of IMS. These variations of shutdown are provided for control over the method of stopping programs and lines, and to control the method of disposing of queues. The effects of these variations are shown in Table 108 on page 155. The request for a checkpoint shutdown might not be responded to immediately if any dependent regions are active. IMS will wait until these regions complete their current processing before continuing with the checkpoint shutdown. Message resynchronization for a specified transaction pipe does not affect IMS shutdown.

**Restriction:** A shutdown checkpoint is not allowed in a shared-queues environment if the CQS is not available.

Connections to external subsystems (not CCTL subsystems) will be quiesced. Connection attempts originating from dependent regions will be prohibited. After all dependent region connections have terminated, the control region will terminate its connection.

The `FREEZE` keyword shuts down IMS pending actions shown in Table 108 on page 155. For DBCTL, active CCTL threads are allowed to complete before this keyword takes effect. This is also true when `ABDUMP` is used with this keyword. In the DBCTL environment, a `/CHECKPOINT` command with the `PURGE` keyword will be processed as though the `FREEZE` keyword was used instead of `PURGE`. `/CHECKPOINT FREEZE` in the DBCTL environment is correct because there are no message queues to empty.

The `DUMPQ` and `SNAPQ` keywords designate starting points from which the message queue data sets can be rebuilt. However, the `SNAPQ` option dumps the message queues online while IMS is running without causing a shutdown of IMS.

The PURGE keyword attempts to empty all the queues so that no outstanding work remains. For DBCTL, IMS performs FREEZE processing because there are no message queues.

In a DBCTL environment, when /CHECKPOINT FREEZE ABDUMP is entered, all of the DBCTL address spaces are dumped.

In a shared-queues environment, the DUMPQ and PURGE keywords cause IMS to shut down (as if you entered a /CHECKPOINT FREEZE command), but the message queues are not dumped or purged because the local IMS subsystem has no local queues. To dump the shared message queues when CQS terminates, use the /CQSET command before issuing the IMS shutdown checkpoint command.

The following table displays when to issue one of the three IMS shutdown options (FREEZE, DUMPQ, and PURGE) and the effect on the IMS resource.

*Table 108. IMS shutdown options and their effect on IMS resources*

Resource status	FREEZE keyword	DUMPQ keyword	PURGE keyword
Message processing regions stopped	At program completion	At program completion	When transaction queues are empty
Batch message processing regions stopped	At checkpoint, SYNC call, or program completion	At checkpoint, SYNC call, or program completion	At program completion
Line input stopped	At message completion	At message completion	At message completion
Line output stopped	At message completion	At message completion	When all messages complete
Transaction and message queues	Retained in queue data sets	Dumped to system log	Emptied normally
Message-driven regions stopped	At message completion	At program completion	When transaction queues are empty
Fast Path output messages queued	Dumped to system log	Dumped to system log	Emptied normally
Fast Path input messages queued	Discarded	Emptied normally	Emptied normally
Fast Path DEDB online utility region	At program completion	At program completion	At program completion
Fast Path DEDBs	Closed	Closed	Closed
MSDBs	Dumped to MSDB checkpoint data set	Dumped to MSDB checkpoint data set	Dumped to MSDB checkpoint data set
CCTL Regions	See DRA thread statistics (System Programming APIs) for a discussion of DRA threads.		

### LEAVEGR

Deletes all affinities from the VTAM affinity table for the IMS subsystem on which this command is issued. Using this keyword removes the IMS subsystem from the generic resource group.

**Recommendation:** Cold start the DC component of an IMS subsystem that has been shut down with the LEAVEGR keyword to ensure that all affinities in IMS control blocks are also deleted. The IMS subsystem rejoins the generic resource group during startup.

If the VTAM ACB is closed (usually because of a /STOP DC command), a shutdown checkpoint command with the LEAVEGR keyword is rejected.

### LEAVEPLEX

In an IMSplex, this keyword is specified if the IMS that is being shut down is not going to rejoin the IMSplex. Specify the LEAVEPLEX keyword when you do not intend to bring the IMS back up in the IMSplex.

If LEAVEPLEX is specified, and the IMS is a member of an IMSplex with global online change enabled, an attempt is made to remove the IMS ID from the OLCSTAT data set. If there is any error in removing the IMS ID from the OLCSTAT, message, DFS3443, DFS3444, or DFS3448 is written out to the system console and the IMS is shut down. In this case, the IMS ID might still be in the OLCSTAT data set and will have to be deleted by using the DFSUOLC utility.

When an IMS is defined to the IMSRSC repository and is leaving the IMSplex, ensure you clean up the information in the IMSRSC repository before or after the /CHE FREEZE | DUMPQ | PURGE LEAVEPLEX command is issued.

The DELETE DEFN TYPE(IMSIDMBR) FOR(IMSID(imsid)) command, where IMSID is the IMS ID of the IMS leaving the IMSplex, must be issued in order to clean-up the information of the IMS from the repository. This command can be issued before the shutdown when the LEAVEPLEX keyword is issued or from another IMS after the IMS has left the IMSplex.

On a XRF system, if the active and alternate are leaving the IMSplex then the DELETE DEFN TYPE(IMSIDMBR) command must be issued with the IMS ID of both the active and the alternate.

#### Note:

- Turn off or disable autoexport to the IMSRSC repository using the UPD IMS SET(AUTOEXPORT(N)) command at the active IMS before the DELETE DEFN TYPE(IMSIDMBR) command is issued to remove the IMS ID from the repository.
- Issue the DELETE DEFN TYPE(IMSIDMBR) command for an active IMS just before it is being shutdown with the LEAVEPLEX keyword.
- Do not route the EXPORT DEFN TARGET(REPO) command to the IMS after its IMS ID has been removed from the repository.

### NOCQSSHUT

Is used with the /CHE DUMPQ, /CHE FREEZE, or /CHE PURGE commands to not shut down the CQS address space when the IMS control region terminates. The CQS address space remains active and connected to the message queue structures. NOCQSSHUT is only applicable when IMS is running in a shared-queues environment. The default is to shut down the CQS address space when the IMS control region terminates.

### QUIESCE

Halts processing of all VTAM terminals. When QUIESCE is specified, IMS sends the VTAM shutdown indicator to all VTAM terminals and waits until these nodes have completed processing before performing the normal checkpoint shutdown. During the processing of a quiesce shutdown, the master terminal operator might want to terminate the VTAM network without waiting for the orderly termination to complete. This can be done by entering the /CHECKPOINT command again, either with FREEZE, DUMPQ, or PURGE but without QUIESCE.

## **SNAPQ**

Requests a simple checkpoint and dumps the contents of the message queues to the system log.

In a shared-queues environment, /CHECKPOINT SNAPQ does not snap the queues because the local IMS subsystem has no local queues. Use the /CQCHKPT command to initiate a CQS structure checkpoint.

In an XRF environment, /CHECKPOINT SNAPQ synchronizes the active and alternate IMS subsystems.

## **STATISTICS**

Requests that IMS performance records be created and written to the system log. No other checkpoint processing occurs.

The /CHECKPOINT STATISTICS command does not create a system checkpoint on the log.

## **Usage notes**

The /CHECKPOINT command can be used to take a simple checkpoint of IMS, with the option of also dumping the contents of the message queue data sets to the system log or to shut down IMS normally. When IMS is shut down by the /CHECKPOINT command, it can be restarted with the /NRESTART command.

This command can be issued to an IMSplex using the Batch SPOC utility.

There are two conditions under which IMS cannot complete a shutdown normally. The most frequent is when there are multiple-page messages on a master terminal non-VTAM line, paging is not complete, and the master terminal is sharing a communication line with other physical terminals. It might be necessary to use the /ASSIGN command to reassign the master terminal to a line by itself so that the /IDLE command can be issued to terminate the multiple page output. If the master terminal cannot be reassigned to another line, assign it to the system console line.

The second condition occurs when a number of system messages generated by IMS are awaiting delivery to the master terminal. All system messages destined for the master terminal will be delivered, because they can affect the way the system is restarted. The master terminal operator should acknowledge delivery of a message by causing an I/O interrupt; that is, pressing PA2, which in turn causes another message to be sent, if one exists. Another option is to assign the master terminal to the system console.

When the /CHECKPOINT command is used to shut down IMS, the /BROADCAST command can be helpful in notifying the remote terminal operators that IMS is shutting down.

If OTMA is active and a tpipe is idle for three consecutive system checkpoints, depending on certain eligibility requirements, the tpipe is removed.

## **Examples**

The following are examples of the /CHECKPOINT command:

### *Example 1 for /CHECKPOINT command*

Entry ET:

/CHECKPOINT

Response ET:

```
DFS058I (time stamp) CHECKPOINT COMMAND IN PROGRESS
DFS994I *CHKPT 82102/110247**SIMPLE*
```

Explanation: A simple checkpoint of IMS is written to the system log at 110247 (time) on 82102 (Julian date). The checkpoint number is 82102/110247.

*Example 2 for /CHECKPOINT command*

Entry ET:

/CHECKPOINT FREEZE

Response ET:

```
DFS058I (time stamp) CHECKPOINT COMMAND IN PROGRESS
DFS994I *CHKPT 82206/120118**FREEZE*
```

Explanation: IMS is terminated after all checkpoint freeze functions are completed. The checkpoint is written to the system log at 120118 (time) on 82206 (Julian date). The checkpoint number is 82206/120118.

*Example 3 for /CHECKPOINT command*

Entry ET:

/CHECKPOINT FREEZE QUIESCE

Response ET:

```
DFS058I (time stamp) CHECKPOINT COMMAND IN PROGRESS
DFS994I *CHKPT 82102/110247**FREEZE*
```

Explanation: IMS is terminated after all VTAM nodes have returned a shutdown-complete indicator to IMS and IMS has completed all checkpoint freeze functions. The checkpoint is written to the system log at 110247 (time) on 82102 (Julian date). The checkpoint number is 82102/110247.

*Example 4 for /CHECKPOINT command*

Entry ET:

/CHECKPOINT PURGE ABDUMP

Response ET:

```
DFS058I (time stamp) CHECKPOINT COMMAND IN PROGRESS
DFS994I *CHKPT 82128/101112**PURGE*
```

Explanation: IMS is terminated after all checkpoint purge functions are completed. The checkpoint is written to the system log at 101112 (time) on 82128 (Julian date). The checkpoint number is 82128/101112.

A z/OS ABEND message is issued when the dump of the IMS control region is completed.

*Example 5 for /CHECKPOINT command*

Entry ET:

```
/CHECKPOINT PURGE
```

Response ET:

```
DFS058I (time stamp) CHECKPOINT COMMAND IN PROGRESS  
DFS994I *CHKPT 82103/131415**PURGE*
```

Explanation: IMS is terminated after all checkpoint purge functions are completed. The checkpoint is written to the system log at 131415 (time) on 82103 (Julian date). The checkpoint number is 82103/131415.

*Example 6 for /CHECKPOINT command*

Entry ET:

```
/CHECKPOINT STATISTICS
```

Response ET:

```
/DFS058I (timestamp) CHECKPOINT COMMAND IN PROGRESS  
/DFS994I STATISTICS CHECKPOINT COMMAND COMPLETED
```

Explanation: IMS performance statistics records are written to the system log. No other information is saved. This checkpoint cannot be used for a system restart.



## Chapter 14. /CLSDST command

The /CLSDST command causes IMS to disconnect an ISC TCP/IP or a VTAM terminal.

Subsections:

- "Environment"
- "Syntax"
- "Keywords"
- "Usage notes" on page 163
- "Example" on page 163

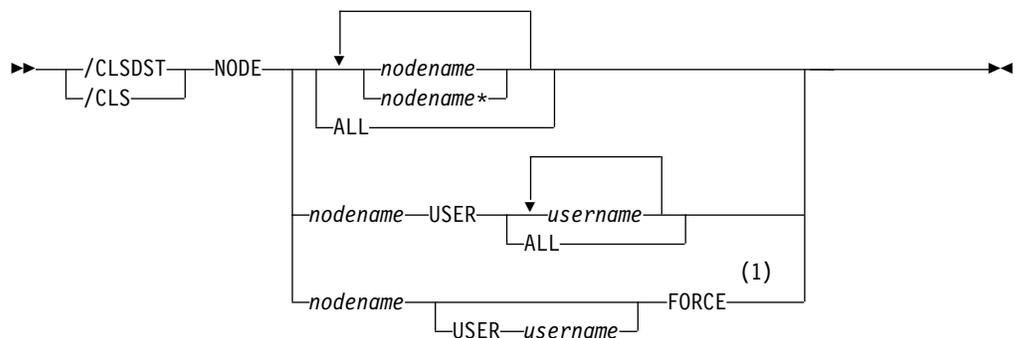
### Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

Table 109. Valid environments for the /CLSDST command and keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
/CLSDST	X		X
FORCE	X		X
NODE	X		X
USER	X		X

### Syntax



Notes:

- 1 ISC TCP/IP nodes do not support the FORCE keyword.

### Keywords

The following keywords are valid for the /CLSDST command:

#### FORCE

Allows IMS to reinitialize a hung node with I/O in progress. FORCE reinitializes one session and is valid only on an active IMS system. To activate FORCE on VTAM nodes, use the command /CLSDST NODE P1 FORCE.

For ISC VTAM nodes with parallel sessions, the USER keyword must be specified and only one USER parameter is allowed.

ISC TCP/IP nodes do not support the FORCE keyword.

FORCE operates under the following conditions:

- A display from VTAM indicates that no session exists. If a session does exist, or is in process, you must issue the VTAM command VARY NET,INACT,FORCE to terminate the session, and the command must successfully complete. This command terminates all parallel sessions.

**Attention:** If VARY NET,INACT,FORCE is not issued or does not successfully complete before you issue the FORCE command, and the terminal is in the process of creation or termination, the results might be unpredictable.

- The session is connected to IMS and output is in progress. A /DISPLAY on the node indicates that a CID exists, that the node is connected (a status of CON is displayed), and that the node is not idle (a status of IDLE is not displayed).

#### **NODE**

Specifies the node to be disconnected by IMS. The specified node must be connected before a command is issued to disconnect it, as indicated by CON on the /DISPLAY NODE command referring to that terminal.

If the USER keyword is omitted for ISC nodes, all half-sessions of a session type 6 node are terminated. If the USER keyword is omitted, generic parameters are allowed for the NODE keyword.

The timing of the disconnection depends on the type of terminal:

- For keyboards or printers, console components and interactive terminals, the disconnection occurs at the next message boundary.
- For component types that group messages (such as a SLU 1 statement reader, printer, or disk), the disconnection occurs at the end of any group where processing is in progress.
- For 3270 displays, the disconnection occurs at the completion of the current (if any) I/O operation.

#### **USER**

Must be specified with the NODE keyword. NODE USER specifies the ISC user allocated to the ISC node or the dynamic user signed on to the dynamic node. When a /CLSDST NODE nodename USER username command is issued, it only affects the NODE if the USER is still associated with the node.

For non-ISC dynamic nodes, this command is valid only if the user is still signed on to the node. For ISC nodes, the half-sessions of the ISC node allocated to the specific users are terminated and the users are not deallocated from the session. On restart, the /OPNDST command must specify the same users and ID pairs.

#### **Restrictions for using NODE and USER parameters together:**

- Commands with the NODE USER keyword pair are valid only if:
  - The USER is signed on to the NODE
  - In an ISC environment, the USER is allocated to the NODE
  - The nodes and users already exist
- /CLSDST NODE USER commands are valid for ISC and non-ISC nodes and users.

## Usage notes

/CLSDST resets preset mode, test mode, lock node, lock LTERM, pstop LTERM, and purge LTERM, because these statuses are not significant and therefore are not kept after a logon or restart.

This command can be issued to an IMSplex using the Batch SPOC utility.

The /CLSDST command will also do some other cleanup depending on the recovery settings for the node. Below are the actions taken:

### RCVYSTSN=NO

/CLSDST acts like a /CHANGE NODE COLDSESS command for FINANCE and SLUP nodes by setting the session status to "cold".  
/CLSDST will act like a /QUIESCE NODE command for ISC (LU6.1) nodes by initiating the shutdown and deallocating the user for the specified node. This action changes the session status to cold. With these actions taken by the /CLSDST command, the next session initiation request for this node will be allowed to again attempt a session cold start. For ETO nodes, the control block structure could be deleted, if no significant status exists.

### RCVYCONV=NO

/CLSDST causes any IMS conversations (active and held) to be terminated. Any conversational message that is queued or being processed will have its output response message delivered asynchronously.

### RCVYFP=NO

/CLSDST causes Fast Path status and messages to be discarded.

### RCVYRESP=NO

/CLSDST resets full-function response mode.

If global resource information is not kept in Resource Manager (RM), /CLSDST logs a node off and resets status locally. If global resource information is kept in RM, /CLSDST resets status globally. If the node has no significant status, /CLSDST deletes the node in RM.

If ROUTE is specified, it should be specified with ROUTE(\*). The command fails if not routed to the IMS where the node is active.

## Example

Entry ET:

```
/CLSDST NODE WEST
```

Response ET:

```
DFS058I CLSDST COMMAND COMPLETED
```

Explanation: The node, WEST, is disconnected from IMS.



---

## Chapter 15. /COMPT command

The /COMPT command sets a particular terminal component to a ready or not-ready state.

Output messages queued for a particular component will not be sent unless the component is ready. Depending on terminal type and the availability of messages queued for other components, output operations for other components can continue.

Subsections:

- “Environment”
- “Syntax” on page 166
- “Keywords” on page 166
- “Usage notes” on page 167
- “Examples” on page 167

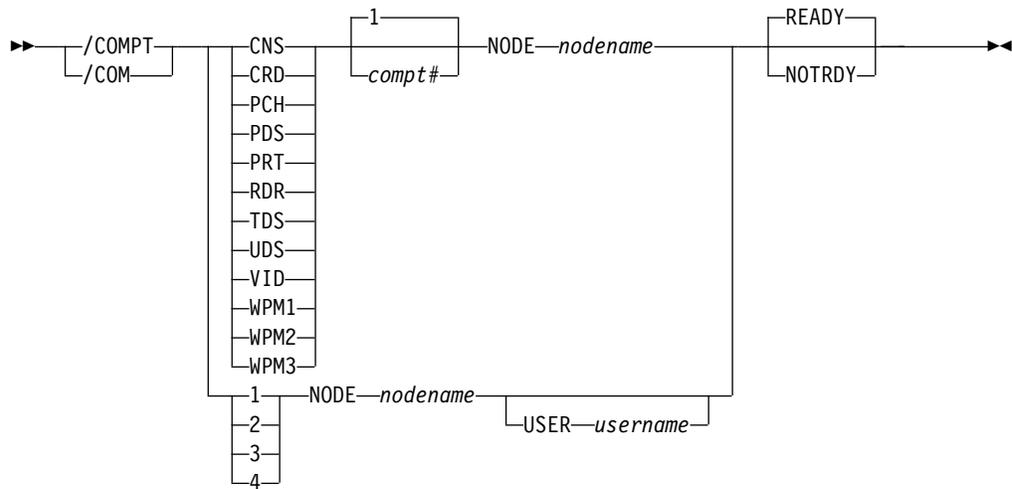
### Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

*Table 110. Valid environments for the /COMPT command and keywords*

Command / Keywords	DB/DC	DBCTL	DCCTL
/COMPT	X		X
CNS	X		X
CRD	X		X
NODE	X		X
NOTRDY	X		X
PCH	X		X
PDS	X		X
PRT	X		X
RDR	X		X
READY	X		X
TDS	X		X
UDS	X		X
USER	X		X
VID	X		X
WPM1	X		X
WPM2	X		X
WPM3	X		X

## Syntax



## Keywords

The following keywords are valid for the /COMPT command:

**CNS | CRD | PCH | PDS | PRT | RDR | TDS | UDS | VID | WPM1 | WPM2 | WPM3 |  
1 | 2 | 3 | 4**

Specifies the component that you want to set to a ready or not-ready state.

The command format can take two forms. That is, a component can be referenced by using a keyword, such as VID, or by using a number, such as 2.

When a keyword is used, a search is made of the components (as defined in the TERMINAL macro during IMS system definition) for the component type defined that corresponds to the specified keyword.

When a match is found, that component is made ready/not ready as specified by the command. If a number other than 1 follows the keyword, the corresponding occurrence of that component type is made ready/not ready.

/COMPT supports up to four components. When a number from 1 through 4 is used instead of a keyword, the component affected is the one defined in that position during system definition, independent of component type.

**Restriction:** When the /COMPT command contains the keyword CRD, it cannot also contain any of the following keywords: WPM1, WPM2, or WPM3.

### READY

Sets the specified terminal component to a ready state.

### NOTRDY

Sets the specified terminal component to a not-ready state.

### NODE

Specifies the node name (as specified by the TERMINAL macro) of the terminal component that is to be set in ready or not-ready state.

### USER

Specifies the ISC subpool name of the *nodename*. The USER keyword is required if the node specified on the command is an ISC node.

## Usage notes

The ready or not-ready state set by the /COMPT command can be altered by the following:

- Another /COMPT command
- A /START, /RSTART, or /RCOMPT command
- An I/O error on the terminal component

The /COMPT command can only refer to a VTAM-attached terminal component.

### Restrictions for using NODE and USER parameters together:

- Commands with the NODE USER keyword pair are valid only if:
  - The USER is signed on to the NODE
  - In an ISC environment, the USER is allocated to the NODE
  - The nodes and users already exist
- /COMPT (1|2|3|4) NODE USER (READY/NOTRDY) commands are valid for ISC and non-ISC nodes and users.

## Examples

The following are examples of the /COMPT command:

### *Example 1 for /COMPT command*

Entry ET:

```
/COMPT 4 NODE ABC READY
```

Response ET:

```
DFS058I  COMPT COMMAND COMPLETED
```

Explanation: The fourth component defined in the TERMINAL macro on node ABC is made ready to IMS.

### *Example 2 for /COMPT command*

Entry ET:

```
/COMPT VID 2 NODE ABC READY
```

Response ET:

```
DFS058I  COMPT COMMAND COMPLETED
```

Explanation: The second display component on node ABC is declared operable to IMS.

### Related reference:

Chapter 9, “/ASSIGN command,” on page 101



---

## Chapter 16. /CQCHKPT command

The /CQCHKPT command initiates a CQS checkpoint for a specific coupling facility list structure or all the coupling facility list structures to which the IMS subsystem is connected.

Subsections:

- “Environment”
- “Syntax”
- “Keywords”
- “Usage notes” on page 170
- “Examples” on page 170

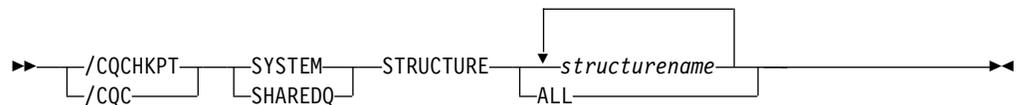
### Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

Table 111. Valid environments for the /CQCHKPT command and keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
/CQCHKPT	X		X
SHAREDQ	X		X
STRUCTURE	X		X
SYSTEM	X		X

### Syntax



### Keywords

The following keywords are valid for the /CQCHKPT command:

#### SHAREDQ

Specifies that the entire queue structure is to be checkpointed to the structure recovery data set. While the checkpoint is in progress for the structure, no CQS can access the structure.

During a structure checkpoint, every CQS connected to that structure also takes a system checkpoint.

**Recommendation:** When possible, issue this command when it will have the least performance impact to your online IMS subsystems.

#### STRUCTURE

Specifies a specific structure name (or all) for which a CQS checkpoint is to be taken. If an overflow structure exists for a structure, a checkpoint is taken for both the primary and overflow structure.

## SYSTEM

Specifies that a system checkpoint is to be taken; the CQS internal tables are checkpointed and written to the CQS log. Only the CQS for which you enter the command takes a system checkpoint.

## Usage notes

This command sends the DFS058 CQCHKPT COMMAND IN PROGRESS message to the inputting terminal, and sends an asynchronous response to the system console and master terminal when the CQS checkpoint is complete.

This command is valid only in a shared-queues environment.

This command can be issued to an IMSplex using the Batch SPOC utility.

## Examples

The following are examples of the /CQCHKPT command:

### *Example 1 for /CQCHKPT command*

Entry ET:

```
/CQCHKPT SYSTEM STRUCTURE IMSMSGQ01
```

Response ET:

```
DFS058I CQCHKPT COMMAND IN PROGRESS  
CQS0030I SYSTEM CHECKPOINT COMPLETE, STRUCTURE IMSMSGQ01 ,  
LOGTOKEN 0000000001687D3F CQS1CQS  
DFS1972I CQCHKPT SYSTEM COMMAND COMPLETE FOR STRUCTURE=IMSMSGQ01
```

Explanation: A CQS system checkpoint completes successfully.

**Recommendation:** Record the log token displayed in the CQS0030I message because you might need it for a CQS restart.

### *Example 2 for /CQCHKPT command*

Entry ET:

```
/CQCHKPT SHAREDQ STRUCTURE IMSMSGQ01
```

Response ET:

```
DFS058I CQCHKPT COMMAND IN PROGRESS  
CQS0220I CQS CQS1CQS STARTED STRUCTURE CHECKPOINT FOR  
STRUCTURE IMSMSGQ01 CQS1CQS  
CQS0200I STRUCTURE IMSMSGQ01 QUIESCED FOR  
STRUCTURE CHECKPOINT CQS1CQS  
CQS0201I STRUCTURE IMSMSGQ01 RESUMED AFTER  
STRUCTURE CHECKPOINT CQS1CQS  
CQS0030I SYSTEM CHECKPOINT COMPLETE, STRUCTURE IMSMSGQ01 ,  
LOGTOKEN 0000000001688652 CQS1CQS  
CQS0221I CQS CQS1CQS COMPLETED STRUCTURE CHECKPOINT  
FOR STRUCTURE IMSMSGQ01 CQS1CQS  
DFS1972I CQCHKPT SHAREDQ COMMAND COMPLETE FOR STRUCTURE=IMSMSGQ01
```

Explanation: A CQS structure checkpoint for a specific structure completes successfully.

---

## Chapter 17. /CQQUERY command

The /CQQUERY command displays information regarding a specific coupling facility list structure or all the coupling facility list structures holding IMS messages.

The /CQQUERY STATISTICS command displays the statistics and status information for the coupling facility list structures specified by the STRUCTURE keyword.

Subsections:

- “Environment”
- “Syntax”
- “Keywords”
- “Usage notes” on page 172
- “Examples” on page 172

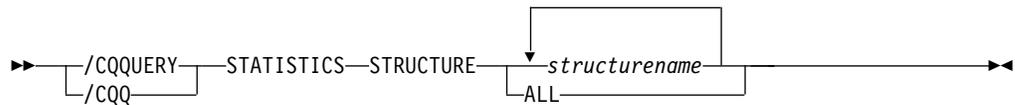
### Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

Table 112. Valid environments for the /CQQUERY command and keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
/CQQUERY	X		X
STATISTICS	X		X
STRUCTURE	X		X

### Syntax



### Keywords

The following keywords are valid for the /CQQUERY command:

#### STATISTICS

Specifies that statistics should be gathered and displayed.

#### STRUCTURE

Specifies that one or more structure names follow.

Only primary, coupling facility list-structure names used by IMS for shared queues are valid.

## Usage notes

This command is valid only in a shared-queues environment.

The following statistics information is displayed:

- Number of data elements that can be allocated in the structure
- Number of list entries that can be allocated in the structure
- Number of data elements in use in the structure
- Number of list entries in use in the structure
- Entry-to-element ratio

The entry-to-element ratio is the ratio of list entries to list elements in the structure when the structure was first allocated, or last altered (for structures defined with ALLOWAUTOALT(YES)). The ratio value is initially derived by CQS from the OBJAVGSZ parameter in the CQSSGxxx PROCLIB member. The ratio can be changed by z/OS dynamically as needed for ALLOWAUTOALT(YES) structures. The ratio might also change slightly when a structure is rebuilt, because CQS uses the saved values of the actual number of allocated entries and elements to derive the entry-to-element ratio for the connection to a rebuild structure.

In an IMSplex, /CQQUERY displays information regarding a specific shared queue coupling facility list structure or all the shared queue coupling facility list structures holding IMS messages. /CQQUERY does not display any information about resource structures.

When the /CQQ command is issued through OM, command processing is not serialized through the IMS CTL TCB.

This command can be issued to an IMSplex by using the Batch SPOC utility.

## Examples

The following are examples of the /CQQUERY command:

### *Example 1 for /CQQUERY command*

Entry ET:

```
/CQQUERY STATISTICS STRUCTURE ALL
```

Response ET:

STRUCTURE NAME	LEALLOC	LEINUSE	ELMALLOC	ELMINUSE	LE/EL
IMSMGQ01	1789	11	1787	19	00001/00001
IMSMGQ010FLW	N/A	N/A	N/A	N/A	N/A
IMSEMHQ01	377	3	375	3	00001/00001
IMSEMHQ010FLW	N/A	N/A	N/A	N/A	N/A

\*08282/211714\*

Explanation: This command displays the structure statistics of the message queue and Fast Path EMH queue structures used by IMS. The list entries allocated (LEALLOC), the list entries in use (LEINUSE), the elements allocated (ELMALLOC) and the elements in use (ELMINUSE) show the usage of the structures. When CQS allocates a structure, it allocates a certain number of list entries and elements to manage data on the structure. A structure is full if all list entries are in use or if all elements are in use.

*Example 2 for /CQQUERY command*

Entry ET:

```
/CQQ STATISTICS STRUCTURE IMSMSGQ01
```

Response ET:

STRUCTURE NAME	LEALLOC	LEINUSE	ELMALLOC	ELMINUSE	LE/EL
IMSMSG101	9132	9027	9130	9071	00001/00001
IMSMSGQ010FLW	1915	1866	1912	1866	00001/00001

\*08282/211714\*

Explanation: This command displays the structure statistics of the message queue structure used by IMS and its associated overflow queue.

**Related concepts:**

 Using structure alter for CQS (System Administration)

**Related reference:**

 CQSSGxxx member of the IMS PROCLIB data set (System Definition)



---

## Chapter 18. /CQSET command

Use a /CQSET SHUTDOWN SHAREDQ ON|OFF command to tell CQS whether to take a structure checkpoint during normal shutdown.

You can specify a structure checkpoint for a specific coupling facility or for all coupling facility list structures used by IMS.

Subsections:

- "Environment"
- "Syntax"
- "Keywords"
- "Usage notes" on page 176
- "Example" on page 176

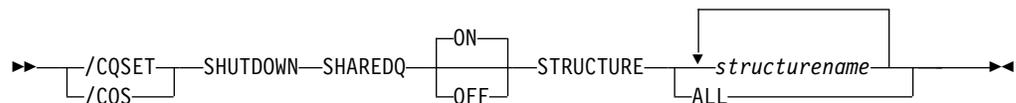
### Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

Table 113. Valid environments for the /CQSET command and keywords

Command / Keyword	DB/DC	DBCTL	DCCTL
/CQSET	X		X
SHAREDQ	X		X
SHUTDOWN	X		X
STRUCTURE	X		X

### Syntax



### Keywords

The following keywords are valid for the /CQSET command:

#### SHAREDQ

Specifies that the entire queue structure is to be checkpointed to the structure recovery data set. While the checkpoint is in progress for the structure, no CQS can access the structure.

During a structure checkpoint, every CQS connected to that structure also takes a system checkpoint.

#### SHUTDOWN

Specifies that CQS should take a structure checkpoint during normal CQS shutdown.

**STRUCTURE**

Specifies a specific structure name (or all) for which a CQS checkpoint is to be taken. If an overflow structure exists for a structure, a checkpoint is taken for both the primary and overflow structure.

**Usage notes**

IMS initiates a normal CQS shutdown during a normal IMS shutdown.

This command is valid only in a shared-queues environment.

In an IMSplex, when the /CQS command is issued through OM command processing, it is not serialized through the IMS CTL TCB.

This command can be issued to an IMSplex using the Batch SPOC utility.

**Example**

Entry ET:

```
/CQSET SHUTDOWN SHAREDQ ON STRUCTURE ALL
```

Response ET:

```
DFS058I CQSET COMMAND COMPLETE
```

Explanation: CQS will take a structure checkpoint when it shuts down.

---

## Chapter 19. CREATE commands

Use the IMS CREATE commands to create resources and resource descriptors.

These commands can be issued through TSO SPOC, IBM Management Console for IMS and DB2® for z/OS, the Manage Resources options in the IMS Applications menu, or other user-written programs that issue commands through OM API. These commands can also be issued to an IMSplex by using the Batch SPOC utility.

Subsections:

- “CREATE DB command”
- “CREATE DBDESC command” on page 185
- “CREATE IMSCON commands” on page 191
- “CREATE OTMADESC command” on page 250
- “CREATE PGM command” on page 260
- “CREATE PGMDESC command” on page 271
- “CREATE RTC command” on page 281
- “CREATE RTCDESC command” on page 287
- “CREATE TRAN command” on page 292
- “CREATE TRANDESC command” on page 316

---

### CREATE DB command

Use the CREATE DB command to create the database resource requirements for one or more physical databases managed by IMS.

Subsections:

- “Environment”
- “Syntax” on page 178
- “Keywords” on page 178
- “Usage notes” on page 180
- “Output fields” on page 181
- “Return, reason, and completion codes” on page 181
- “Examples” on page 183

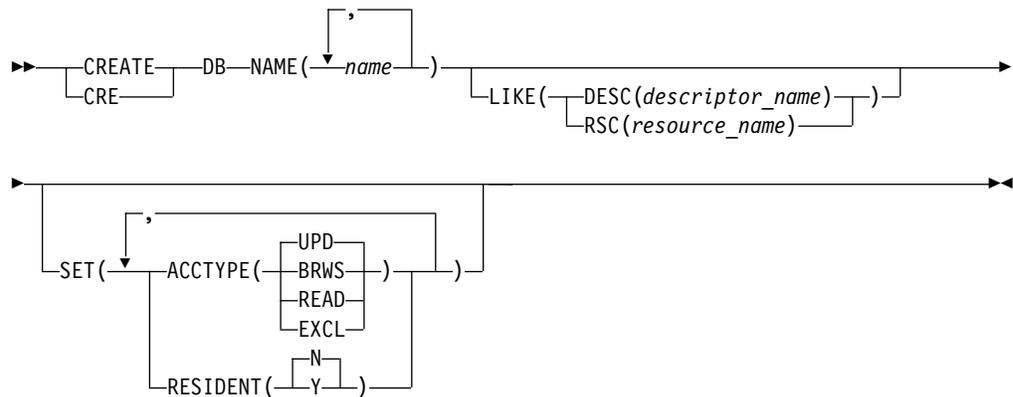
#### Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

*Table 114. Valid environments for the CREATE DB command and keywords*

Command / Keywords	DB/DC	DBCTL	DCCTL
CREATE DB	X	X	
LIKE	X	X	
NAME	X	X	
SET	X	X	

## Syntax



Default values shown in this syntax diagram reflect the default values that are defined in the IMS database descriptor, DFSDSDB1. If you define another descriptor as the default with a CREATE or UPDATE DBDESC command, the default values shown in this syntax diagram might not match the value defined in the current default descriptor.

## Keywords

The following keywords are valid for the CREATE DB command:

### LIKE

Specifies that the resource is created using an existing resource or descriptor as a model, instead of the default descriptor. The default descriptor is either the IMS descriptor DFSDSDB1, or the one that you define. The model type can either be a descriptor (DESC), or a resource (RSC). The new resource is created with all the same attributes as the model, except for the DEFAULT(Y) value. Attributes set explicitly by the CREATE command override the model attributes. Later changes to the model are not propagated to resources or descriptors that were created from it.

You can use QUERY commands to display the model name and model type used to create the resource. The model name and type returned from the QUERY command are provided for informational purposes only. The EXPORT command exports a resource definition, including the model name and model type, to the definition member. The IMPORT command imports a resource definition, including the model name and model type, from the definition member. An IMPORT command that creates a resource saves the model name and model type, but does not use the models attributes.

When you specify LIKE on a CREATE command, IMS sets the attributes of the newly created resource or descriptor to the models current attributes, except for the default attribute.

### DESC(descriptor\_name)

Specifies the name of the descriptor to use as a model to define this resource.

**RSC(resource\_name)**

Specifies the name of the resource to use as a model to define this resource.

**NAME**

Specifies the 1-8 character name of the database (DBD name). Names must be alphanumeric (A through Z, 0 through 9, #, \$, and @). All database names must begin with an alphabetic character (A through Z, #, \$, and @) followed by 0-7 alphanumeric characters. Names cannot include a blank, comma, period, hyphen, or equal sign. Names cannot begin with DFS, except for DFSIVPxx and DFSSAMxx. Names cannot be BASICEDT, DBCDM, DBRC, ISCEDT, MSDB, SDB, or WTOR. A database resource and a database descriptor can have the same name.

**SET**

Specifies the attributes of the database to be created. If the LIKE keyword is omitted, attributes not specified take the value defined in the default descriptor, which is either IMS descriptor DFSDSDB1 or user-defined. If the LIKE keyword is specified, attributes not specified take the value defined in the model.

**ACCTYPE()**

Specifies the access intent for the named database, which is how the subsystem requesting access plans to use the database. Access type is used with the database sharing level declared to DBRC. Access types include the following options:

**BRWS**

The database is available for read-only processing on this IMS subsystem. The only programs that can use the database on this subsystem are those that have a PCB processing option of GO (PROCOPT=GO). Programs that access the data using the GO processing option might see uncommitted data since a sharing IMS subsystem could be updating the database. The database is opened for read-only processing.

**EXCL**

The database is to be used exclusively by this IMS subsystem. This exclusive access is guaranteed only when the database is registered to DBRC.

**READ**

The database is available for read-only processing in this IMS subsystem. Programs with update intent can be scheduled, but cannot update the database. Access type READ differs from access type BRWS in that the data is read with integrity (locking is performed) and all programs can access the data, not just those with a processing option of GO. The database is opened for read-only processing.

**UPD**

The database is available for update as well as read processing in the IMS subsystem. This is the default.

**RESIDENT**

Specifies the resident option. The RESIDENT(N) option takes effect immediately. The RESIDENT(Y) option takes effect at the next restart, unless the database was created as RESIDENT(Y) after the checkpoint from which this IMS is performing emergency restart. A database defined as a

DEDB or an MSDB in ACBLIB always sets the RESIDENT(Y) attribute, regardless of the RESIDENT value specified.

- N** The DMB associated with the named database resource is not made resident in storage. The DMB is loaded at scheduling time.
- Y** The DMB associated with the named database resource is made resident in storage at the next IMS restart. At the next IMS restart, IMS loads the DMB and initializes it. A resident database is accessed from local storage, which eliminates I/O to the ACBLIB. In an online environment, the DMB control blocks are stored in the ACBLIB. If the DLI/SAS address space exists, DLI/SAS loads the DMB; otherwise, it is the IMS control region that loads it. This makes the DMB dependent on the existence of the corresponding database resource.

## Usage notes

The database can be HSAM, HISAM, HDAM, HIDAM, DEDB, MSDB, or a HALDB master database. Two database resources must be defined for a HIDAM database; one for the INDEX DBD, and one for the HIDAM DBD. No database resource should be defined for a PHIDAM primary index. One database must be defined for each secondary index that refers to any database defined to the online system. The database cannot be a HALDB partition. HALDB partitions can be defined using the HALDB Partition Definition utility.

A BLDL is performed on the DMB associated with this database. The database is created even if the DMB is not defined in ACBLIB. In this case, the database cannot be used. The database status is NOTINIT when a /DISPLAY DB or QUERY DB is issued for the database. Online Change must be used to add the DMB to ACBLIB, before the database can be used.

After a CREATE DB command for a HALDB master database, the user must issue a QUERY DB command or a /DISPLAY DB command for the HALDB master to determine if the partitions were built. If the partitions were not built, the user must issue one of the following commands to initialize the partitions:

- A /START DB *HALDB\_Master* OPEN command
- An UPDATE DB NAME(*HALDB\_Master*) START(ACCESS) OPTION(OPEN) command

Fast Path DEDB or MSDB-related control blocks are loaded.

If the DMB is in the ACBLIB library, IMS loads the DMB and can determine what type of database it is. IMS takes different actions depending on whether it is a full function database, a DEDB, or an MSDB.

- For full function databases, the DMB is loaded into the DMB pool. Even if the database is created with RESIDENT(Y), DMB residency does not take effect until the next IMS restart.
- For DEDBs, the DMCB (DEDB version of a DMB), is chained into the DMCB chain (all DEDB DMCBs are resident). The parameters are then checked for CI size and to see whether the DMCB contains an area that already exists. If the CI size of any AREA is larger than the CI size specified for the DBBF buffers (fast path global buffer pool), then the CREATE command fails with completion code E3. If the DMCB contains an area definition that already exists in the FPAL (Fast Path Area List), the CREATE command fails with completion code E4.
- For MSDBs, a special procedure is required to make the MSDB available. The MSDB can then be used at the next IMS restart

If the DMB is not in the ACBLIB library, then the DDIR control block is successfully created, but has an initial status of NOTINIT.

Resources exist in IMS until they are deleted using a DELETE command. Resources are recoverable across an IMS warm start or emergency restart. Resources are lost if IMS is cold started, unless cold start imports definitions that were exported while IMS was up.

The CREATE DB command is similar to local online change (using a /MODIFY command) or global online change (using an INITIATE OLC command) for database resources in the MODBLKS data set, except that databases are added dynamically.

**Restrictions:**

- The CREATE DB command can be issued only through the OM API.
- This command is not valid:
  - On the XRF alternate, RSR tracker, or FDBR region.
  - If online change for MODBLKS is enabled (DFSDFxxx or DFSCGxxx are defined with MODBLKS=OLC, or MODBLKS is not defined).

**Output fields**

The following table shows the CREATE DB output fields. The columns in the table are as follows:

**Short label**

Contains the short label generated in the XML output.

**Keyword**

Identifies keyword on the command that caused the field to be generated. N/A appears for output fields that are always returned. *error* appears for output fields that are returned only in case of an error.

**Meaning**

Provides a brief description of the output field.

*Table 115. Output fields for the CREATE DB command*

Short label	Keyword	Meaning
CC	N/A	Completion code.
CCTXT	<i>error</i>	Completion code text that briefly explains the meaning of the non-zero completion code.
DB	DB	Database name.
MBR	N/A	IMSplex member that built the output line.

**Return, reason, and completion codes**

The following table includes the return and reason codes and a brief explanation of the codes. The return or reason code returned for the command might also indicate an error from a CSL request.

Table 116. Return and reason codes for CREATE DB

Return code	Reason code	Meaning
X'00000000'	X'00000000'	Command completed successfully. The command output contains a line for each resource, accompanied by its completion code. See the completion code table for details.
X'00000008'	X'00002048'	Invalid SET attribute.
X'00000008'	X'00002117'	Invalid LIKE descriptor name.
X'00000008'	X'00002118'	Invalid LIKE resource name.
X'0000000C'	X'00003000'	Command was successful for some resources but failed for others. The command output contains a line for each resource, accompanied by its completion code. See the completion code table for details.
X'0000000C'	X'00003004'	Command was not successful for any of the resources. The command output contains a line for each resource, accompanied by its completion code. See the completion code table for details.
X'00000010'	X'0000400C'	Command is not valid on the XRF alternate.
X'00000010'	X'00004014'	Command is not valid on the RSR tracker.
X'00000010'	X'00004204'	Model is quiesced. Cannot quiesce the model.
X'00000010'	X'00004120'	Online change phase is in progress.
X'00000010'	X'00004300'	Command is not allowed because online change for MODBLKS is enabled (DFSDFxxx or DFSCGxxx defined with MODBLKS=OLC, or MODBLKS not defined).
X'00000010'	X'00004301'	Command is not allowed because the database is defined RESIDENT(Y), the DMB indicates there is a logical relationship for the database, but the other database is not defined.
X'00000014'	X'00005004'	DFSOCMD response buffer could not be obtained.
X'00000014'	X'00005008'	DFSPOOL storage could not be obtained.
X'00000014'	X'0000500C'	AWE could not be obtained.
X'00000014'	X'00005010'	Unable to obtain latch.
X'00000014'	X'00005114'	Resource or descriptor that is specified as a model is in the process of being imported from the change list in the IMSRSC repository or was not imported successfully from the change list at the end of IMS restart. This resource or descriptor cannot be referenced as a model on a CREATE command until it is successfully imported from the repository.

Errors unique to the processing of this command are returned as completion codes. The following table includes an explanation of the completion codes.

Table 117. Completion codes for the CREATE DB command

Completion code	Completion code text	Meaning
0		Command completed successfully for database.
11	RESOURCE ALREADY EXISTS	Database exists.

Table 117. Completion codes for the CREATE DB command (continued)

Completion code	Completion code text	Meaning
17	ANOTHER CMD IN PROGRESS	Quiescing the default descriptor failed because it was being referenced by another command.
5F	INVALID CHARACTERS IN NAME	Name is invalid because it contains an invalid character.
60	GETMAIN STORAGE ERROR	A GETMAIN request for storage was unsuccessful.
61	DFSBCB STORAGE ERROR	Request to get BCB storage for DDIR control block failed.
65	DMB POOL STORAGE ERROR	During a CREATE command for a DB resource, there was no storage available in the DMB pool to load a resident DMB.
8A	WILDCARD PARAMETER NOT SUPPORTED	The CREATE command does not support wildcard parameters. You must explicitly specify the names of the resources you want to create.
90	INTERNAL ERROR	An IMS system service failure.
93	RESERVED NAME	Name is reserved. For example, name starts with DFS (except for DFSSAMxx or DFSIVPxx), or is an IMS reserved name such as BASICEDT or WTOR.
BC	MAX 32767 DB EXCEEDED	The CREATE DB command failed, because it attempted to create the 32768th database. A maximum of 32,767 databases are allowed.
DE	ACBLIB READ FAILURE	A read request for a member in ACBLIB failed.
FA	AREA SIZE GREATER THAN BUFF SIZE	A CREATE command for a DEDB failed because one of the areas in the DEDB has a buffer size greater than the system buffer size (BSIZ).
E4	DUPLICATE AREA ALREADY EXISTS	A CREATE command for a DEDB failed because one of the areas in the DEDB already exists.

## Examples

The following are examples of the CREATE DB command:

### Example 1 for CREATE DB command

TSO SPOC input:

```
CREATE DB NAME(DB000001,DB000002,DB000003) SET(ACCTYPE(UPD),RESIDENT(Y))
```

TSO SPOC output:

```

DBName  MbrName  CC
DB000001 IMS1      0
DB000002 IMS1      0
DB000003 IMS1      0

```

OM API input:

```
CMD(CREATE DB NAME(DB000001,DB000002,DB000003) SET(ACCTYPE(UPD),RESIDENT(Y)))
```

OM API output:

```

<imsout>
<ctl>
<omname>OM10M </omname>
<omvsn>1.3.0</omvsn>
<xmlvsn>20 </xmlvsn>
<statime>2006.254 23:22:04.212509</statime>
<stotime>2006.254 23:22:04.267816</stotime>
<staseq>BF6461C25FD1D8C5</staseq>
<stoseq>BF6461C26D528480</stoseq>
<rqsttkn1>USRT011 10162204</rqsttkn1>
<rc>00000000</rc>
<rsn>00000000</rsn>
</ctl>
<cmd>
<master>IMS1 </master>
<userid>USRT011 </userid>
<verb>CRE </verb>
<kwd>DB </kwd>
<input>CREATE DB NAME(DB000001,DB000002,DB000003)
        SET(ACCTYPE(UPD),RESIDENT(Y)) </input>
</cmd>
<cmdrsphdr>
<hdr slbl="DB" llbl="DBName" scope="LCL" sort="a" key="1" scroll="no"
  len="8" dtype="CHAR" align="left" />
<hdr slbl="MBR" llbl="MbrName" scope="LCL" sort="a" key="2" scroll="no"
  len="8" dtype="CHAR" align="left" />
<hdr slbl="CC" llbl="CC" scope="LCL" sort="n" key="0" scroll="yes"
  len="4" dtype="INT" align="right" skipb="no" />
<hdr slbl="CCTXT" llbl="CCText" scope="LCL" sort="n" key="0"
  scroll="yes" len="*" dtype="CHAR" skipb="yes" align="left" />
</cmdrsphdr>
<cmdrspdata>
<rsp>DB(DB000001) MBR(IMS1) CC( 0) </rsp>
<rsp>DB(DB000002) MBR(IMS1) CC( 0) </rsp>
<rsp>DB(DB000003) MBR(IMS1) CC( 0) </rsp>
</cmdrspdata>
</imsout>

```

TSO SPOC input:

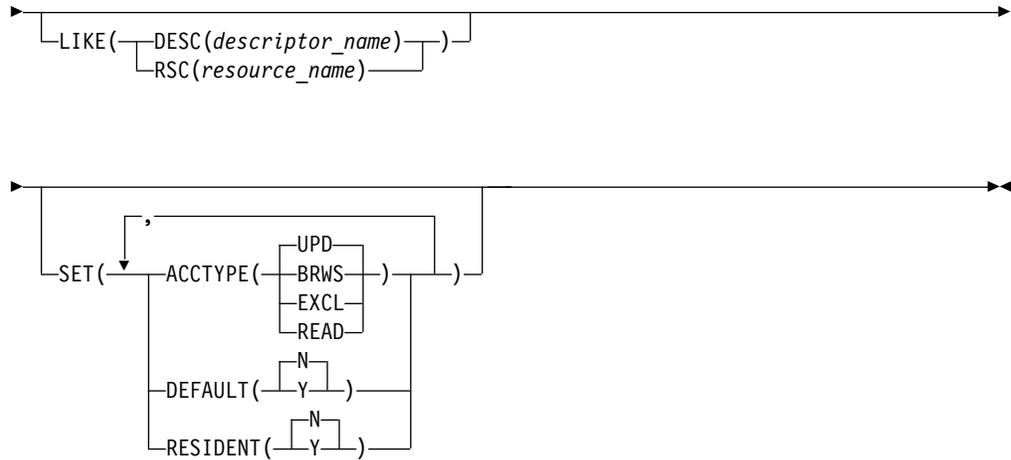
```
QRY DB NAME(DB00*) SHOW(ACCTYPE,RESIDENT,DEFNTYPE,MODEL)
```

TSO SPOC output:

DBName	MbrName	CC	TYPE	LAcc	Rsdnt	LRsdnt	ModelName	ModelType	DefnType
DB000001	IMS1	0		UPD	Y	N	DFSDB1	DESC	CREATE
DB000002	IMS1	0		UPD	Y	N	DFSDB1	DESC	CREATE
DB000003	IMS1	0		UPD	Y	N	DFSDB1	DESC	CREATE

**Explanation:** Several databases are created dynamically, with an access type of update (UPD) and an attribute of resident. Attributes not specified on the command are set to the attributes defined in the database default descriptor. The QRY DB command displays the database attributes specified on the SHOW keyword. The database TYPE is shown as blank for both databases, since there is no DMB in ACBLIB and IMS cannot tell what the database type is. The resident





Default values shown in this syntax diagram reflect the default values that are defined in the IMS database descriptor, DFSDSDB1. If you define another descriptor as the default with a CREATE or UPDATE DBDESC command, the default values shown in this syntax diagram might not match the value defined in the current default descriptor.

## Keywords

The following keywords are valid for the CREATE DBDESC command:

### LIKE

Specifies that the descriptor is created using an existing descriptor as a model, instead of the default descriptor. The default descriptor is either the IMS descriptor DFSDSDB1, or user-defined. The model type can either be a descriptor (DESC), or a resource (RSC). The descriptor is created with all the same attributes as the model, except for the DEFAULT(Y) value. Attributes set explicitly by the CREATE command override the model attributes. DEFAULT(Y) must be specified explicitly to make a descriptor the default descriptor. Later changes to the model are not propagated to resources or descriptors that were created from it.

Use QUERY commands to display the model name and model type used to create the descriptor. The model name and type returned from the QUERY command are provided for informational purposes only. The EXPORT command exports a resource definition including the model name and model type to the definition member. The IMPORT command imports a resource definition including the model name and model type from the definition member. An IMPORT command that creates a resource saves the model name and model type, but does not use the model's attributes.

When you specify LIKE on a CREATE command, IMS uses the model's current attributes to set the attributes of the newly-created descriptor, except for the default attribute.

### DESC(descriptor\_name)

Specifies the name of the descriptor to use as a model to define this descriptor.

**RSC(resource\_name)**

Specifies the name of the resource to use as a model to define this descriptor.

**NAME**

Specifies the 1-8 character name of the database descriptor. Names must be alphanumeric (A through Z, 0 through 9, #, \$, and @). All database descriptor names must begin with an alphabetic character (A through Z, #, \$, and @) followed by 0-7 alphanumeric characters. Names cannot include a blank, comma, period, hyphen, or equal sign. Names cannot begin with DFS, except for DFSIVPxx and DFSSAMxx. Names cannot be BASICEDT, DBCDM, DBRC, ISCEDT, MSDB, SDB, or WTOR. A database resource and a database descriptor can have the same name.

**SET**

Specifies the attributes of the database to be created. If the LIKE keyword is omitted, attributes not specified take the value defined in the default descriptor, which is either IMS descriptor DFSDSDB1 or user-defined. If the LIKE keyword is specified, attributes not specified take the value defined in the model.

**ACCTYPE()**

Specifies the access intent for the named database, which is how the subsystem requesting access plans to use the database. Access type is used in conjunction with the database sharing level declared to DBRC.

**UPD**

The database is available for update as well as read processing in the IMS subsystem. This is the default.

**BRWS**

The database is available for read-only processing on this IMS subsystem. The only programs that can use the database on this subsystem are those that have a PCB processing option of GO (PROCOPT=GO). Programs that access the data using the GO processing option might see uncommitted data since a sharing IMS subsystem could be updating the database. The database is opened for read-only processing.

**EXCL**

The database is to be used exclusively by this IMS subsystem. This exclusive access is guaranteed only when the database is registered to DBRC.

**READ**

The database is available for read-only processing in this IMS subsystem. Programs with update intent can be scheduled, but cannot update the database. Access type READ differs from access type BRWS in that the data is read with integrity (locking is performed) and all programs can access the data, not just those with a processing option of GO. The database is opened for read-only processing.

**DEFAULT**

Specifies whether the descriptor is the default.

**N** The descriptor is not the default.

**Y** The descriptor is the default, which resets the existing default descriptor to DEFAULT(N). When a descriptor is created without the LIKE keyword, any attribute not specified on the CREATE command takes the value defined in the default descriptor. Only one descriptor

can be defined as the default for a resource type. IMS defines a default database descriptor called DFSDSDB1, where all attributes are defined with the default value. Defining a user-defined descriptor as the default overrides the IMS-defined descriptor. Since only one database descriptor can be the default at one time, only one database name may be specified with DEFAULT(Y).

### RESIDENT

Specifies the resident option. The RESIDENT(N) option takes effect right away. The RESIDENT(Y) option takes effect at the next restart, unless the database was created as RESIDENT(Y) after the checkpoint from which this IMS is performing emergency restart. A database defined as a DEDB in ACBLIB always sets the RESIDENT(Y) attribute when the DEDB is loaded, regardless of the RESIDENT value specified. RESIDENT(N) is rejected for a DEDB

- N** The DMB associated with the named database resource is not made resident in storage. The DMB is loaded at scheduling time.
- Y** The DMB associated with the named database resource is made resident in storage at the next IMS restart. At the next IMS restart, IMS loads the DMB and initializes it. A resident database is accessed from local storage, which eliminates I/O to the ACBLIB. In an online environment, the DMB control blocks are stored in the ACBLIB. If the DLI/SAS address space exists, DLI/SAS loads the DMB; otherwise, it is the IMS control region that loads it. This makes the DMB dependent on the existence of the corresponding database resource.

## Usage notes

Attributes not explicitly specified on the CREATE command take the default value. Any database resource or descriptor can be created using this descriptor as a model, by specifying the CREATE command with LIKE(DESC(*descriptor\_name*)).

Descriptors exist for the life of the IMS unless they are deleted using a DELETE command. The descriptors are recoverable across an IMS warm start or emergency restart. Descriptors are lost if IMS is cold started, unless cold start imports definitions that were exported while IMS was up.

The CREATE DBDESC command can only be issued through the Operations Manager API. This command applies to DB/DC and DBCTL systems.

This command is not valid on the XRF alternate, RSR tracker, or FDBR region. The CREATE DBDESC command is not valid if online change for MODBLKS is enabled (DFSDFxxx or DFSCGxxx defined with MODBLKS=OLC or MODBLKS not defined).

## Output fields

The following table shows the CREATE DBDESC output fields. The columns in the table are as follows:

### Short label

Contains the short label generated in the XML output.

### Keyword

Identifies keyword on the command that caused the field to be generated.

N/A appears for output fields that are always returned. *error* appears for output fields that are returned only in case of an error.

**Meaning**

Provides a brief description of the output field.

*Table 119. Output fields for the CREATE DBDESC command*

Short label	Keyword	Meaning
CC	N/A	Completion code.
CCTXT	<i>error</i>	Completion code text that briefly explains the meaning of the non-zero completion code.
DESC	DBDESC	Database descriptor name.
MBR	N/A	IMSplex member that built the output line.
OLDDEF	DBDESC	Old default descriptor name, if this descriptor is made the default by specifying DEFAULT(Y). The old default descriptor is no longer the default.

**Return, reason, and completion codes**

The following table includes the return and reason codes and a brief explanation of the codes. The return or reason code returned for the command might also indicate an error from a CSL request.

*Table 120. Return and reason codes for CREATE DBDESC*

Return code	Reason code	Meaning
X'00000000'	X'00000000'	Command completed successfully. The command output contains a line for each descriptor, accompanied by its completion code. See the completion code table for details.
X'00000008'	X'00002048'	Invalid SET attribute.
X'00000008'	X'00002117'	Invalid LIKE descriptor name.
X'00000008'	X'00002118'	Invalid LIKE resource name.
X'00000008'	X'00002133'	Multiple name parameters specified with DEFAULT(Y) is invalid. Only one descriptor may be the default at one time.
X'0000000C'	X'00003000'	Command was successful for some descriptors but failed for others. The command output contains a line for each descriptor, accompanied by its completion code. See the completion code table for details.
X'0000000C'	X'00003004'	Command was not successful for any of the descriptors. The command output contains a line for each descriptor, accompanied by its completion code. See the completion code table for details.
X'00000010'	X'0000400C'	Command is not valid on the XRF alternate.
X'00000010'	X'00004014'	Command is not valid on the RSR tracker.
X'00000010'	X'00004120'	Online change phase is in progress.
X'00000010'	X'00004204'	Model is quiesced. Cannot quiesce model.

Table 120. Return and reason codes for CREATE DBDESC (continued)

Return code	Reason code	Meaning
X'00000010'	X'00004300'	Command is not allowed because online change for MODBLKS is enabled (DFSDFxxx or DFSCGxxx defined with MODBLKS=OLC, or MODBLKS not defined).
X'00000010'	X'00004301'	Command is not allowed because the database is defined RESIDENT(Y), the DMB indicates there is a logical relationship for the database, but the other database is not defined.
X'00000014'	X'00005004'	DFSOCMD response buffer could not be obtained.
X'00000014'	X'00005008'	DFSPOOL storage could not be obtained.
X'00000014'	X'0000500C'	AWE could not be obtained.
X'00000014'	X'00005010'	Unable to obtain latch.
X'00000014'	X'00005114'	Resource or descriptor that is specified as a model is in the process of being imported from the change list in the IMSRSC repository or was not imported successfully from the change list at the end of IMS restart. This resource or descriptor cannot be referenced as a model on a CREATE command until it is successfully imported from the repository.

Errors unique to the processing of this command are returned as completion codes. The following table includes an explanation of the completion codes.

Table 121. Completion codes for the CREATE DBDESC command

Completion code	Completion code text	Meaning
0		Command completed successfully for database descriptor.
11	RESOURCE ALREADY EXISTS	Database descriptor already exists.
17	ANOTHER CMD IN PROGRESS	Quiescing the default descriptor failed because it was being referenced by another command.
5F	INVALID CHARACTERS IN NAME	Name is invalid because it contains an invalid character.
60	GETMAIN STORAGE ERROR	A GETMAIN request for storage was unsuccessful.
61	DFSBCB STORAGE ERROR	Request to get BCB storage for DDIR control block failed.
8A	WILDCARD PARAMETER NOT SUPPORTED	You must explicitly specify the names of the descriptors you want to create. The CREATE command does not support wildcard parameters.
90	INTERNAL ERROR	An IMS system service failure.
93	RESERVED NAME	Name is reserved. For example, name starts with DFS (except for DFSSAMxx or DFSIVPxx), or is an IMS reserved name such as BASICEDT or WTOR.

## Examples

The following are examples of the CREATE DBDESC command:

### *Example 1 for CREATE DBDESC command*

In this example, the database descriptors listed in the command with the NAME keyword are created with update access. Attributes not specified on the command are assigned the system defaults.

TSO SPOC input:

```
CREATE DBDESC NAME(DBHDAM1,DBHDAM2,DBHDAM3,DBHDAM4) SET(ACCTYPE(UPD))
```

TSO SPOC output:

DBName	MbrName	CC
DBHDAM1	IMS1	0
DBHDAM2	IMS1	0
DBHDAM3	IMS1	0
DBHDAM4	IMS1	0

### **Related concepts:**

 [How to interpret CSL request return and reason codes \(System Programming APIs\)](#)

### **Related reference:**

Chapter 2, “Command keywords and their synonyms,” on page 63

---

## CREATE IMSCON commands

Use the CREATE IMSCON commands to add IMS Connect port or data store definitions during runtime processing without having to restart your IMS Connect instance.

The TYPE keyword specifies the type of IMS Connect resource to create.

The CREATE IMSCON command is processed by every IMS Connect to which OM routes the command, whether or not OM has designated a particular IMS Connect as the command master.

Subsections:

- “CREATE IMSCON TYPE(DATASTORE) command”
- “CREATE IMSCON TYPE(IMSPLEX) command” on page 202
- “CREATE IMSCON TYPE(PORT) command” on page 208

### **Related reference:**

 [IMS Connect type-2 commands \(Commands\)](#)

 [HWSCFGxx member of the IMS PROCLIB data set \(System Definition\)](#)

## CREATE IMSCON TYPE(DATASTORE) command

Use the CREATE IMSCON TYPE(DATASTORE) command to add IMS Connect data store definitions during runtime processing without having to restart your IMS Connect instance.

The CREATE IMSCON TYPE(DATASTORE) command is used to add a communications path to an IMS data store from IMS Connect.

You can also create data store definitions by specifying the DATASTORE statement in the HWSCFGxx member of the IMS PROCLIB data set.

Subsections:

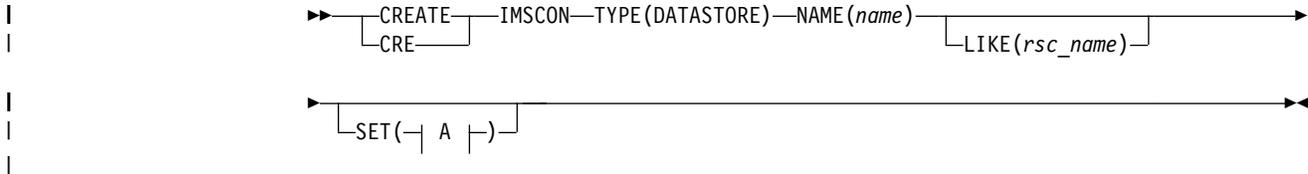
- “Environment”
- “Syntax”
- “Keywords” on page 193
- “Usage notes” on page 197
- “Equivalent WTOR and z/OS commands” on page 197
- “Output fields” on page 197
- “Return, reason, and completion codes” on page 198
- “Return, reason, and completion codes handled by OM” on page 199
- “Examples” on page 199

## Environment

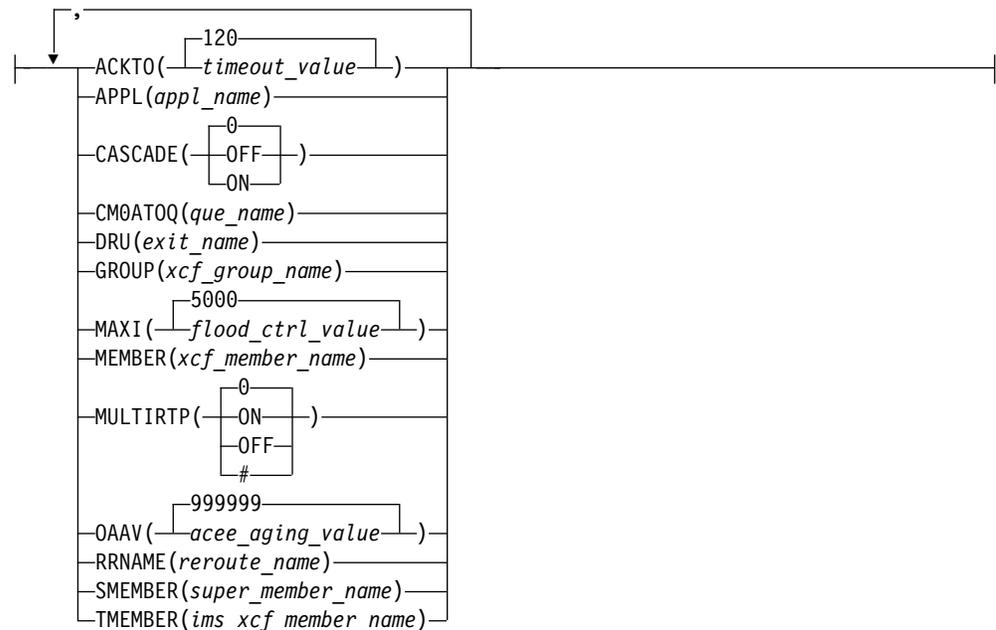
The CREATE IMSCON command is applicable only to IMS Connect. To issue this command, the following conditions must be satisfied:

- IMS Connect must be active and configured to communicate with the Common Service Layer (CSL) Structured Call Interface (SCI).
- A type-2 command environment with Structured Call Interface (SCI) and Operations Manager (OM) must be active.

## Syntax



A:



## Keywords

The following keywords are valid for the `CREATE IMSCON TYPE(DATASTORE)` command:

### LIKE

Specifies the existing IMS Connect data store to use as a model. The new resource is created with all of the same attribute values as the specified model data store.

Attributes that are set explicitly by the `CREATE IMSCON TYPE(DATASTORE)` command override the attribute values that are generated from the model. Later changes to the model are not propagated to resources that were created from it.

### NAME

Specifies the 1- to 8-character name of the IMS data store. The name can consist of alphanumeric characters and must be unique within IMS Connect.

This keyword is equivalent to the `ID` parameter of the `DATASTORE` statement in the `HWSCFGxx` member of the IMS `PROCLIB` data set.

This parameter is required.

### SET

Specifies the attributes of the IMS data store to be created.

If the `LIKE` keyword is omitted, the required attributes must be specified. If an optional attribute is not specified, IMS Connect uses its default value.

If the `LIKE` keyword is specified, the attribute values that are specified with the `SET` keyword override the values that are generated from the model.

**ACKTO**

Specifies the timeout interval for acknowledgements to OTMA for CM0 and CM1 output messages and for IMS-to-IMS transaction messages. The timeout value can be 0 - 255 seconds.

This parameter is optional and defaults to 120. If the timeout value is 0 or is not specified, the OTMA ACK timeout default value of 120 seconds is set.

For IMS-to-IMS transaction messages, if an acknowledgement is not received by OTMA before the timeout interval expires, OTMA reroutes the transaction message to the timeout queue, DFS\$\$TOQ.

**APPL**

Specifies the 1- to 8-character alphanumeric TCP/IP APPL name defined to RACF in the PTKTDATA statement.

This parameter is optional and defaults to blanks. If you are using PassTicket and user message exits, you must specify the APPL parameter.

**CASCADE=**

For this data store connection only, enables or disables support for the cascading of global, two-phase commit (synclevel=syncpoint) z/OS Resource Recovery Services transactions from IMS TM Resource Adapter via a TCP/IP connection to an IMS TM or IMS DB/DC system that is on a different z/OS image (LPAR) than IMS Connect. Cascading transactions that are received via a LOCAL option connection is not supported.

The CASCADE parameter applies only to the support for cascading global, RRS transactions between IMS Connect and IMS TM when they are on different z/OS images. The specification of the IMS Connect CASCADE parameter does not affect support between other components or subsystems, such as between IMS subsystems in a shared queues environment.

The CASCADE specification in the definition of a data store connection overrides the value of CASCADE that is in effect for the IMS Connect system.

Until the next restart of IMS Connect, a CASCADE specification made by the UPDATE IMSCON TYPE(DATASTORE) SET(CASCADE()) command overrides the CASCADE specification on the DATASTORE configuration statement.

- 0 When the LIKE keyword is used to create a data store connection by copying the attributes of an existing data store connection, the numerical character "0" omits the CASCADE specification from the copied attributes. If CASCADE=0 is specified when the LIKE keyword is not used, the result is the same as omitting the CASCADE keyword. In either case, the resulting data store connection inherits the CASCADE value that is in effect for the IMS Connect system configuration.

CASCADE(0) is the default.

**OFF**

Disables support for cascading of global, two-phase commit (synclevel=syncpoint) RRS transactions to an IMS TM or IMS DB/DC system that is running on a different LPAR than IMS Connect. If a synclevel=syncpoint transaction is routed to an IMS system that resides

on a different LPAR than IMS Connect, IMS Connect rejects the transaction with return code 4 and reason code NCASCADE.

When support is disabled, you can send a synclevel=syncpoint transaction to IMS on this data store connection only if IMS Connect and IMS reside on the same LPAR.

**ON** Enables support for cascading of global, two-phase commit (synclevel=syncpoint) RRS transactions to an IMS TM or IMS DB/DC system that is running on different LPAR than IMS Connect.

IMS Connect detects whether the IMS system is on the same LPAR. If IMS Connect and IMS are running on different LPARs, two-phase commit processing uses cascaded transactions. If IMS Connect and IMS are running on the same LPAR, two-phase commit processing does not use cascaded transactions.

#### **CMOATOQ**

Specifies the 1- to 8-character alphanumeric name for the OTMA CM0 ACK timeout queue.

The value that is specified here overrides both the OTMA default value of DFS\$\$TOQ and the value that is set in the HWS statement of the HWSCFGxx member of the IMS PROCLIB data set.

This parameter is optional and defaults to blanks.

#### **DRU**

Specifies the 1- to 8-character alphanumeric name for the OTMA destination resolution user (DRU) exit that is passed to OTMA. The DRU exit is required to support asynchronous output to IMS Connect clients.

This parameter is optional and defaults to DFSYDRU0.

#### **GROUP**

Specifies the 1- to 8-character alphanumeric name of the z/OS cross-system coupling facility (XCF) group for the IMS OTMA. IMS Connect uses this value to join the XCF group.

Because IMS Connect and IMS must be in the same XCF group to communicate, this group name must match the XCF group name that you define to IMS (*GRNAME*) in the IMS startup JCL (for example, "OTMA=Y,GRNAME=&GROUP,USERVAR=&MEMBER",...).

This parameter is required if the LIKE keyword is not specified.

#### **MAXI**

Specifies the OTMA input message flood control value. The valid range is 0 - 9999.

If you specify a value of 0, the OTMA default value of 5000 is used.

If you specify a value 1 - 200, the OTMA minimum value of 200 is used.

This parameter is optional and defaults to 5000.

#### **MEMBER**

Specifies the 1- to 16-character alphanumeric XCF member name that identifies IMS Connect in the XCF group that is specified by the GROUP parameter.

|  
|  
|

This name is the XCF name that IMS uses to communicate with IMS Connect in that XCF group. This XCF member name for IMS Connect must be unique in the data store definitions for all data stores that are members of the same XCF group.

This parameter is required.

#### **MULTIRTP**

Specifies whether IMS Connect requests an OTMA tpipe that supports multiple active resume tpipe requests (MULTIRTP) when IMS Connect establishes the DATASTORE connection with the IMS data store.

Any specified MULTIRTP value other than 0 overrides the MULTIRTP value that is in effect for the IMS Connect system configuration. To accept the MULTIRTP value that is in effect for the IMS Connect system configuration, omit the MULTIRTP parameter or specify MULTIRTP=(0), which is the default.

The following values are valid on the MULTIRTP keyword:

- 0** Specifies that the MULTIRTP value for this data store connection is determined by the IMS Connect system configuration. If the IMS Connect system does not specify a MULTIRTP value, the OTMA client descriptor determines whether the tpipes for this data store connection support multiple active resume tpipe requests.
- #** Specifies that the OTMA client descriptor determines the MULTIRTP value for this data store connection.

#### **OFF**

Disables support for multiple active resume tpipe requests on tpipes for this data store connection.

- ON** Enables support for multiple active resume tpipe requests on tpipes for this data store connection.

#### **OAAV**

Specifies the decimal integer that defines the OTMA accessor environment element (ACEE) aging value, in seconds, for this IMS data store.

When the OTMA ACEE aging value is reached, OTMA refreshes the ACEE before it processes the next input message that is received from IMS Connect.

Valid values are 0 - 999999. If you specify 0, OTMA uses the default value of 999999. If you specify a value 1 - 300, OTMA uses a value of 300 seconds.

This parameter is optional and defaults to 999999.

#### **RRNAME**

Specifies the 1- to 8-character alphanumeric name of the alternative destination of a client reroute request. The name can consist of alphanumeric characters (A - Z, 0 - 9) and special characters (@, #, \$). IMS Connect translates lowercase characters to uppercase characters.

This parameter is optional and defaults to HWS\$DEF.

#### **SMEMBER**

Specifies the 1-4 character name of the OTMA super member to which this IMS data store belongs.

If specified, this value overrides the attribute values for the SMEMBER parameter in the HWS statement of the HWSCFGxx member of the IMS PROCLIB data set.

To disable the value of SMEMBER specified on this IMS data store, specify the parameter with no value, for example, SMEMBER().

This parameter is optional and defaults to blanks.

#### **TMEMBER**

Specifies the 1- to 16-character alphanumeric XCF member name of the IMS that this IMS Connect communicates with in the XCF group.

This target member name must match the member name that IMS uses when it joins the XCF group. The XCF member name for IMS is specified in the IMS startup JCL (for example, "... ,OTMA=Y,GRNAME=&GROUP,OTMANM=&TMEMBER,...").

This parameter is required if the LIKE keyword parameter is not specified.

### **Usage notes**

You can issue the CREATE IMSCON TYPE(DATASTORE) command only through the Operations Manager (OM) API.

IMS Connect can process IMS Connect type-2 commands only if the IMSplex from which the commands were issued has a status of ACTIVE.

When a keyword is specified without any value, that keyword is ignored. IMS Connect processes the command as if the keyword were not specified.

At the completion of the CREATE IMSCON TYPE(DATASTORE) command, IMS Connect starts the IMS data store and sets the IMS data store status to CONNECT if it is successful and DISCONNECT if it is not successful.

IMS Connect resources that are created by use of this command are not saved across restarts of IMS Connect, unless a corresponding configuration statement is also added to the HWSCFGxx member of the IMS PROCLIB data set before the next restart.

### **Equivalent WTOR and z/OS commands**

Equivalent WTOR and z/OS commands that perform similar functions as the CREATE IMSCON TYPE(DATASTORE) command are not available.

### **Output fields**

The following table shows the CREATE IMSCON TYPE(DATASTORE) output fields. The following is a list of the columns that are available in the table:

#### **Short label**

Contains the short label that is generated in the XML output.

#### **Long label**

Contains the column heading for the output field in the formatted output.

#### **Keyword**

Identifies the keyword on the command that caused the field to be generated. N/A is displayed for output fields that are always returned. For output fields that are returned only in case of an error, error is displayed.

## Meaning

Provides a brief description of the output field.

Table 122. Output fields for the CREATE IMSCON TYPE(DATASTORE) command

Short label	Long label	Keyword	Meaning
CC	CC	N/A	Completion code that indicates whether IMS Connect was able to process the command for the specified resource. The completion code is always returned.
CCTXT	CCText	N/A	Completion code text that briefly explains the meaning of the nonzero completion code. This field is returned only for an error completion code.
DS	DataStore	N/A	The data store name. The data store name is always returned.
MBR	MbrName	N/A	Identifier of the IMS Connect that built the output line. The identifier is always returned.

## Return, reason, and completion codes

An IMS Connect return and reason code is returned to OM by the CREATE IMSCON TYPE(DATASTORE) command. The OM return and reason codes that might be returned as a result of the CREATE IMSCON TYPE(DATASTORE) command are standard for all commands that are entered through the OM API.

The following table includes the return and reason codes and a brief explanation of the codes. The return or reason code returned for the command might also indicate an error from a CSL request.

Table 123. Return and reason codes for the CREATE IMSCON TYPE(DATASTORE) command

Return code	Reason code	Meaning
X'00000000'	X'00000000'	The command completed successfully. The command output contains a line for each resource, which is accompanied by its completion code. See the completion codes table for details.
X'0C000008'	X'00002010'	Wildcards (* and %) are not supported.
X'0C000008'	X'00002018'	The required NAME parameter is not specified.
X'0C000008'	X'00002024'	The LIKE keyword parameter value is invalid. The resource cannot be found.
X'0C000008'	X'0000202C'	The resource already exists.
X'0C000008'	X'0000204C'	The required GROUP parameter is not specified.
X'0C000008'	X'00002050'	The required MEMBER parameter is not specified.
X'0C000008'	X'00002054'	The required TMEMBER parameter is not specified.
X'0C000008'	X'0000207C'	The MEMBER parameter value is a duplicate of an existing DATASTORE MEMBER value.
X'0C000008'	X'00002133'	Multiple NAME parameter values are not allowed.
X'0C00000C'	X'00003000'	The command was successful for some resources but failed for others. The command output contains a line for each resource, which is accompanied by its completion code.

Table 123. Return and reason codes for the CREATE IMSCON TYPE(DATASTORE) command (continued)

Return code	Reason code	Meaning
X'0C00000C'	X'00003004'	The command was not successful for any resource. The command output contains a line for each resource, which is accompanied by its completion code.

The following table includes an explanation of the completion codes. Errors unique to the processing of this command are returned as completion codes. A completion code is returned for each action against an individual resource.

Table 124. Completion codes for the CREATE IMSCON TYPE(DATASTORE) command

Completion code	Completion code text	Meaning
0		The CREATE IMSCON TYPE(DATASTORE) command completed successfully.
13	The resource already exists.	The data store already exists.
3C	Resource created, start failed	The CREATE command has created the definition of the requested resource in IMS Connect, but the resource failed to start.  The command output might also contain one of the following messages, which describe why the resource failed to start: <ul style="list-style-type: none"> <li>• HWSD0222W</li> <li>• HWSD0292I</li> <li>• HWSO1205W</li> <li>• HWSO1210W</li> <li>• HWSO1215W</li> <li>• HWSO1220W</li> <li>• HWSO1305W</li> <li>• HWSO1310W</li> <li>• HWSO1315W</li> <li>• HWSO1320W</li> <li>• HWSO1325W</li> </ul>

## Return, reason, and completion codes handled by OM

IMS Connect registers its type-2 command set with OM. When OM detects a command parsing error, OM returns a return code and reason code to the requester, and the command is not routed to IMS Connect.

Clients such as the TSO SPOC send commands to OM by using the CSLOMCMD API. CSLOMCMD return and reason codes are listed in the description of the CSLOMCMD request.

## Examples

The following are examples of the CREATE IMSCON TYPE(DATASTORE) command:

### Example 1 for CREATE IMSCON TYPE(DATASTORE) command

TSO SPOC input:

```
CREATE IMSCON TYPE(DATASTORE) NAME(IMS2)
SET(GROUP(XCFGRP1),MEMBER(ICON1),TMEMBER(IMS2))
```

TSO SPOC output:

DataStore	MbrName	CC
IMS2	HWS1	0

OM API input:

```
CMD(CRE IMSCON TYPE(DATASTORE) NAME(IMS2)
SET(GROUP(XCFGRP1),MEMBER(ICON1),TMEMBER(IMS2)))
```

OM API output:

```
<?xml version="1.0"?>
<!DOCTYPE imsout SYSTEM "imsout.dtd">
<imsout>
<ctl>
<omname>OM10M </omname>
<omvsn>1.6.0</omvsn>
<xmlvsn>20 </xmlvsn>
<statime>2012.314 00:31:32.035293</statime>
<stotime>2012.314 00:31:32.035883</stotime>
<staseq>CA7155B83AADD297</staseq>
<stoseq>CA7155B83AD2B797</stoseq>
<rqsttkn1>USRT011 10163132</rqsttkn1>
<rc>00000000</rc>
<rsn>00000000</rsn>
</ctl>
<cmd>
<master>HWS1 </master>
<userid>USRT011 </userid>
<verb>CRE </verb>
<kwd>IMSCON </kwd>
<input>CREATE IMSCON TYPE(DATASTORE) NAME(IMS2) SET(GROUP(XCFGRP1),MEMBER(ICON1),
TMEMBER(IMS2)) </input>
</cmd>
<cmdrsphdr>
<hdr s1b1="DS" l1b1="DataStore" scope="LCL" sort="a" key="1" scroll="no" len="8"
dtype="CHAR" align="left" skipb="no" />
<hdr s1b1="MBR" l1b1="MbrName" scope="LCL" sort="a" key="2" scroll="no" len="8"
dtype="CHAR" align="left" skipb="no" />
<hdr s1b1="CC" l1b1="CC" scope="LCL" sort="n" key="0" scroll="yes" len="4"
dtype="INT" align="right" skipb="no" />
<hdr s1b1="CCTXT" l1b1="CCText" scope="LCL" sort="n" key="0" scroll="yes" len="32"
dtype="CHAR" align="left" skipb="yes" />
</cmdrsphdr>
<cmdrspdata>
<rsp>DS(IMS2 ) MBR(HWS1 ) CC( 0) </rsp>
</cmdrspdata>
</imsout>
```

**Explanation:** Data store IMS2 is created with the specified attributes.

### Example 2 for CREATE IMSCON TYPE(DATASTORE) command

TSO SPOC input:

```
CREATE IMSCON TYPE(DATASTORE) NAME(IMS2) LIKE(IMS1)
SET(GROUP(XCFGRP1),MEMBER(ICON1B),TMEMBER(IMS2),ACKTO(60),
SMEMBER(SM02))
```

TSO SPOC output:

```
DataStore MbrName      CC
IMS2      HWS1         0
```

OM API input:

```
CMD(CRE IMSCON TYPE(DATASTORE) NAME(IMS2) LIKE(IMS1)
SET(GROUP(XCFGRP1),MEMBER(ICON1B),TMEMBER(IMS2),ACKTO(60),
SMEMBER(SM02)))
```

OM API output:

```
<?xml version="1.0"?>
<!DOCTYPE imsout SYSTEM "imsout.dtd">
<imsout>
<ctl>
<omname>OM10M </omname>
<omvsn>1.6.0</omvsn>
<xmlvsn>20 </xmlvsn>
<statime>2012.314 00:55:31.481740</statime>
<stotime>2012.314 00:55:31.482361</stotime>
<staseq>CA715B14FE08C99D</staseq>
<stoseq>CA715B14FE2F921D</stoseq>
<rqsttkn1>USRT011 10165531</rqsttkn1>
<rc>00000000</rc>
<rsn>00000000</rsn>
</ctl>
<cmd>
<master>HWS1 </master>
<userid>USRT011 </userid>
<verb>CRE </verb>
<kwd>IMSCON </kwd>
<input>CREATE IMSCON TYPE(DATASTORE) NAME(IMS2) LIKE(IMS1) SET(GROUP(XCFGRP1),
MEMBER(ICON1B),TMEMBER(IMS2),ACKTO(60),
SMEMBER(SM02)) </input>
</cmd>
<cmdrsphdr>
<hdr s1b1="DS" l1b1="DataStore" scope="LCL" sort="a" key="1" scroll="no" len="8"
dtype="CHAR" align="left" skipb="no" />
<hdr s1b1="MBR" l1b1="MbrName" scope="LCL" sort="a" key="2" scroll="no" len="8"
dtype="CHAR" align="left" skipb="no" />
<hdr s1b1="CC" l1b1="CC" scope="LCL" sort="n" key="0" scroll="yes" len="4"
dtype="INT" align="right" skipb="no" />
<hdr s1b1="CCTXT" l1b1="CCText" scope="LCL" sort="n" key="0" scroll="yes" len="32"
dtype="CHAR" align="left" skipb="yes" />
</cmdrsphdr>
<cmdrspdata>
<rsp>DS(IMS2 ) MBR(HWS1 ) CC( 0) </rsp>
</cmdrspdata>
</imsout>
```

**Explanation:** Data store IMS2 is created with the same attribute values as data store IMS1. However, the GROUP, MEMBER, TMEMBER, ACKTO, and SMEMBER attributes are also specified, and their values override the values from the IMS1 data store model.

**Related reference:**

-  CSLOMCMDC: command request (System Programming APIs)
-  IMSPLEX statement (System Definition)
-  DATASTORE statement (System Definition)



This parameter is required.

### **SET**

Specifies the attributes of the IMSplex to be created.

If the LIKE keyword is omitted, the required attributes must be specified. If an optional attribute is not specified, IMS Connect uses its default value.

If the LIKE keyword is specified, the attribute values that are specified with the SET keyword override the values that are generated from the model.

### **MEMBER**

Specifies a 1- to 8-character alphanumeric name that identifies IMS Connect in the IMSplex.

IMS Connect registers this name with SCI. SCI uses the name to manage communications between IMS Connect and other IMSplex members, such as Operations Manager (OM), ODBM, or MSC. The name must start with an alphabetic character.

This parameter is required if the LIKE keyword parameter is not specified.

## **Usage notes**

You can issue the CREATE IMSCON TYPE(IMSPLEX) command only through the OM API.

IMS Connect can process IMS Connect type-2 commands only if the IMSplex from which the commands were issued has a status of ACTIVE.

The CREATE IMSCON TYPE(IMSPLEX) command is equivalent to adding an SCI communications path for an IMSplex by specifying the IMSPLEX statement in the IMS Connect configuration PROCLIB member.

At the completion of the CREATE IMSCON TYPE(IMSPLEX) command, IMS Connect starts the communications path to the IMSplex and sets the IMSPLEX status to ACTIVE if it is successful and NOTACTIVE if it is not successful.

**Important:** Type-2 commands require IMS Connect to have a communications path to an IMSplex. Your IMS Connect must already have a communications path to the SCI of an IMSplex to be able to issue a CREATE IMSCON TYPE(IMSPLEX) command to add a communications path to a different IMSplex.

Modifications that are made to IMS Connect resources by use of this command are not saved across restarts of IMS Connect.

## **Equivalent WTOR and z/OS commands**

There are no equivalent WTOR and z/OS commands that perform similar functions as the CREATE IMSCON TYPE(IMSPLEX) command.

## **Output fields**

The following table shows the CREATE IMSCON TYPE(IMSPLEX) output fields. The columns in the table are:

### **Short label**

Contains the short label that is generated in the XML output.

**Long label**

Contains the column heading for the output field in the formatted output.

**Keyword**

Identifies the keyword on the command that caused the field to be generated. N/A is displayed for output fields that are always returned. *error* is displayed for output fields that are returned only in case of an error.

**Meaning**

Provides a brief description of the output field.

Table 125. Output fields for the CREATE IMSCON TYPE(IMSPLEX) command

Short label	Long label	Keyword	Meaning
CC	CC	N/A	Completion code that indicates whether IMS Connect was able to process the command for the specified resource. The completion code is always returned.
CCTXT	CCText	N/A	Completion code text that briefly explains the meaning of the nonzero completion code. This field is returned only for an error completion code.
IMSPLX	IMSplex	N/A	The IMSplex name. The IMSplex name is always returned.
MBR	MbrName	N/A	Identifier of the IMS Connect that built the output line. The identifier is always returned.

**Return, reason, and completion codes**

An IMS Connect return and reason code is returned to OM by the CREATE IMSCON TYPE(IMSPLEX) command. The OM return and reason codes that might be returned as a result of the CREATE IMSCON TYPE(IMSPLEX) command are standard for all commands that are entered through the OM API.

The following table includes the return and reason codes and a brief explanation of the codes. The return or reason code returned for the command might also indicate an error from a CSL request.

Table 126. Return and reason codes for the CREATE IMSCON TYPE(IMSPLEX) command

Return code	Reason code	Meaning
X'00000000'	X'00000000'	The command completed successfully. The command output contains a line for each resource, which is accompanied by its completion code. For details, see Table 127 on page 205.
X'0C000008'	X'00002024'	The LIKE keyword parameter value is invalid. The resource cannot be found.
X'0C000008'	X'00002050'	The required MEMBER parameter is not specified.
X'0C000008'	X'00002098'	The specified MEMBER value is too long. IMSplex member name is maximum of 8 characters.
X'0C00000C'	X'00003000'	The command was successful for some resources but failed for others. The command output contains a line for each resource, which is accompanied by its completion code.

Table 126. Return and reason codes for the CREATE IMSCON TYPE(IMSPLEX) command (continued)

Return code	Reason code	Meaning
X'0C00000C'	X'00003004'	The command was not successful for any resource. The command output contains a line for each resource, which is accompanied by its completion code.

The following table includes an explanation of the completion codes. Errors unique to the processing of this command are returned as completion codes. A completion code is returned for each action against an individual resource.

Table 127. Completion codes for the CREATE IMSCON TYPE(IMSPLEX) command

Completion code	Completion code text	Meaning
0		The CREATE IMSCON TYPE(IMSPLEX) command completed successfully.
13	Resource already exists.	IMSpIex already exists.
1A	BPEGETM error	IMS Connect encountered an error while attempting to obtain storage from the BPEGETM service.
1C	FWEB creation failed	IMS Connect encountered an error while attempting to obtain a block of storage of type FWEB from the BPECBGET service.
1D	CVBB creation failed	IMS Connect encountered an error while attempting to obtain a block of storage of type CVBB from the BPECBGET service.
3C	Resource created, start failed	The CREATE command created a definition of the requested resource in IMS Connect, but the resource failed to start.  The command output might also contain the following messages, which describe why the resource failed to start: <ul style="list-style-type: none"> <li>• HWSI1705W</li> <li>• HWSI1720W</li> <li>• HWSI1815W</li> <li>• HWSI1816W</li> </ul>
3E	Wildcard parameter not supported	The CREATE command does not support wildcard parameters. You must explicitly specify the names of the IMSpIex you want to create.
3F	DCTB creation failed	IMS Connect encountered an error while attempting to obtain a block of storage of type DCTB from the BPECBGET service.
40	Get latch failed	IMS Connect failed to obtain a latch.

Table 127. Completion codes for the CREATE IMSCON TYPE(IMSPLEX) command (continued)

Completion code	Completion code text	Meaning
41	NAME value is too long	The specified NAME value is too long. IMSPlex name is maximum of 5 characters.

## Return, reason, and completion codes handled by OM

IMS Connect registers its type-2 command set with OM. When OM detects a command parsing error, OM returns a return code and reason code to the requester, and the command is not routed to IMS Connect.

Clients such as the TSO SPOC send commands to OM by using the CSLOMCMDB API. CSLOMCMDB return and reason codes are listed in the description of the CSLOMCMDB request.

## Examples

The following are examples of the CREATE IMSCON TYPE(IMSPLEX) command:

### Example 1 for CREATE IMSCON TYPE(IMSPLEX) command

TSO SPOC input:

```
CREATE IMSCON TYPE(IMSPLEX) NAME(PLEX2)
SET(MEMBER(ICON1))
```

TSO SPOC output:

```
IMSPlex MbrName      CC
PLEX2   HWS1         0
```

OM API input:

```
CMD(CRE IMSCON TYPE(IMSPLEX) NAME(PLEX2)
SET(MEMBER(ICON1)))
```

OM API output:

```
<?xml version="1.0"?>
<!DOCTYPE imsout SYSTEM "imsout.dtd">
<imsout>
<ctl>
<omname>OM10M </omname>
<omvsn>1.7.0</omvsn>
<xmlvsn>20 </xmlvsn>
<statime>2014.246 20:20:49.113887</statime>
<stotime>2014.246 20:20:49.165217</stotime>
<staseq>CDB3F4183E71FA55</staseq>
<stoseq>CDB3F4184AFA1FD5</stoseq>
<rqsttkn1>USRT011 10132049</rqsttkn1>
<rc>00000000</rc>
<rsn>00000000</rsn>
</ctl>
<cmd>
<master>HWS1 </master>
<userid>USRT011 </userid>
<verb>CRE </verb>
<kwd>IMSCON </kwd>
<input>CREATE IMSCON TYPE(IMSPLEX) NAME(PLEX2) SET(MEMBER(ICON1))
```

```

| </input>
| </cmd>
| <cmdrsphdr>
| <hdr s1b1="IMSPLEX" l1b1="IMSpIex" scope="LCL" sort="a" key="1" scroll="no" len="8"
|   dtype="CHAR" align="left" skipb="no" />
| <hdr s1b1="MBR" l1b1="MbrName" scope="LCL" sort="a" key="2" scroll="no" len="8"
|   dtype="CHAR" align="left" skipb="no" />
| <hdr s1b1="CC" l1b1="CC" scope="LCL" sort="n" key="0" scroll="yes" len="4"
|   dtype="INT" align="right" skipb="no" />
| <hdr s1b1="CCTXT" l1b1="CCText" scope="LCL" sort="n" key="0" scroll="yes"
|   len="32" dtype="CHAR" align="left" skipb="yes" />
| </cmdrsphdr>
| <cmdrspdata>
| <rsp>IMSPLEX(PLEX2 ) MBR(HWS1 ) CC( 0) </rsp>
| </cmdrspdata>
| </imsout>

```

**Explanation:** IMSpIex PLEX2 is created with the specified attributes.

### *Example 2 for CREATE IMSCON TYPE(IMSPLEX) command*

TSO SPOC input:

```
CREATE IMSCON TYPE(IMSPLEX) NAME(PLEX2) LIKE(PLEX1)
```

TSO SPOC output:

```
IMSpIex  MbrName          CC
PLEX2    HWS1              0
```

OM API input:

```
CMD(CRE IMSCON TYPE(IMSPLEX) NAME(PLEX2) LIKE(PLEX1))
```

OM API output:

```

| <?xml version="1.0"?>
| <!DOCTYPE imsout SYSTEM "imsout.dtd">
| <imsout>
| <ctl>
| <omname>OM10M </omname>
| <omvsn>1.7.0</omvsn>
| <xmlvsn>20 </xmlvsn>
| <statime>2014.246 20:38:14.619891</statime>
| <stotime>2014.246 20:38:14.658339</stotime>
| <stoseq>CDB3F7FD50EF3C46</stoseq>
| <stoseq>CDB3F7FD5A523CC4</stoseq>
| <rqsttkn1>USRT011 10133814</rqsttkn1>
| <rc>00000000</rc>
| <rsn>00000000</rsn>
| </ctl>
| <cmd>
| <master>HWS1 </master>
| <userid>USRT011 </userid>
| <verb>CRE </verb>
| <kwd>IMSCON </kwd>
| <input>CREATE IMSCON TYPE(IMSPLEX) NAME(PLEX2) LIKE(PLEX1) </input>
| </cmd>
| <cmdrsphdr>
| <hdr s1b1="IMSPLEX" l1b1="IMSpIex" scope="LCL" sort="a" key="1" scroll="no"
|   len="8" dtype="CHAR" align="left" skipb="no" />
| <hdr s1b1="MBR" l1b1="MbrName" scope="LCL" sort="a" key="2" scroll="no"
|   len="8" dtype="CHAR" align="left" skipb="no" />
| <hdr s1b1="CC" l1b1="CC" scope="LCL" sort="n" key="0" scroll="yes" len="4"
|   dtype="INT" align="right" skipb="no" />
| <hdr s1b1="CCTXT" l1b1="CCText" scope="LCL" sort="n" key="0" scroll="yes"
|   len="32" dtype="CHAR" align="left" skipb="yes" />

```

```

| </cmdrsphdr>
| <cmdrspdata>
| <rsp>IMSPLEX(PLEX2 ) MBR(HWS1 ) CC( 0) </rsp>
| </cmdrspdata>
| </imsout>

```

**Explanation:** IMSplex PLEX2 is created like IMSplex PLEX1 with the MEMBER attribute of the model PLEX1 attribute.

## CREATE IMSCON TYPE(PORT) command

Use the CREATE IMSCON TYPE(PORT) command to add IMS Connect port definitions during runtime processing without having to restart your IMS Connect instance.

The CREATE IMSCON TYPE(PORT) command is used to add a listening port.

Specifying PORTTYPE(REG) creates a regular listening port.

Specifying PORTTYPE(DRDA) creates a DRDA listening port.

*Table 128. Type-2 CREATE IMSCON TYPE(PORT) command and HWSCFGxx member configuration equivalents*

Type-2 command	Configuration for the HWSCFGxx member of the IMS PROCLIB data set
CREATE IMSCON TYPE(PORT) PORTTYPE(REG)	PORT= substatement of the TCPIP statement
CREATE IMSCON TYPE(PORT) PORTTYPE(DRDA)	DRDAPORT= substatement of the ODACCESS statement

Subsections:

- “Environment”
- “Syntax” on page 209
- “Keywords” on page 209
- “Usage notes” on page 211
- “Equivalent WTOR and z/OS commands” on page 212
- “Output fields” on page 212
- “Return, reason, and completion codes” on page 212
- “Return, reason, and completion codes handled by OM” on page 214
- “Examples” on page 214

### Environment

The CREATE IMSCON command is applicable only to IMS Connect. To issue this command, the following conditions must be satisfied:

- IMS Connect must be active and configured to communicate with the Common Service Layer (CSL) Structured Call Interface (SCI).
- A type-2 command environment with Structured Call Interface (SCI) and Operations Manager (OM) must be active.

## Syntax

```
► CREATE IMSCON TYPE(PORT) A
```

**A:**

```
NAME (name) LIKE(rsc_name)
```

```
SET (
  EDITRTN (exit_name)
  IDLETO (timeout_value)
  KEEPAV (value)
  PORTTMOT (value)
  PORTTYPE (REG | DRDA)
)
```

## Keywords

The following keywords are valid for the CREATE IMSCON TYPE(PORT) command:

### LIKE

Specifies the existing IMS Connect port resource to use as a model. The new resource is created with all of the same attribute values as the specified model port resource.

Attributes that are set explicitly by the CREATE IMSCON TYPE(PORT) command override the attribute values that are generated from the model. Later changes to the model are not propagated to resources that were created from it.

### NAME

Specifies the 1- to 5-character decimal field of the port number. Valid values are decimal numbers 1 - 65535.

Port numbers must be unique for a specific instance of IMS Connect and must not conflict with other ports selected in the TCP/IP domain.

This parameter is required.

### SET

Specifies the attributes of the port to be created.

If the LIKE keyword is omitted, the required attributes must be specified. If an optional attribute is not specified, IMS Connect uses its default value.

If the LIKE keyword is specified, the attribute values that are specified with the SET keyword override the values that are generated from the model.

### **EDITRTN**

Specifies the 1- to 8-character name of the IMS Connect Port Message Edit exit routine.

This parameter is equivalent to the EDIT parameter of the PORT substatement of the TCPIP statement of the HWSCFGxx member of the IMS PROCLIB data set.

The exit routine can modify messages that do not conform to the standard message formats for IMS Connect. The exit routine must be accessible to IMS Connect by JOBLIB, STEPLIB, or LinkList.

This parameter is valid only when the (PORTTYPE(REG)) keyword is specified.

When the LIKE keyword is used to create a port by copying the attributes of an existing port, specifying the numerical character "0" omits the EDITRTN specification from the copied attributes and sets it to blanks, thus disabling the exit routine. If EDITRTN(0) is specified when the LIKE keyword is not specified, the result is the same as omitting the EDITRTN keyword.

This parameter is optional and defaults to blanks.

### **IDLETO**

Specifies the timeout interval for open socket connections on this port that are idle and waiting in a RECV state for the next message. When the timeout interval expires, IMS Connect closes the socket connection due to inactivity.

The timeout interval is in hundredths of seconds. Valid values are 0 to 2 147 483 647 (X'7FFFFFFF').

A value of 0 disables this timeout function, which prevents inactive connections from timing out.

If the IDLETO parameter is omitted, the default value is -1, which indicates that the port uses the global-level idle timeout value, if any.

An IDLETO specification in a port definition overrides the global default IDLETO value.

This parameter is valid only when the PORTTYPE(REG) keyword is specified.

### **KEEPAV**

Specifies the number of seconds for the z/OS TCP/IP KeepAlive interval for sockets on this port. TCP/IP accepts a range 1 - 2147460 seconds.

Use the TCP/IP KeepAlive function to detect error situations for inactive sockets. Specify the KEEPAV parameter to override the default TCP/IP KeepAlive interval value.

This parameter is optional and defaults to 0. If you specify 0, the KeepAlive interval value is bypassed, and the setting for the TCP/IP stack is used.

### **PORTTMOT**

Specifies the numeric value of the DRDA port timeout value.

This value is the amount of time that IMS Connect waits for the next input message from a client application that is connected on a DRDA port before IMS Connect disconnects the client.

The timeout interval is specified as a decimal integer in hundredths of a second. Valid values are 0 - 2,147,483,647 (X'7FFFFFFF').

Specifying a value of 0 disables the timeout function.

This parameter is optional and defaults to 6000 (1 minute).

This parameter is valid only when the PORTTYPE(DRDA) keyword is specified.

#### **PORTTYPE**

Specifies the type of port.

This keyword is optional and defaults to PORTTYPE(REG).

#### **DRDA**

Indicates that this port is being created for listening to DRDA messages.

Specifying this parameter is equivalent to creating a port by defining the DRDAPORT substatement in the ODACCESS statement of the HWSCFGxx member of the IMS PROCLIB data set.

#### **REG**

Indicates that this port is being created for listening to regular IMS Connect client messages.

Specifying this parameter is equivalent to creating a port by defining the PORT substatement in the TCPIP statement of the HWSCFGxx member of the IMS PROCLIB data set.

## **Usage notes**

You can issue the CREATE IMSCON TYPE(PORT) command only through the Operations Manager (OM) API.

IMS Connect can process IMS Connect type-2 commands only if the IMSplex from which the commands were issued has a status of ACTIVE.

When a keyword is specified without any value, that keyword is ignored. IMS Connect processes the command as if the keyword were not specified. The following command examples are valid:

- CREATE IMSCON TYPE(PORT) NAME(1111) SET()  
None of the SET attribute keywords are required, so the command is processed as if it were CREATE IMSCON TYPE(PORT) NAME(1111).
- CREATE IMSCON TYPE(PORT) NAME(2222) SET(KEEPAV())  
The command is processed as if it were CREATE IMSCON TYPE(PORT) NAME(2222).
- CREATE IMSCON TYPE(PORT) NAME(3333) LIKE()  
The command is processed as if it were CREATE IMSCON TYPE(PORT) NAME(3333).

At the completion of the CREATE IMSCON TYPE(PORT) command, IMS Connect starts the port and sets the port status to ACTIVE if it is successful and NOT ACTIVE if it is not successful.

If TCP/IP is up, at the successful completion of the CREATE IMSCON TYPE(PORT) command, IMS Connect issues a HWSS0790I message to the system console. If TCP/IP is down, at the successful completion of the CREATE IMSCON TYPE(PORT) command, IMS Connect issues a HWSS0771W message to the system console.

IMS Connect resources that are created by use of this command are not saved across restarts of IMS Connect, unless a corresponding configuration statement is also added to the HWSCFGxx member of the IMS PROCLIB data set before the next restart.

## Equivalent WTOR and z/OS commands

Equivalent WTOR and z/OS commands that perform similar functions as the CREATE IMSCON TYPE(PORT) command are not available.

## Output fields

The following table shows the CREATE IMSCON TYPE(PORT) output fields. The following is a list of the columns that are available in the table:

### Short label

Contains the short label that is generated in the XML output.

### Long label

Contains the column heading for the output field in the formatted output.

### Keyword

Identifies the keyword on the command that caused the field to be generated. N/A is displayed for output fields that are always returned. For output fields that are returned only in case of an error, error is displayed.

### Meaning

Provides a brief description of the output field.

Table 129. Output fields for the CREATE IMSCON TYPE(PORT) command

Short label	Long label	Keyword	Meaning
CC	CC	N/A	Completion code that indicates whether IMS Connect was able to process the command for the specified resource. The completion code is always returned.
CCTXT	CCText	N/A	Completion code text that briefly explains the meaning of the nonzero completion code. This field is returned only for an error completion code.
MBR	MbrName	N/A	Identifier of the IMS Connect that built the output line. The identifier is always returned.
PORT	Port	N/A	The port number. The port number is always returned.

## Return, reason, and completion codes

An IMS Connect return and reason code is returned to OM by the CREATE IMSCON TYPE(PORT) command. The OM return and reason codes that might be returned as a result of the CREATE IMSCON TYPE(PORT) command are standard for all commands that are entered through the OM API.

The following table includes the return and reason codes and a brief explanation of the codes. The return or reason code returned for the command might also indicate an error from a CSL request.

Table 130. Return and reason codes for the CREATE IMSCON TYPE(PORT) command

Return code	Reason code	Meaning
X'00000000'	X'00000000'	The command completed successfully. The command output contains a line for each resource, which is accompanied by its completion code. See the completion codes table for details.
X'0C000008'	X'00002004'	The command contains an invalid primary keyword or no client is registered for the keyword.
X'0C000008'	X'00002010'	Wildcards (* and %) are not supported.
X'0C000008'	X'00002024'	The LIKE keyword parameter value is invalid. The resource cannot be found.
X'0C000008'	X'00002028'	The LIKE keyword parameter value cannot be an SSL port.
X'0C000008'	X'00002058'	IMS Connect could not load the Port Message Edit exit routine.
X'0C000008'	X'00002084'	The EDITRTN keyword is invalid for PORTTYPE(DRDA).
X'0C000008'	X'00002088'	The PORTTMOT keyword is invalid for PORTTYPE(REG).
X'0C000008'	X'000020BC'	Command specified a port type of DRDA. Command was rejected.
X'0C00000C'	X'00003000'	The command was successful for some resources but failed for others. The command output contains a line for each resource, which is accompanied by its completion code.
X'0C00000C'	X'00003004'	The command was not successful for any resource. The command output contains a line for each resource, which is accompanied by its completion code.

The following table includes an explanation of the completion codes. Errors unique to the processing of this command are returned as completion codes. A completion code is returned for each action against an individual resource.

Table 131. Completion codes for the CREATE IMSCON TYPE(PORT) command

Completion code	Completion code text	Meaning
0		The CREATE IMSCON TYPE(PORT) command completed successfully.
13	The resource already exists.	The port already exists.

Table 131. Completion codes for the CREATE IMSCON TYPE(PORT) command (continued)

Completion code	Completion code text	Meaning
39	Resource created, retrying start	<p>IMS Connect created the port resource but was unable to start the port. It continues to retry starting the port in the background.</p> <p>The command output might also contain one of the following messages:</p> <ul style="list-style-type: none"> <li>• HWSP1400W</li> <li>• HWSP1402W</li> <li>• HWSP1405W</li> <li>• HWSP1415I</li> <li>• HWSP1415E</li> <li>• HWSP1420E</li> <li>• HWSP1485E</li> <li>• HWSS0771W</li> <li>• HWSS0777W</li> </ul>
3A	Resource created, sched failed	<p>IMS Connect created the port resource, but was unable to start the port scheduler. The command output also contains the following message: HWSS0712W.</p>
3C	Resource created, start failed	<p>The CREATE command has created the definition of the requested resource in IMS Connect, but the resource failed to start.</p> <p>The command output might also contain the following message, which describes why the resource failed to start: HWSS0770I.</p>

## Return, reason, and completion codes handled by OM

IMS Connect registers its type-2 command set with OM. When OM detects a command parsing error, OM returns a return code and reason code to the requester, and the command is not routed to IMS Connect.

Clients such as the TSO SPOC send commands to OM by using the CSLOMCMD API. CSLOMCMD return and reason codes are listed in the description of the CSLOMCMD request.

## Examples

The following are examples of the CREATE IMSCON TYPE(PORT) command:

### *Example 1 for CREATE IMSCON TYPE(PORT) command*

TSO SPOC input:

```
CREATE IMSCON TYPE(PORT) NAME(8888)
SET(EDITRTN(HWSEDIT0),KEEPAV(120))
```

TSO SPOC output:

Port	MbrName	CC
8888	HWS1	0

OM API input:

```
CMD(CRE IMSCON TYPE(PORT) NAME(8888)
SET(EDITRTN(HWSEDT0),KEEPAV(120)))
```

OM API output:

```
<?xml version="1.0"?>
<!DOCTYPE imsout SYSTEM "imsout.dtd">
<imsout>
<ctl>
<omname>OM10M </omname>
<omvsn>1.6.0</omvsn>
<xmivsn>20 </xmivsn>
<stime>2012.314 16:51:36.534771</stime>
<stotime>2012.314 16:51:36.535211</stotime>
<staseq>CA723167E3C4711B</staseq>
<stoseq>CA7231682AD21F14</stoseq>
<rqsttkn1>USR011 10085423</rqsttkn1>
<rc>00000000</rc>
<rsn>00000000</rsn>
</ctl>
<cmd>
<master>HWS1 </master>
<userid>USR011 </userid>
<verb>CRE </verb>
<kwd>IMSCON </kwd>
<input>CREATE IMSCON TYPE(PORT) NAME(8888) SET(EDITRTN(HWSEDT0),KEEPAV(120)) </input>
</cmd>
<cmdrspHdr>
<hdr s1b1="PORT" l1b1="Port" scope="LCL" sort="a" key="1" scroll="no" len="8"
dtype="CHAR" align="left" skipb="no" />
<hdr s1b1="MBR" l1b1="MbrName" scope="LCL" sort="a" key="2" scroll="no" len="8"
dtype="CHAR" align="left" skipb="no" />
<hdr s1b1="CC" l1b1="CC" scope="LCL" sort="n" key="0" scroll="yes" len="4"
dtype="INT" align="right" skipb="no" />
<hdr s1b1="CCTXT" l1b1="CCText" scope="LCL" sort="n" key="0" scroll="yes" len="32"
dtype="CHAR" align="left" skipb="yes" />
</cmdrspHdr>
<cmdrspdata>
<rsp>PORT(8888 ) MBR(HWS1 ) CC( 0) </rsp>
</cmdrspdata>
</imsout>
```

**Explanation:** Regular port 8888 is created with the EDITRTN and KEEPAV attributes specified.

*Example 2 for CREATE IMSCON TYPE(PORT) command*

TSO SPOC input:

```
CREATE IMSCON TYPE(PORT) NAME(8888) LIKE(9999)
SET(EDITRTN(HWSEDT0))
```

TSO SPOC output:

Port	MbrName	CC
8888	HWS1	0

OM API input:

```
CMD(CRE IMSCON TYPE(PORT) NAME(8888) LIKE(9999)
SET(EDITRTN(HWSEDT0)))
```

OM API output:

```

<?xml version="1.0"?>
<!DOCTYPE imsout SYSTEM "imsout.dtd">
<imsout>
<ctl>
<omname>OM10M </omname>
<omvsn>1.6.0</omvsn>
<xmlvsn>20 </xmlvsn>
<statime>2012.314 17:20:25.453406</statime>
<stotime>2012.314 17:20:25.535260</stotime>
<staseq>CA7237396535EC18</staseq>
<stoseq>CA7237397931C697</stoseq>
<rqsttkn1>USRT011 10092025</rqsttkn1>
<rc>00000000</rc>
<rsn>00000000</rsn>
</ctl>
<cmd>
<master>HWS1 </master>
<userid>USRT011 </userid>
<verb>CRE </verb>
<kwd>IMSCON </kwd>
<input>CREATE IMSCON TYPE(PORT) NAME(8888) LIKE(9999) SET(EDITRTN(HWSEdit0)) </input>
</cmd>
<cmdrspHdr>
<hdr slbl="PORT" llbl="Port" scope="LCL" sort="a" key="1" scroll="no" len="8"
dtype="CHAR" align="left" skipb="no" />
<hdr slbl="MBR" llbl="MbrName" scope="LCL" sort="a" key="2" scroll="no" len="8"
dtype="CHAR" align="left" skipb="no" />
<hdr slbl="CC" llbl="CC" scope="LCL" sort="n" key="0" scroll="yes" len="4"
dtype="INT" align="right" skipb="no" />
<hdr slbl="CCTXT" llbl="CCText" scope="LCL" sort="n" key="0" scroll="yes" len="32"
dtype="CHAR" align="left" skipb="yes" />
</cmdrspHdr>
<cmdrspdata>
<rsp>PORT(8888 ) MBR(HWS1 ) CC( 0) </rsp>
</cmdrspdata>
</imsout>

```

**Explanation:** Regular port 8888 is created with the same attributes as port 9999. However, the EDITRTN attribute is also specified, and its values override the values from the 9999 port model.

### *Example 3 for CREATE IMSCON TYPE(PORT) command*

TSO SPOC input:

```

CREATE IMSCON TYPE(PORT) NAME(8888,9999)
SET(PORTTYPE(DRDA))

```

TSO SPOC output:

Port	MbrName	CC	CCText
8888	HWS1	0	
9999	HWS1	13	Resource already exists

OM API input:

```

CMD(CRE IMSCON TYPE(PORT) NAME(8888,9999)
SET(PORTTYPE(DRDA)))

```

OM API output:

```

<?xml version="1.0"?>
<!DOCTYPE imsout SYSTEM "imsout.dtd">
<imsout>
<ctl>
<omname>OM10M </omname>
<omvsn>1.6.0</omvsn>
<xmlvsn>20 </xmlvsn>
<statime>2012.314 17:30:09.273414</statime>

```

```

<stotime>2012.314 17:30:09.277816</stotime>
<staseq>CA7239662B64659E</staseq>
<stoseq>CA7239662C778D9C</stoseq>
<rqsttkn1>USRT011 10093009</rqsttkn1>
<rc>0200000C</rc>
<rsn>00003008</rsn>
<rsnmsg>CSLN054I</rsnmsg>
<rsntxt>None of the clients were successful.</rsntxt>
</ctl>
<cmderr>
<mbr name="HWS1  ">
<typ>IMSCON </typ>
<rc>0C00000C</rc>
<rsn>00003000</rsn>
<rsntxt>At least one request successful </rsntxt>
</mbr>
</cmderr>
<cmd>
<master>HWS1 </master>
<userid>USRT011 </userid>
<verb>CRE </verb>
<kwd>IMSCON </kwd>
<input>CREATE IMSCON TYPE(PORT) NAME(8888,9999) SET(PORTTYPE(DRDA))
</input>
</cmd>
<cmdrsphdr>
<hdr s1b1="PORT" l1b1="Port" scope="LCL" sort="a" key="1" scroll="no" len="8"
dtype="CHAR" align="left" skipb="no" />
<hdr s1b1="MBR" l1b1="MbrName" scope="LCL" sort="a" key="2" scroll="no" len="8"
dtype="CHAR" align="left" skipb="no" />
<hdr s1b1="CC" l1b1="CC" scope="LCL" sort="n" key="0" scroll="yes" len="4"
dtype="INT" align="right" skipb="no" />
<hdr s1b1="CCTXT" l1b1="CCText" scope="LCL" sort="n" key="0" scroll="yes" len="32"
dtype="CHAR" align="left" skipb="yes" />
</cmdrsphdr>
<cmdrspdata>
<rsp>PORT(8888 ) MBR(HWS1 ) CC( 0) </rsp>
<rsp>PORT(9999 ) MBR(HWS1 ) CC( 13) CCTXT(Resource already exists) </rsp>
</cmdrspdata>
</imsout>

```

**Explanation:** DRDA port 8888 is created without any attribute overrides. DRDA port 9999 is not created and returns completion code 13, which indicates that port 9999 already exists.

#### *Example 4 for CREATE IMSCON TYPE(PORT) command*

TSO SPOC input:

```
CREATE IMSCON TYPE(PORT) NAME(1234) SET(IDLETO(2000))
```

TSO SPOC output:

Port	MbrName	CC
1234	HWS1	0

OM API input:

```
CMD(CREATE IMSCON TYPE(PORT) NAME(1234) SET(IDLETO(2000)))
```

OM API output:

```

<cmd>
<master>HWS1 </master>
<userid>USRT002 </userid>
<verb>CRE </verb>
<kwd>IMSCON </kwd>
<input>CREATE IMSCON TYPE(PORT) NAME(1234) SET(IDLETO(2000)) </input>
</cmd>
<cmdrsphdr>

```

```

| <hdr s1b1="PORT" l1b1="Port" scope="LCL" sort="a" key="1" scroll="no"
|   len="8" dtype="CHAR" align="left" skipb="no" />
| <hdr s1b1="MBR" l1b1="MbrName" scope="LCL" sort="a" key="2" scroll="no"
|   len="8" dtype="CHAR" align="left" skipb="no" />
| <hdr s1b1="CC" l1b1="CC" scope="LCL" sort="n" key="0" scroll="yes"
|   len="4" dtype="INT" align="right" skipb="no" />
| <hdr s1b1="CCTXT" l1b1="CCText" scope="LCL" sort="n" key="0"
|   scroll="yes" len="32" dtype="CHAR" align="left" skipb="yes" />
| </cmdrsphdr>
| <cmdrspdata>
| <rsp>PORT(1234   ) MBR(HWS1           ) CC(   0) </rsp>
| </cmdrspdata>

```

**Related reference:**

- [CSLOMCMD: command request \(System Programming APIs\)](#)
- [IMS Connect Port Message Edit exit routine \(Exit Routines\)](#)
- [IMSPLEX statement \(System Definition\)](#)
- [ODACCESS statement \(System Definition\)](#)
- [TCPIP statement \(System Definition\)](#)

---

## CREATE LTERM command

Use the CREATE LTERM command to create a Multiple Systems Coupling (MSC) remote logical terminal (LTERM) and assign it to an MSC logical link path between two IMS subsystems.

**Subsections:**

- “Environment”
- “Syntax diagram”
- “Keywords” on page 219
- “Usage notes” on page 219
- “Output fields” on page 219
- “Return and reason codes” on page 220
- “Completion codes” on page 221
- “Example of the CREATE LTERM command” on page 221

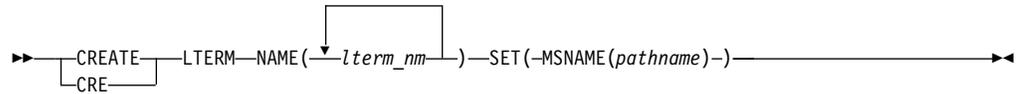
### Environment

The following table lists the environments (DB/BC, DBCTL, and DCCTL) from which the command keywords can be issued.

*Table 132. Valid environments for the CREATE LTERM command and keywords*

Command / Keywords	DB/DC	DBCTL	DCCTL
CREATE LTERM	X		X
NAME	X		X
SET	X		X
TYPE	X		X

### Syntax diagram



## Keywords

The following keywords are valid for the CREATE LTERM command:

### NAME

Specifies one or more 1- to 8-character names of one or more MSC remote logical terminals.

The name can contain only alphanumeric characters: the letters A through Z, digits 0 through 9, and national characters #, \$, and @.

WTOR, DFSMTCNT, and names that begin with the character sequence INQU are invalid names for an LTERM.

The name of a remote logical terminal cannot be the same as the name of a transaction code, a logical link path, or a local LTERM.

This keyword is required.

### SET()

Specifies the attributes of the LTERM to be created.

#### MSNAME

Specifies the 1- to 8-character name of the logical link path to which this remote LTERM is assigned.

## Usage notes

The CREATE LTERM command can be issued only through the Operations Manager API. These commands apply to DB/DC and DCCTL systems. The syntax for this command is defined in XML and is available to automation programs that communicate with OM.

MSC remote logical terminals that are added by the CREATE LTERM command are logged to save them across warm and emergency restarts. If you dynamically created MSC remote logical terminal resources with the CREATE LTERM command and want to restore the definitions at the next IMS cold start, you must also perform one of the following actions:

- Export the resources to the IMSRSC repository by using the automatic export function and import the resources at IMS cold start by using the automatic import function.
- Define the resources with stage-1 system definition macros and regenerate the MSC resources into the DFSCLR0x member in the IMS.SDFSRESL data set.

## Output fields

The following table shows the CREATE LTERM output fields. The columns in the table are as follows:

### Short label

Contains the short label that is generated in the XML output.

### Keyword

Identifies keyword on the command that caused the field to be generated.

N/A appears for output fields that are always returned. The value *error* appears for output fields that are returned only in case of an error.

**Meaning**

Provides a brief description of the output field.

*Table 133. Output fields for the CREATE LTERM command*

Short label	Long label	Keyword	Meaning
CC	CC	N/A	Completion code.
CCTXT	CCText	N/A	Completion code text that briefly explains the meaning of the non-zero completion code.
MBR	Mbrname	N/A	IMSplex member that built the output line.
LTERM	Lterm	N/A	Logical terminal name.

**Return and reason codes**

The OM return and reason codes that might be returned as a result of the CREATE LTERM command are standard for all commands that are entered through the OM.

The following table includes the return and reason codes and a brief explanation of the codes. The return or reason code returned for the command might also indicate an error from a CSL request.

*Table 134. Return and reason codes for the CREATE LTERM command*

Return code	Reason code	Meaning
X'00000000'	X'00000000'	The command completed successfully. The command output contains a line for each resource, which is accompanied by its completion code. See the completion code table for details.
X'00000008'	X'0000211A'	Invalid MSNAME name.
X'00000008'	X'00002157'	Invalid BACKUP parameter.
X'00000008'	X'00002158'	Invalid BUFSIZE value.
X'0000000C'	X'00003000'	Command was successful for some resources but failed for others. The command output contains a line for each resource, which is accompanied by its completion code. See the completion code table for details.
X'0000000C'	X'00003004'	Command was not successful for any of the resources. The command output contains a line for each resource, which is accompanied by its completion code. See the completion code table for details.
X'00000010'	X'0000400C'	Command is not valid on the XRF alternate.
X'00000010'	X'00004014'	Command is not valid on the RSR tracker.
X'00000010'	X'0000402C'	Command is not valid because MSC is not enabled in the IMS system.
X'00000010'	X'0000402D'	Command is not allowed because dynamic definition is not enabled for MSC. MSCRSCS=DYN is not specified in the MSC section of the DFSDFxxx member of the IMS PROCLIB data set.
X'00000014'	X'00005004'	A DFSOCMD response buffer could not be obtained.

Table 134. Return and reason codes for the CREATE LTERM command (continued)

Return code	Reason code	Meaning
X'00000014'	X'00005008'	DFSPOOL storage could not be obtained.
X'00000014'	X'00005010'	Unable to obtain latch.

## Completion codes

The following table includes an explanation of the completion codes.

Table 135. Completion codes for the CREATE LTERM command

Completion code	Completion code text	Meaning
0		The command completed successfully for the remote logical terminal.
11	RESOURCE ALREADY EXISTS	A remote logical terminal with the same name already exists.
17	ANOTHER CMD IN PROGRESS	Another command is in progress for the LTERM. For example, if an LTERM name is specified twice in a CREATE LTERM command, the second occurrence of the LTERM results in this error.
4E	LTERM ALREADY EXISTS	A local LTERM already exists by that name.
5B	MSNAME ALREADY EXISTS	An MSNAME already exists by that name.
5F	INVALID CHARACTERS IN NAME	Name is invalid because it contains an invalid character.
61	DFSBCB STORAGE ERROR	Request to get BCB storage for RCNT control block failed.
7F	TRANSACTION ALREADY EXISTS	A transaction already exists by that name.
8A	WILDCARD PARAMETER NOT SUPPORTED	The CREATE command does not support wildcard parameters. You must explicitly specify the names of the resources you want to create.
90	INTERNAL ERROR	An IMS system service failure.
93	RESERVED NAME	Name is reserved. For example, name starts with DFS (except for DFSSAMxx or DFSIVPxx), or is an IMS reserved name such as BASICEDT or WTOR.
212	HASH TABLE ADD RCNT FAILED	Request to add the RCNT to the hash table failed.

## Example of the CREATE LTERM command

In the following example, the CREATE LTERM command is used to create several MSC remote logical terminals for the MSC logical link path (MSNAME) named LINK31V.

TSO SPOC input:

```
CREATE LTERM NAME(H0G50,H0G51,H0G52) SET(MSNAME(LINK31V))
```

TSO SPOC output:

Lterm	MbrName	CC
H0G50	IMS1	0
H0G51	IMS1	0
H0G52	IMS1	0

OM API input:

```
CMD(CRE LTERM NAME(H0G50,H0G51,H0G52) SET(MSNAME(LINK31V)))
```

OM API output:

```
<imsout>
<ctl>
<omname>OM10M </omname>
<omvsn>1.7.0</omvsn>
<xmlvsn>20 </xmlvsn>
<statime>2014.268 15:37:32.492517</statime>
<stotime>2014.268 15:37:32.493343</stotime>
<staseq>CDCF5DCF24AE5483</staseq>
<stoseq>CDCF5DCF24E1F007</stoseq>
<rqsttkn1>USRT011 10083732</rqsttkn1>
<rc>00000000</rc>
<rsn>00000000</rsn>
</ctl>
<cmd>
<master>IMS1 </master>
<userid>USRT011 </userid>
<verb>CRE </verb>
<kwd>LTERM </kwd>
<input>CRE LTERM NAME(H0G50,H0G51,H0G52) SET(MSNAME(LINK31V)) </input>
</cmd>
<cmdrsphdr>
<hdr s1b1="LTERM" l1b1="Lterm" scope="LCL" key="1" len="8" dtype="CHAR"
  align="left" />
<hdr s1b1="MBR" l1b1="MbrName" scope="LCL" sort="a" key="2" scroll="no"
  len="8" dtype="CHAR" align="left" />
<hdr s1b1="CC" l1b1="CC" scope="LCL" sort="n" key="0" scroll="yes"
  len="4" dtype="INT" align="right" skipb="no" />
<hdr s1b1="CCTXT" l1b1="CCText" scope="LCL" sort="n" key="0"
  scroll="yes" len="*" dtype="CHAR" skipb="yes" align="left" />
</cmdrsphdr>
<cmdrspdata>
<rsp>LTERM(H0G50 ) MBR(IMS1 ) CC( 0) </rsp>
<rsp>LTERM(H0G51 ) MBR(IMS1 ) CC( 0) </rsp>
<rsp>LTERM(H0G52 ) MBR(IMS1 ) CC( 0) </rsp>
</cmdrspdata>
</imsout>
```

---

## CREATE MSLINK command

Use the CREATE MSLINK command to dynamically create a Multiple Systems Coupling (MSC) logical link between two MSC-enabled IMS systems.

You can assign a logical link to a physical link when you define it or later by using either the IMS type-2 UPDATE MSLINK command or the type-1 /MSASSIGN LINK command.

You can define a maximum of 1018 logical links in an IMS system.

Subsections:

- “Environment”
- “Syntax diagrams”
- “Keywords” on page 224
- “Usage notes” on page 227
- “Output fields” on page 227
- “Return and reason codes” on page 228
- “Completion codes” on page 229
- “Example of the CREATE MSLINK command” on page 230

## Environment

The following table lists the environments (DB/BC, DBCTL, and DCCTL) from which the command keywords can be issued.

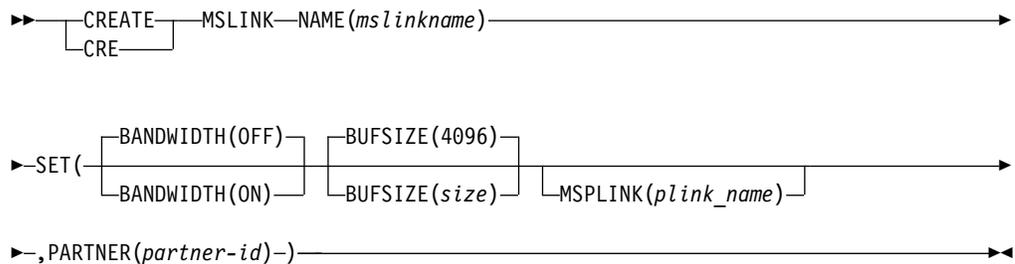
Table 136. Valid environments for the CREATE MSLINK command and keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
CREATE MSLINK NAME () SET()	X		X

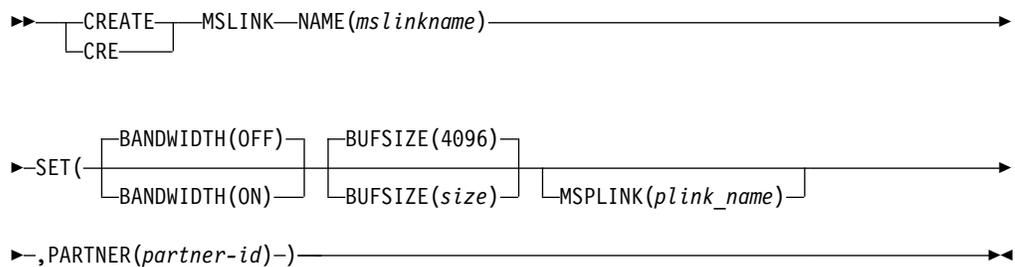
## Syntax diagrams

The syntax diagram for the CREATE MSLINK command is shown as several different syntax diagrams based on the type of physical link the logical link will be assigned to.

### CTC: Syntax diagram for logical links that are assigned to CTC physical links



### MTM: Syntax diagram for logical links that are assigned to MTM physical links





- The type-2 command UPDATE MSLINK NAME(*linkname*) SET(ASR(OFF | ON))

### BACKUP

For XRF-capable IMS systems, controls the automatic restart of TCP/IP and VTAM links after an XRF takeover.

The specification of BACKUP in the definition of a logical link overrides any switching options that are specified in the definition of the associated physical link. Use only if HSB=YES is specified on the IMSCTRL macro.

The value of *n* is a numeric integer from 1 to 7, inclusive, that sets the priority for reestablishing the session. When either the BACKUP keyword is omitted or no value is specified for BACKUP in the definition of either the physical or logical link, the default is 4. NO suppresses the session recovery of the physical link at takeover.

BACKUP is an MSC physical link attribute that you can set by using the CREATE MSPLINK, CREATE MSLINK, or UPDATE MSPLINK command. You can display the physical link BACKUP value by using the QUERY MSPLINK and QUERY MSLINK commands.

**Important:** Even though BACKUP is a physical link attribute, the CREATE MSLINK NAME(*linkname*) SET(MSPLINK(*msplinkname*)) command can also change the BACKUP value of the specified physical link. You can use the QUERY MSPLINK command to display the updated BACKUP value. You can also use the QUERY MSLINK command and specify the MSC logical links, including the newly created logical link, that are assigned to the physical link to display the updated BACKUP value.

Although BACKUP prioritizes the order in which IMS restarts TCP/IP and VTAM links, the active requests might be completed in any order because of variables such as internal VTAM conflicts and pacing, TCP/IP network traffic, and so on.

### BANDWIDTH(OFF | ON)

Specifies whether the logical link uses bandwidth mode.

#### OFF

Specifies that the logical link does not use bandwidth mode. However, if the physical link type specified by MSPLINK parameter is TCPIP, the bandwidth mode is set on. OFF is the default.

**ON** Specifies that the logical link uses bandwidth mode.

### BUFSIZE

Specifies the input and output buffer sizes for the newly created logical link.

The valid range of buffer sizes for all link types is 1024 - 65536 bytes.

The same buffer size must be specified by the IMS systems at each end of a physical link.

If BUFSIZE is specified, the created logical link sets its buffer size to the value that is passed by BUFSIZE parameter.

If BUFSIZE is not specified but MSPLINK is specified, the buffer size of the created logical link inherits the buffer size from its physical link.

If both BUFSIZE and MSPLINK parameters are not specified, the created logical link takes the default value 4096 for its buffer size.

## MODETBL

Specifies the name of the VTAM logon mode table entry (logon mode name) containing the SNA bind parameters to be used when a session is established for this terminal.

This function allows a specification for referencing an entry other than the default entry in the user's VTAM logon mode table.

With this function, if MODETBL is not specified, no functional or operational change affects the user.

If MODETBL is specified, the specified entry name is used.

You can display the current MODETBL name by using either the /DISPLAY LINK command or the QUERY MSLINK command. You can override the MODETBL name by using any of the following commands:

- The VARY ACT, LOGON= command by the network terminal operator
- The /RST LINK or the /CHANGE LINK command by the master terminal operator
- The type-2 command UPDATE MSLINK NAME(*linkname*) SET(MODETBL(*modetablename*)) through the OM API

If the VTAM-node=name being defined is in another domain (that is, a cross-domain resource), the MODETBL parameter need not be specified.

**Related reading:** See *z/OS Communications Server: SNA Programming* for additional information.

## MSPLINK

Assigns the logical link to the specified physical link.

If MSPLINK is specified, the physical link must exist in the system. When the physical link is either a TCP/IP physical link or a VTAM physical link, it must be stopped. When the physical link is either a CTC physical link or a MTM physical link, it must not be assigned to a logical link.

If you do not specify this keyword, the logical link is not assigned to a physical link.

You can assign this logical link to a physical link later by using either the /MSASSIGN LINK command or the type-2 command UPDATE MSLINK NAME(*linkname*) SET(MSPLINK(*msplinkname*)).

You must assign the logical link to a physical link before communication can be established between the two systems.

## PARTNER

A 2-character alphanumeric identification that associates this logical link with its associated logical definition in the partner IMS system. The partner ID ensures that the two related logical link definitions in the partner IMS systems are always logically and physically connected. Both systems must have logical links that are defined with the same partner ID. If the logical link is assigned to a different physical link, the two systems can still communicate through the logical link, which remains as defined.

You can modify the partner ID by using the type-2 command UPDATE MSLINK NAME(*linkname*) SET(PARTNER(*partner-id*)).

## SYNCOPT(SYNCSSESS | FORCSSESS)

For logical links that are assigned to TCP/IP or VTAM physical links only, specifies the session resynchronization options for the logical link.

### **SYNCSESS**

Indicates that session initiation is to be completed only if session resynchronization is successful. Successful session resynchronization occurs when the message sequence numbers of the two logical units in session agree, or when the sequence number of the sender is not less than the sequence number of the receiver. SYNCSESS is the default.

### **FORCSESS**

Forces the session initiation to be completed regardless of whether session resynchronization is successful.

The SYNCOPT setting can be overridden by either the type-1 command /CHANGE LINK *nn* FORCSESS | SYNCSESS | COLDSESS or the type-2 command UPDATE MSLINK NAME(*linkname*) SET(SYNOPT(FORCSESS | SYNCSESS | COLDSESS)).

## **Usage notes**

The CREATE MSLINK command can be issued only through the Operations Manager API. These commands apply to DB/DC and DCCTL systems. The syntax for this command is defined in XML and is available to automation programs that communicate with OM.

MSC logical link resources that are added by the CREATE MSLINK command are logged to save them across warm and emergency restarts. If you dynamically created MSC logical link resources with the CREATE MSLINK command and want to restore the definitions at the next IMS cold start, you must also perform one of the following actions:

- Export the resources to the IMSRSC repository by using the automatic export function and import the resources at IMS cold start by using the automatic import function.
- Define the resources with stage-1 system definition macros and regenerate the MSC resources into the DFSCLL3x member in the IMS.SDFSRESL data set.

The CREATE MSLINK NAME(*mslinkname*) command specified with SET(MSPLINK(*msplinkname*)) is rejected if at least one logical link attribute specified in the command (ASR, BACKUP, MODETBL, or SYNOPT) conflicts with the physical link type. However, after the logical link is created, it can be reassigned to any type of physical link with an UPDATE MSLINK command or an /MSASSIGN command. If a logical link defined with attributes applicable to specific physical link types is reassigned to a different physical link type that does not support those attributes, they are ignored when the link is started. The attributes are still defined for the logical link and can be displayed.

## **Output fields**

The following table shows the CREATE MSLINK output fields. The columns in the table are as follows:

### **Short label**

Contains the short label that is generated in the XML output.

### **Keyword**

Identifies keyword on the command that caused the field to be generated. N/A appears for output fields that are always returned. The value *error* appears for output fields that are returned only in case of an error.

## Meaning

Provides a brief description of the output field.

*Table 137. Output fields for the CREATE MSLINK command*

Short label	Long label	Keyword	Meaning
CC	CC	N/A	Completion code
CCTXT	CCText	N/A	Completion code text that briefly explains the meaning of the non-zero completion code
MBR	MbrName	N/A	IMSplex member that built the output line
MSCL	MSLink	N/A	Logical link name

## Return and reason codes

The OM return and reason codes that might be returned as a result of the CREATE MSLINK command are standard for all commands that are entered through the OM.

The following table includes the return and reason codes and a brief explanation of the codes. The return or reason code returned for the command might also indicate an error from a CSL request.

*Table 138. Return and reason codes for the CREATE MSLINK command*

Return code	Reason code	Meaning
X'00000000'	X'00000000'	The command completed successfully. The command output contains a line for each resource, which is accompanied by its completion code. See the completion code table for details.
X'00000008'	X'00002157'	Invalid BACKUP parameter
X'00000008'	X'00002158'	Invalid BUFSIZE value.
X'00000008'	X'0000215F'	Invalid MODETBL name.
X'00000008'	X'0000216E'	TCP/IP requires BANDWIDTH(ON).
X'00000008'	X'0000216F'	PARTNER ID is not specified, but is required.
X'00000008'	X'00002170'	The MSPLINK has an active MSLINK.
X'00000008'	X'00002171'	The specified MSPLINK is a TCP/IP or VTAM link that is not stopped.
X'00000008'	X'00002172'	MSPLINK is already assigned to a logical link.
X'00000008'	X'00002173'	PARTNER ID already exists.
X'00000008'	X'00002174'	PARTNER ID is invalid.
X'00000008'	X'00002175'	SYNCOPT parameter conflicts with the MSPLINK type.
X'0000000C'	X'00003000'	Command was successful for some resources but failed for others. The command output contains a line for each resource, which is accompanied by its completion code. See the completion code table for details.

Table 138. Return and reason codes for the CREATE MSLINK command (continued)

Return code	Reason code	Meaning
X'0000000C'	X'00003004'	Command was not successful for any of the resources. The command output contains a line for each resource, which is accompanied by its completion code. See the completion code table for details.
X'00000010'	X'0000400C'	Command is not valid on the XRF alternate.
X'00000010'	X'00004014'	Command is not valid on the RSR tracker.
X'00000010'	X'0000402C'	Command is not valid because MSC is not enabled in the IMS system.
X'00000010'	X'0000402D'	Command is not allowed because dynamic definition is not enabled for MSC. MSCRSCS=DYN is not specified in the MSC section of the DFSDFxxx member of the IMS PROCLIB data set.
X'00000010'	X'0000402E'	Command is not valid because the system is not an XRF active system.
X'00000014'	X'00005004'	A DFSOCMD response buffer could not be obtained.
X'00000014'	X'00005008'	DFSPOOL storage could not be obtained.
X'00000014'	X'00005010'	Unable to obtain latch.

## Completion codes

The following table includes an explanation of the completion codes.

Table 139. Completion codes for the CREATE MSLINK command

Completion code	Completion code text	Meaning
0		The command completed successfully for the logical link.
11	RESOURCE ALREADY EXISTS	A logical link with the same name already exists.
17	ANOTHER CMD IN PROGRESS	Another command (such as DELETE MSLINK or UPDATE MSLINK) is in progress for the logical link.
5F	INVALID CHARACTERS IN NAME	Name is invalid because it contains an invalid character.
61	DFSBCB STORAGE ERROR	Request to get BCB storage for LLB control block failed.
62	HIOP STORAGE ERROR	Request to get HIOP storage failed.
8A	WILDCARD PARAMETER NOT SUPPORTED	The CREATE command does not support wildcard parameters. You must explicitly specify the names of the resources you want to create.
8D	RESOURCE IS NOT STOPPED	The MSC logical link resource is not stopped.
90	INTERNAL ERROR	An IMS system service failure.

Table 139. Completion codes for the CREATE MSLINK command (continued)

Completion code	Completion code text	Meaning
93	RESERVED NAME	Name is reserved. For example, name starts with DFS (except for DFSSAMxx or DFSIVPxx), or is an IMS reserved name, such as BASICEDT or WTOR.
206	CREATED LINK ENQUEUE FAILED	The enqueue of the logical link LLB control block onto one of the LLB chains failed.
208	MAX LINK NUMBER REACHED	The maximum number of MSC logical links has been reached.
20E	NAME CANNOT BEGIN WITH A NUMBER	The MSLINK name must begin with an alphabetic character. It cannot begin with a number.
215	ENQUEUE INTERNAL LNB FAILED	Hash internal LNB failed.
216	CREATE MSLINK ITASK FAILED	Create MSLINK itask failed.
217	BUILD INTERNAL MSNAME FAILED	The build of the internal MSNAME - MSN $_{mm}$ for the newly created logical link failed.

## Example of the CREATE MSLINK command

In the following example, one MSC VTAM logical link is dynamically created. Only one logical link can be created at a time because the partner ID must be unique.

TSO SPOC input:

```
CREATE MSLINK NAME(LNKV0001) SET(MSPLINK(PLNK12V),PARTNER(ZA))
```

TSO SPOC output:

```
MSLink  MbrName  CC
LNKV0001 IMS1    0
```

OM API input:

```
CMD(CRE MSLINK NAME(LNKV0001) SET(MSPLINK(PLNK12V),PARTNER(ZA)))
```

OM API output:

```
<imsout>
<ctl>
<omname>OM10M  </omname>
<omvsn>1.7.0</omvsn>
<xm1vsn>20  </xm1vsn>
<statime>2015.058 22:58:56.342531</statime>
<stotime>2015.058 22:58:56.343694</stotime>
<staseq>CE92A18E8E6039E3</staseq>
<stoseq>CE92A18E8EA8EE8A</stoseq>
<rqsttkn1>USRT011 10145856</rqsttkn1>
<rc>00000000</rc>
<rsn>00000000</rsn>
</ctl>
<cmd>
<master>IMS1  </master>
```

```

<userid>USRT011 </userid>
<verb>CRE </verb>
<kwd>MSLINK </kwd>
<input>CREATE MSLINK NAME(LNKV0001) SET(MSPLINK(PLNK12V),PARTNER(ZA))
</input>
</cmd>
<cmdsphdr>
<hdr s1b1="MSCL" l1b1="MSLink" scope="LCL" sort="a" key="1" scroll="
no" len="8" dtype="CHAR" align="left" />
<hdr s1b1="MBR" l1b1="MbrName" scope="LCL" sort="a" key="2"
scroll="no" len="8" dtype="CHAR" align="left" />
<hdr s1b1="CC" l1b1="CC" scope="LCL" sort="n" key="0"
scroll="yes" len="4" dtype="INT" align="right" skipb="no" />
<hdr s1b1="CCTXT" l1b1="CCText" scope="LCL" sort="n" key="0"
scroll="yes" len="*" dtype="CHAR" skipb="yes" align="left" />
</cmdsphdr>
<cmdspdata>
MSCL(LNKV0001) MBR(IMS1 ) CC( 0)
</cmdspdata>
</imsout>

```

---

## CREATE MSNAME command

Use the CREATE MSNAME command to define a unique Multiple Systems Coupling (MSC) logical link path between a local IMS system and a remote IMS system.

In a logical link path, the local IMS system is identified by local system identifier (SIDL) and the remote IMS system is identified by the remote system identifier (SIDR).

In an IMS system, each logical link path is defined by a unique SIDL and SIDR pair and a one-to-one relationship exists between a logical link path and a SIDR. When defining a remote application program to a local IMS system, the application and transaction definitions must refer to a remote SYSID of a logical link path that is defined in the same IMS system. You can define an application program as remote either dynamically with the command CREATE TRAN SET(PGM(*pgmname*),SIDL(*sidl*),SIDR(*sidr*)) or statically with the SYSID parameter on either the APPLCTN or TRANSACT system definition macro.

Subsections:

- “Environment”
- “Syntax diagram” on page 232
- “Keywords” on page 232
- “Usage notes” on page 232
- “Output fields” on page 233
- “Return and reason codes” on page 234
- “Completion codes” on page 235
- “Example of creating a logical link path” on page 235

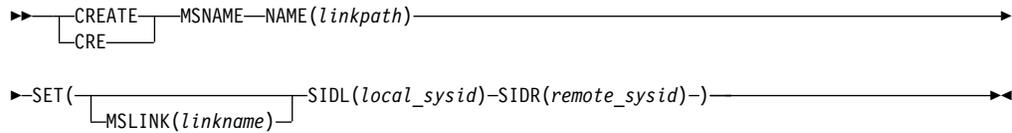
### Environment

The following table lists the environments (DB/BC, DBCTL, and DCCTL) from which the command keywords can be issued.

Table 140. Valid environments for the CREATE MSNAME command and keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
CREATE MSNAME	X		X

## Syntax diagram



## Keywords

The following keywords are valid for the CREATE MSNAME command:

### NAME()

The 1- to 8-character name of the MSC logical link path to be created.

### SET()

Specifies the attributes of the logical link path.

### MSLINK()

The 1- to 8-character name of the MSC logical link that you want to assign this logical link path to. The logical link name specified must already exist.

Before the logical link path can be assigned to a logical link, the specified logical link must be stopped. If the logical link is not stopped, the command terminates without creating the logical link path.

### SIDL()

A numeric value from 1 - 2036 that defines the local IMS system identifier of the logical link path. The local system ID is used for routing messages back to this system.

The value specified on the SIDL keyword cannot be the same as the remote system ID (SIDR) value of any logical link path that is defined to the local IMS system.

### SIDR()

A numeric value from 1 - 2036 that defines the remote IMS system identifier of the logical link path.

The value specified on the SIDR keyword must be unique among the SIDL and SIDR values of all of the logical link paths that are defined to the local IMS system.

Also, the SIDR value cannot be the same as the local system ID (SIDL) value specified on any APPLCTN macro or for any transaction that is defined to the local IMS system.

## Usage notes

The CREATE MSNAME command can be issued only through the Operations Manager API. These commands apply to DB/DC and DCCTL systems. The syntax for this command is defined in XML and is available to automation programs that communicate with OM.

To determine if a proposed NAME, SIDR, or SIDL value is already in use by another logical link path that is defined to the local IMS system, you can issue the QUERY MSNAME NAME(\*) SHOW(SYSID) type-2 command.

To determine if a proposed SIDR value is already in use as a SIDL by a transaction defined to the local IMS system, you can issue the QUERY TRAN NAME(\*) SHOW(MSNAME) type-2 command.

After a logical link path is defined, you can change its remote or local system ID by using the type-2 command UPDATE MSNAME NAME(*msname*) SET(SIDR(*remote\_sysid*), SIDL(*local\_SID*)).

In a shared-queues environment, the CREATE MSNAME command creates the MSC MSNAME (logical link path) locally in the IMS that has a physical link outside the IMSplex, and causes a dynamic MSNAME to be created in all the other IMSs in the shared-queues group.

MSC logical link path resources that are added by the CREATE MSNAME command are logged to save them across warm and emergency restarts. If you dynamically created MSC logical link path resources with the CREATE MSNAME command and want to restore the definitions at the next IMS cold start, you must also perform one of the following actions:

- Export the resources to the IMSRSC repository by using the automatic export function and import the resources at IMS cold start by using the automatic import function.
- Define the resources with stage-1 system definition macros and regenerate the MSC resources into the DFSCLC0x member in the IMS.SDFSRESL data set.

## Output fields

The following table shows the CREATE MSNAME output fields. The columns in the table are as follows:

### Short label

Contains the short label that is generated in the XML output.

### Keyword

Identifies keyword on the command that caused the field to be generated. N/A appears for output fields that are always returned. The value *error* appears for output fields that are returned only in case of an error.

### Meaning

Provides a brief description of the output field.

Table 141. Output fields for the CREATE MSNAME command

Short label	Long label	Keyword	Meaning
CC	CC	N/A	Completion code.
CCTXT	CCText	N/A	Completion code text that briefly explains the meaning of the non-zero completion code.
MBR	MbrName	N/A	IMSplex member that built the output line.
MSN	MSName	N/A	The name of the logical link path.

## Return and reason codes

The OM return and reason codes that might be returned as a result of the CREATE MSNAME command are standard for all commands that are entered through the OM.

The following table includes the return and reason codes and a brief explanation of the codes. The return or reason code returned for the command might also indicate an error from a CSL request.

*Table 142. Return and reason codes for the CREATE MSNAME command*

Return code	Reason code	Meaning
X'00000000'	X'00000000'	The command completed successfully. The command output contains a line for each resource, which is accompanied by its completion code. See the completion code table for details.
X'00000008'	X'00002127'	SIDL/SIDR must be specified as a pair. Either SIDL was specified alone, or SIDR was specified alone.
X'00000008'	X'00002136'	The name specified in MSLINK() is not a valid logical link name.
X'0000000C'	X'00003000'	Command was successful for some resources but failed for others. The command output contains a line for each resource, which is accompanied by its completion code. See the completion code table for details.
X'0000000C'	X'00003004'	Command was not successful for any of the resources. The command output contains a line for each resource, which is accompanied by its completion code. See the completion code table for details.
X'00000010'	X'00004000'	Command is not valid during IMS restart.
X'00000010'	X'0000400C'	Command is not valid on the XRF alternate.
X'00000010'	X'00004014'	Command is not valid on the RSR tracker.
X'00000010'	X'0000402C'	Command is not valid because MSC is not enabled in the IMS system.
X'00000010'	X'0000402D'	Command is not allowed because dynamic definition is not enabled for MSC. MSCRSCS=DYN is not specified in the MSC section of the DFSDFxxx member of the IMS PROCLIB data set.
X'00000014'	X'00005004'	Command processing terminated because a DFSOCMD response buffer could not be obtained.
X'00000014'	X'00005008'	DFSPOOL storage could not be obtained.
X'00000014'	X'0000500C'	AWE get error.
X'00000014'	X'00005010'	Unable to obtain latch.
X'00000014'	X'00005058'	Synchronizing with DFSMNZ00 error.
X'00000014'	X'00005FFF'	IMS internal error.

## Completion codes

The following table includes an explanation of the completion codes.

Table 143. Completion codes for the CREATE MSNAME command

Completion code	Completion code text	Meaning
0		The command completed successfully for the physical link.
11	RESOURCE ALREADY EXISTS	A logical link path with the same name already exists.
45	INVALID SIDR VALUE	The value specified in SIDR() is invalid. The value must be currently undefined in this system.
4E	LTERM ALREADY EXISTS	An LTERM already exists by that name.
5F	INVALID CHARACTERS IN NAME	Name is invalid because it contains an invalid character.
7F	TRANSACTION ALREADY EXISTS	Transaction already exists by that name.
83	INVALID SIDL VALUE	The value specified in SIDL() is invalid. The value must be currently undefined or local in this system.
8A	WILDCARD PARAMETER NOT SUPPORTED	The CREATE command does not support wildcard parameters. You must explicitly specify the names of the resources you want to create.
90	INTERNAL ERROR	An IMS system service failure.
93	RESERVED NAME	Name is reserved. For example, name starts with DFS (except for DFSSAMxx or DFSIVPxx), or is an IMS reserved name such as BASICEDT or WTOR.
102	MSLINK NOT STOPPED	The create cannot complete because the link specified in MSLINK() is not stopped.
10D	INVALID SIDL, LESS THAN LOW SIDL	The value specified in SIDL() is invalid. The value cannot be lower than the current lowest local SID in this system.
20C	MSNAME ITASK CREATE FAILED	IMS encountered an internal error and failed to create an LNB for MSNAME.

## Example of creating a logical link path

In the following example, the CREATE MSNAME command creates a logical link path named MSNV001. The remote system identifier is 55 and the local system identifier is 755.

TSO SPOC input:

```
CRE MSNAME NAME(MSNV001) SET(SIDR(55),SIDL(755))
```

TSO SPOC output:

```
MSName MbrName CC  
MSNV001 IMS1 0
```

OM API input:

```
CMD(CRE MSNAME NAME(MSNV001) SET(SIDR(55),SIDL(755))
```

OM API output:

```
<imsout>  
<ctl>  
<omname>OM10M </omname>  
<omvsn>1.7.0</omvsn>  
<xm1vsn>20 </xm1vsn>  
<statime>2015.061 23:55:44.495618</statime>  
<stotime>2015.061 23:55:44.496502</stotime>  
<staseq>CE9673D93500258C</staseq>  
<stoseq>CE9673D935376B16</stoseq>  
<rqsttkn1>USRT011 10155544</rqsttkn1>  
<rc>00000000</rc>  
<rsn>00000000</rsn>  
</ctl>  
<cmd>  
<master>IMS1 </master>  
<userid>USRT011 </userid>  
<verb>CRE </verb>  
<kwd>MSNAME </kwd>  
<input>CREATE MSNAME NAME(MSNV001)  
SET(MSLINK(LNK12V02),SIDR(55),SIDL(705)) </input>  
</cmd>  
<cmdrsphdr>  
<hdr s1b1="MSN" l1b1="MSName" scope="LCL" sort="a" key="1" scroll="no"  
len="8" dtype="CHAR" align="left" skipb="no" />  
<hdr s1b1="MBR" l1b1="MbrName" scope="LCL" sort="a" key="3" scroll="no"  
len="8" dtype="CHAR" align="left" skipb="no" />  
<hdr s1b1="CC" l1b1="CC" scope="LCL" sort="n" key="0" scroll="yes"  
len="4" dtype="INT" align="right" skipb="no" />  
</cmdrsphdr>  
<cmdrspdata>  
<rsp>MSN(MSNV001 ) MBR(IMS1) CC( 0) </rsp>  
</cmdrspdata>  
</imsout>
```

---

## CREATE MSPLINK command

Use the CREATE MSPLINK command to dynamically create a Multiple Systems Coupling (MSC) physical link between two MSC-enabled IMS systems.

Using the CREATE MSPLINK command, you can create any of the following types of MSC physical links:

- Channel to channel (CTC)
- Main storage to main storage (MTM)
- Transmission Control Protocol/Internet Protocol (TCP/IP)
- Virtual Telecommunications Access Method (VTAM)

You can define a maximum of 1018 physical links in an IMS system.

Subsections:

- “Environment” on page 237
- “Syntax diagrams” on page 237

- “Keywords” on page 238
- “Usage notes” on page 242
- “Output fields” on page 243
- “Return and reason codes” on page 243
- “Completion codes” on page 245
- “Examples of the CREATE MSPLINK command” on page 246

## Environment

The following table lists the environments (DB/BC, DBCTL, and DCCTL) from which the CREATE MSPLINK command and its keywords can be issued.

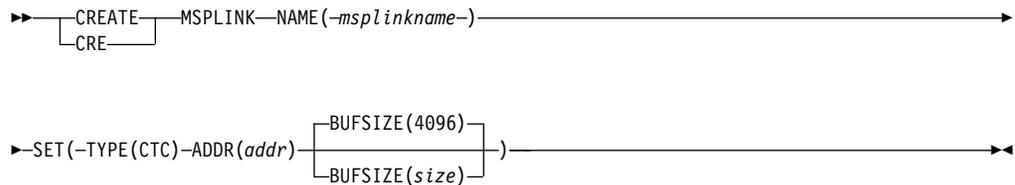
Table 144. Valid environments for the CREATE MSPLINK command and keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
CREATE MSPLINK NAME () SET()	X		X

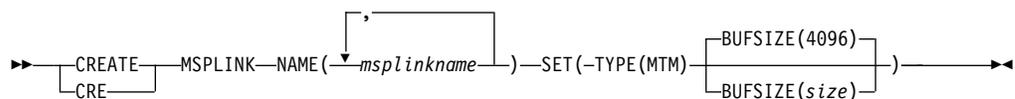
## Syntax diagrams

The syntax diagrams for each type of MSC physical link are shown separately.

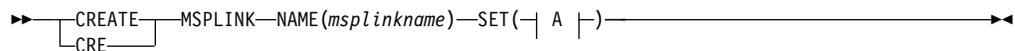
### Syntax diagram for CTC physical links



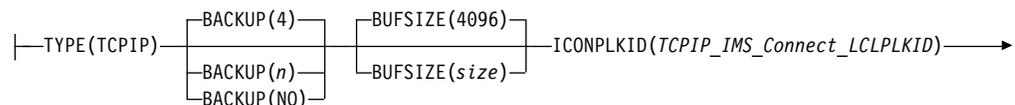
### Syntax diagram for MTM physical links

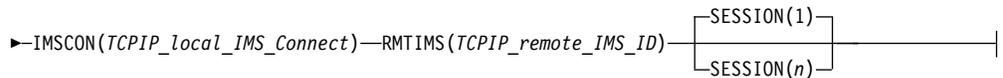


### Syntax diagram for TCPIP physical links

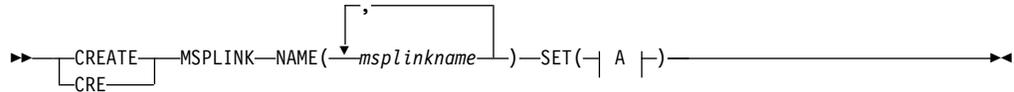


**A:**

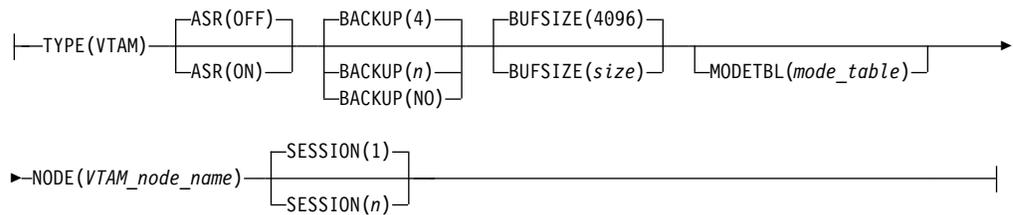




## Syntax diagram for VTAM physical links



**A:**



## Keywords

The following keywords are valid for the CREATE MSPLINK command:

### NAME

Specifies the 1- to 8-character name of the MSC physical link that is to be created.

For MTM and VTAM links, you can specify multiple names on the NAME keyword. When multiple names are specified, multiple physical links are created, one for each name specified. Other than the name, all of the physical links that are created this way will have the same attributes.

For CTC and TCP/IP links, only a single 1- to 8-character name can be specified on the NAME keyword, because only one CTC or TCP/IP physical link can be created at a time by the CREATE MSPLINK command. Certain attributes of CTC and TCP/IP physical links are unique to each physical link.

- For CTC links, the ADDR attribute must be unique among all CTC physical links.
- For TCP/IP physical links the ICONPLKID and IMSCON attribute pair must be unique among all TCP/IP physical links.
- Consequently, a set of CTC or TCP/IP physical links cannot be created with identical attributes.

Names must be alphanumeric (A - Z, 0 - 9, #, \$, and @). All MSPLINK names must begin with an alphabetic character (A - Z, #, \$, and @) followed by 0 - 7 alphanumeric characters.

Names cannot include a blank, comma, period, hyphen, or equal sign.

Names cannot begin with DFS, with the exception of DFSIVPxx and DFSSAMxx.

The following reserved words cannot be the name of a physical link:

- AFTER
- BASICEDT

- DBCDM
- DBRC
- FOR
- ISCEDT
- MSDB
- ON
- SDB
- SECURITY
- TO
- WTOR

The physical link name is used to assign a logical link to it. The physical link name can be changed by using the UPDATE MSPLINK command after it is created or defined in the system.

### SET()

Sets various attributes of the physical link. The required and optional attributes differ depending on the type of physical link that you are defining.

#### ADDR()

Specifies the address of the channel-to-channel adapter. The address value can be three or four hexadecimal digits. This operand is required for CTC link types and is invalid for other link types.

All devices that are attached to the same channel as the CTC adapter must be accessible through an alternative channel. In addition, do not attach system resources (for example, paging devices) to the same channel as the CTC adapter.

The ADDR value must be unique among all CTC physical links defined in the IMS system.

You can change the value specified on the ADDR keyword by using the type-2 UPDATE MSPLINK SET(ADDR(*addr*)) command.

#### ASR()

Specifies the Automatic Session Restart option. This optional parameter applies only to VTAM links.

##### OFF

Sets Automatic Session Restart off. The default is OFF.

**ON** For VTAM links that are supported by the VTAM session outage notification (SON) facility, sets Automatic Session Restart on.

For all logical links that are assigned to the physical link, the ASR specification on a physical link definition serves as the default ASR specification. The ASR specification on the physical link can be overridden in the definition of a logical link.

To display the current ASR option, you can use the type-2 command QUERY MSLINK. To change the current ASR option, you can use either of the following commands:

- The type-1 command /CHANGE
- The type-2 command UPDATE MSPLINK NAME(*msplinkname*) SET(ASR(OFF|ON))

ASR works only if both sides of the link are using the ASR option.

## BACKUP()

For XRF-capable IMS systems, controls the automatic restart of TCP/IP and VTAM links after an XRF takeover.

**n** A numeric integer from 1 to 7 that sets the priority for reestablishing the session. The default is 4.

**NO** Specifying BACKUP=NO suppresses the automatic restart of the MSC physical link.

Although BACKUP prioritizes the order in which IMS restarts TCP/IP and VTAM links, the active requests might be completed in any order because of variables such as internal VTAM conflicts and pacing, and Internet Protocol network traffic.

You can change the value specified on the BACKUP keyword by using the type-2 UPDATE MSPLINK SET(BACKUP(*n* | NO)) command.

BACKUP is an MSC physical link attribute that you can set by using the CREATE MSPLINK, CREATE MSLINK, or UPDATE MSPLINK command.

You can display the physical link BACKUP value by using the QUERY MSPLINK and QUERY MSLINK commands.

**Important:** Even though BACKUP is a physical link attribute, the CREATE MSLINK NAME(*linkname*) SET(MSPLINK(*msplinkname*)) command can also change the BACKUP value of the specified physical link. You can use the QUERY MSPLINK command to display the updated BACKUP value. You can also use the QUERY MSLINK command and specify the MSC logical links, including the newly created logical link, that are assigned to the physical link to display the updated BACKUP value.

## BUFSIZE()

Specifies the input and output buffer sizes for each logical link that is assigned to this physical link.

The valid range of buffer sizes for all link types is 1024 - 65536 bytes.

The same buffer size must be specified by the IMS systems at each end of a physical link.

IMS initializes the MSC link buffers to the size specified on the BUFSIZE parameter. If BUFSIZE is not specified, IMS uses a buffer size of 4096.

If you are using bandwidth mode, a BUFSIZE value of 1024 is too small to send multiple messages with one buffer. Specify a value of at least 4096.

You can change the size of a buffer of an individual logical link by using the UPDATE MSLINK NAME(*linkname*) SET(BUFSIZE(*new\_bufsize*)) or the UPDATE MSPLINK NAME(*linkname*) SET(BUFSIZE(*new\_bufsize*)) type-2 command. The BUFSIZE value for a physical link applies to logical links that are assigned to the physical link. The physical link does not have a buffer size. Specifically, the value is used to set the buffer size for logical links when they are later assigned to the physical link, but only if the logical links do not already have a buffer size.

## ICONPLKID()

For TCP/IP links, specifies the name that is used by IMS Connect to identify this physical link. The name that is specified on the ICONPLKID keyword of the CREATE MSPLINK command must match the name that is specified on the LCLPLKID parameter of the MSC statement that defines the physical link to the local IMS Connect instance that is managing

TCP/IP communications for this link. The name that is specified on the ICONPLKID must start with an alphabetic character and can be 1 - 8 alphanumeric characters long.

ICONPLKID is a required keyword for TCP/IP physical links and it is invalid for other link types.

The ICONPLKID and IMSCON values must define a unique attribute pair among all TCP/IP physical links.

#### **IMSCON()**

For TCP/IP links, specifies the IMSplex name of the local IMS Connect instance that manages TCP/IP communications for this physical link. The name that is specified on the IMSCON keyword must match the name that is specified on the MEMBER parameter of either the MSC statement or the IMSPLEX statement in the IMS Connect configuration PROCLIB member (HWSCFGxx). IMSplex names start with an alphabetic character and are 1- to 8-alphanumeric characters long.

The IMSCON keyword is required for TCP/IP physical links and is invalid for other link types.

You can use UPDATE MSPLINK command with the IMSCON keyword to switch the IMS Connect instance that an MSC physical link uses.

The ICONPLKID and IMSCON values must define a unique attribute pair among all TCP/IP physical links.

#### **MODETBL()**

Specifies the name of the VTAM logon mode table entry (logon mode name) containing the SNA bind parameters to be used when a session is established for this terminal.

With MODETBL(), you can have a specification for referencing an entry other than the default entry in the user VTAM logon mode table. Normally, the mode table entry name is specified when a terminal operator logs on to a terminal; this result was not possible, however, from the IMS master terminal because IMS initiated the session.

If MODETBL= is not specified, no functional or operational change affects the user.

If MODETBL= is specified, the specified entry name is used.

You can display the current MODETBL name for each assigned logical link by using either the /DISPLAY LINK command or the QUERY MSLINK command. You can override the MODETBL name by using:

- The LOGON APPLID entry by the remote terminal operator
- The VARY ACT, LOGON= command by the network terminal operator
- The /RST or the /CHANGE command by the master terminal operator
- Either the type-2 command UPDATE MSPLINK or UPDATE MSLINK through the Operations Manager API

The MODETBL= parameter is required for the IMS master terminal if the VTAM default mode table is not configured specifically for the device to be used as the IMS master terminal.

Specifying MODETBL on the CREATE MSPLINK command circumvents the requirement to specify the mode table entry when an operator logs on to a terminal that always requires the specification of the same mode table entry name.

If the terminal that is being defined is in another domain (that is, a cross-domain resource), the MODETBL() keyword does not need to be specified.

**Related reading:** For more information, see z/OS Communications Server: SNA Programming.

#### **NODE()**

Specifies a 1- to 8-alphanumeric name for the remote VTAM node (APPLID) name for the VTAM line. The NODE() keyword is required for VTAM physical links.

This node is the VTAM node name of the remote IMS system at the other end of the link. This node is the label on the VTAM APPL statement for the remote IMS system. For an MSC VTAM link that is communicating with an XRF complex, the node name must be the VTAM USERVAR that is associated with the partner XRF complex.

You can change the value that is specified on the NODE() keyword by using the type-2 IMS command UPDATE MSPLINK command.

#### **RMTIMS()**

Specifies the IMS ID of the remote IMS system that this physical link connects to. The remote IMS ID must be 1 to 8 alphanumeric characters.

RMTIMS is a required parameter for TCP/IP physical links and is invalid for other link types.

You can change the remote IMS system that an MSC physical link references by using the type-2 IMS command UPDATE MSPLINK command.

#### **SESSION()**

For TCP/IP and VTAM physical link types only, specifies the number of parallel sessions that can be active for the physical link. Valid values are 1 - 1018. The default is 1.

If you specify a larger number than you currently need, you can dynamically assign more logical links to the physical link later.

Because a logical parallel session uses the same amount of storage for control blocks and buffers as a physical link, allocating many parallel sessions can use an excessive amount of common storage area. Predetermine how much common storage area you are likely to need.

You can change the value specified on the SESSION keyword by using the type-2 UPDATE MSPLINK SET(SESSION(*n*)) command.

#### **TYPE()**

Specifies the type of physical link that is being defined. Valid parameters are CTC, MTM, TCPIP, and VTAM.

With the MTM type link, you can connect IMS systems that are running in the same z/OS system without a hardware link.

### **Usage notes**

The CREATE MSPLINK command can be issued only through the Operations Manager API. These commands apply to DB/DC and DCCTL systems. The syntax for this command is defined in XML and is available to automation programs that communicate with OM.

When creating CTC links with the CREATE MSPLINK command, you do not need to specify a DDNAME for the JCL statement that describes the physical CTC connection. IMS creates the DDNAME dynamically by using the following naming convention: DFSMnnnn, where *nnnn* is the link number.

MSC physical link resources that are added by the CREATE MSPLINK command are logged to save them across warm and emergency restarts. If you dynamically created MSC physical link resources with the CREATE MSPLINK command and want to restore the definitions at the next IMS cold start, you must also perform one of the following actions:

- Export the resources to the IMSRSC repository by using the automatic export function and import the resources at IMS cold start by using the automatic import function.
- Define the resources with stage-1 system definition macros and regenerate the MSC resources into the DFSCLL3x member in the IMS.SDFSRESL data set.

## Output fields

The following table shows the CREATE MSPLINK output fields. The columns in the table are as follows:

### Short label

Contains the short label that is generated in the XML output.

### Keyword

Identifies keyword on the command that caused the field to be generated. N/A appears for output fields that are always returned. The value *error* appears for output fields that are returned only in case of an error.

### Meaning

Provides a brief description of the output field.

*Table 145. Output fields for the CREATE MSPLINK command*

Short label	Long label	Keyword	Meaning
CC	CC	N/A	Completion code.
CCTXT	CCText	N/A	Completion code text that briefly explains the meaning of the non-zero completion code.
MBR	Mbrname	N/A	IMSplex member that built the output line.
MSP	MSPLink	N/A	Physical link name.

## Return and reason codes

The OM return and reason codes that might be returned as a result of the CREATE MSPLINK command are standard for all commands that are entered through the OM.

The following table includes the return and reason codes and a brief explanation of the codes. The return or reason code returned for the command might also indicate an error from a CSL request.

Table 146. Return and reason codes for the CREATE MSPLINK command

Return code	Reason code	Meaning
X'00000000'	X'00000000'	The command completed successfully. The command output contains a line for each resource, which is accompanied by its completion code. See the completion code table for details.
X'00000008'	X'00002152'	The ADDR keyword is required for TYPE(CTC).
X'00000008'	X'00002153'	The ADDR keyword is supported only when TYPE(CTC) is specified.
X'00000008'	X'00002154'	Invalid ADDR value.
X'00000008'	X'00002155'	The ASR keyword is not supported with the specified TYPE value.
X'00000008'	X'00002156'	The BACKUP keyword is supported only when TYPE(TCPIP) or TYPE(VTAM) is specified.
X'00000008'	X'00002157'	Invalid BACKUP value.
X'00000008'	X'00002158'	Invalid BUFSIZE value.
X'00000008'	X'00002159'	The ICONPLKID keyword is required when TYPE(TCPIP) is specified.
X'00000008'	X'0000215A'	Invalid ICONPLKID value.
X'00000008'	X'0000215B'	The ICONPLKID keyword is not supported with the specified TYPE value.
X'00000008'	X'0000215C'	The IMSCON keyword is required when TYPE(TCPIP) is specified.
X'00000008'	X'0000215D'	Invalid IMSCON value.
X'00000008'	X'0000215E'	The IMSCON keyword is not supported with the specified TYPE value.
X'00000008'	X'0000215F'	Invalid MODETBL value.
X'00000008'	X'00002160'	The MODETBL keyword is not supported with the specified TYPE value.
X'00000008'	X'00002161'	The NODE keyword is required when TYPE(VTAM) is specified.
X'00000008'	X'00002162'	Invalid NODE name.
X'00000008'	X'00002163'	The NODE keyword is not supported with the specified TYPE value.
X'00000008'	X'00002164'	The NODE name cannot be the local VTAM node name.
X'00000008'	X'00002165'	The TYPE keyword is not specified.
X'00000008'	X'00002166'	The RMTIMS keyword is required when TYPE(TCPIP) is specified.
X'00000008'	X'00002167'	Invalid RMTIMS value.
X'00000008'	X'00002168'	The RMTIMS keyword is not supported with the specified TYPE value.
X'00000008'	X'00002169'	The RMTIMS value cannot be the same as the local system ID.
X'00000008'	X'0000216A'	Invalid SESSION value.
X'00000008'	X'0000216B'	The SESSION keyword is not supported with the specified TYPE value.

Table 146. Return and reason codes for the CREATE MSPLINK command (continued)

Return code	Reason code	Meaning
X'0000000C'	X'00003000'	Command was successful for some resources but failed for others. The command output contains a line for each resource, which is accompanied by its completion code. See the completion code table for details.
X'0000000C'	X'00003004'	Command was not successful for any of the resources. The command output contains a line for each resource, which is accompanied by its completion code. See the completion code table for details.
X'00000010'	X'0000400C'	Command is not valid on the XRF alternate.
X'00000010'	X'00004014'	Command is not valid on the RSR tracker.
X'00000010'	X'0000402C'	Command is not valid because MSC is not enabled in the IMS system.
X'00000010'	X'0000402D'	Command is not allowed because dynamic definition is not enabled for MSC. MSCRSCS=DYN is not specified in the MSC section of the DFSDFxxx member of the IMS PROCLIB data set.
X'00000014'	X'00005004'	A DFSOCMD response buffer could not be obtained.
X'00000014'	X'00005008'	DFSPOOL storage could not be obtained.
X'00000014'	X'00005010'	Unable to obtain latch.

## Completion codes

The following table includes an explanation of the completion codes.

Table 147. Completion codes for the CREATE MSPLINK command

Completion code	Completion code text	Meaning
0		The command completed successfully for the physical link.
11	RESOURCE ALREADY EXISTS	A physical link with the same name already exists.
5F	INVALID CHARACTERS IN NAME	Name is invalid because it contains an invalid character.
60	GETMAIN STORAGE ERROR	A GETMAIN request for storage was unsuccessful.
61	DFSBCB STORAGE ERROR	DFSBCB storage error. Could not get storage for MSCP control block.
8A	WILDCARD PARAMETER NOT SUPPORTED	The CREATE command does not support wildcard parameters. You must explicitly specify the names of the resources you want to create.
90	INTERNAL ERROR	An IMS system service failure.
93	RESERVED NAME	Name is reserved. For example, name starts with DFS (except for DFSSAMxx or DFSIVPxx), or is an IMS reserved name such as BASICEDT or WTOR.

Table 147. Completion codes for the CREATE MSPLINK command (continued)

Completion code	Completion code text	Meaning
172	FIRST CHAR MUST NOT BE NUMERIC	The specified parameter value cannot begin with a numeric character.
200	NO DDNAME AVAILABLE FOR CRE	All DDNAMEs from DFSM00000 to DFSM99999 are in use.
201	UNABLE TO ALLOCATE DEVICE	Dynamic allocation of CTC device failed.
202	UNABLE TO OPEN CTC DEVICE	CTC DCB open failed.
203	ADDR IS NOT A CTC DEVICE	The device at the specified address is not a CTC adapter.
204	UNAVAILABLE CTC DEVICE	The CTC device at the address specified on the ADDR parameter is not available.
205	MSPLINK ITASK CREATE FAILED	Creation of MSPLINK ITASK failed.
206	CREATED LINK ENQUEUE FAILED	ENQUEUE LCB TO HASH TABLE ERROR
208	MAX LINK NUMBER REACHED	The CREATE MSPLINK command failed because the maximum number of MSC physical links are already defined. A maximum of 1018 physical links are allowed.

## Examples of the CREATE MSPLINK command

The following examples show the CREATE MSPLINK command creating different types of MSC physical links.

### CTC example

The following example of the CREATE MSPLINK command defines an MSC physical link with the following attributes:

- A name of PLNKC001
- A link type of CTC
- A CTC adapter address of 211

TSO SPOC input:

```
CRE MSPLINK NAME(PLNKC001) SET(TYPE(CTC) ADDR(211))
```

TSO SPOC output:

```
MSPLink MbrName    CC CText
PLNKC001 IMS1      0
PLNKC001 IMS2     204 UNAVAILABLE CTC DEVICE
```

OM API input:

```
CMD(CRE MSPLINK NAME(PLNKC001) SET(TYPE(CTC) ADDR(211)))
```

OM API output:

```
<imsout>
<ctl>
<omname>OM10M </omname>
```

```

<omvsn>1.7.0</omvsn>
<xmlvsn>20 </xmlvsn>
<statime>2014.274 15:39:27.567620</statime>
<stotime>2014.274 15:39:27.586550</stotime>
<staseq>CDD6E96DA7304D87</staseq>
<stoseq>CDD6E96DABCF6209</stoseq>
<rqsttkn1>USRT011 10083927</rqsttkn1>
<rc>0200000C</rc>
<rsn>00003000</rsn>
<rsnmsg>CSLN023I</rsnmsg>
<rsntxt>At least one request was successful.</rsntxt>
</ctl>
<cmderr>
<mbr name="IMS2 ">
<typ>IMS </typ>
<styp>DBDC </styp>
<rc>0000000C</rc>
<rsn>00003004</rsn>
<rsntxt>No requests were successful</rsntxt>
</mbr>
</cmderr>
<cmd>
<master>IMS2 </master>
<userid>USRT011 </userid>
<verb>CRE </verb>
<kwd>MSPLINK </kwd>
<input>CREATE MSPLINK NAME(PLNKC001) SET(TYPE(CTC)ADDR(211)) </input>
</cmd>
<cmdrsphdr>
<hdr s1b1="MSCP" l1b1="MSPLink" scope="LCL" sort="a" key="1" scroll="
no" len="8" dtype="CHAR" align="left" />
<hdr s1b1="MBR" l1b1="MbrName" scope="LCL" sort="a" key="2" scroll="no"
len="8" dtype="CHAR" align="left" />
<hdr s1b1="CC" l1b1="CC" scope="LCL" sort="n" key="0" scroll="yes"
len="4" dtype="INT" align="right" skipb="no" />
<hdr s1b1="CCTXT" l1b1="CCText" scope="LCL" sort="n" key="0"
scroll="yes" len="*" dtype="CHAR" skipb="yes" align="left" />
</cmdrsphdr>
<cmdrspdata>
<rsp>MSCP(PLNKC001) MBR(IMS2 ) CC( 204) CCTXT(UNAVAILABLE CTC
DEVICE) </rsp>
<rsp>MSCP(PLNKC001) MBR(IMS1 ) CC( 0) </rsp>
</cmdrspdata>
</imsout>

```

**Explanation:** The CREATE MSPLINK command is issued to create one MSC channel-to-channel physical link. The command is routed to both IMS1 and IMS2. The command fails on IMS2 with completion code 204, because that CTC address does not exist on IMS2. The command should have been routed only to IMS1. A DD name is not specified in the example because IMS generates the DDNAME automatically. The generated DCB DD names use the format DFSMnnnnn, where nnnn is the link number.

## MTM example

The following example of the CREATE MSPLINK command defines an MSC physical link with the following attributes:

- A name of PLNKM001
- A link type of MTM
- A buffer size of 4096 (default)

TSO SPOC input:

```
CRE MSPLINK NAME(PLNKM001) SET(TYPE(MTM))
```

#### TSO SPOC output:

```
MSPLink MbrName CC
PLNKM001 IMS1 0
PLNKM001 IMS2 0
```

#### OM API input:

```
CMD(CRE MSPLINK NAME(PLNKM001) SET(TYPE(MTM)))
```

#### OM API output:

```
<imsout>
<ctl>
<omname>OM10M </omname>
<omvsn>1.7.0</omvsn>
<xmlvsn>20 </xmlvsn>
<statime>2014.274 15:44:20.886013</statime>
<stotime>2014.274 15:44:20.901620</stotime>
<staseq>CDD6EA85621FD408</staseq>
<stoseq>CDD6EA8565EF440C</stoseq>
<rqsttkn1>USRT011 10084420</rqsttkn1>
<rc>00000000</rc>
<rsn>00000000</rsn>
</ctl>
<cmd>
<master>IMS2 </master>
<userid>USRT011 </userid>
<verb>CRE </verb>
<kwd>MSPLINK </kwd>
<input>CREATE MSPLINK NAME(PLNKM001) SET(TYPE(MTM)) </input>
</cmd>
<cmdrsphdr>
<hdr s1b1="MSCP" l1b1="MSPLink" scope="LCL" sort="a" key="1" scroll="no"
  len="8" dtype="CHAR" align="left" />
<hdr s1b1="MBR" l1b1="MbrName" scope="LCL" sort="a" key="2" scroll="no"
  len="8" dtype="CHAR" align="left" />
<hdr s1b1="CC" l1b1="CC" scope="LCL" sort="n" key="0" scroll="yes"
  len="4" dtype="INT" align="right" skipb="no" />
<hdr s1b1="CCTXT" l1b1="CCText" scope="LCL" sort="n" key="0"
  scroll="yes" len="*" dtype="CHAR" skipb="yes" align="left" />
</cmdrsphdr>
<cmdrspdata>
<rsp>MSCP(PLNKM001) MBR(IMS2 ) CC( 0) </rsp>
<rsp>MSCP(PLNKM001) MBR(IMS1 ) CC( 0) </rsp>
</cmdrspdata>
</imsout>
```

**Explanation:** The CREATE MSPLINK command is issued to create one MSC memory-to-memory physical link. The command is routed to both IMS1 and IMS2 and the physical link is created on both IMS systems.

### TCPIP example

The following example of the CREATE MSPLINK command defines an MSC physical link with the following attributes:

- A name of PLNKT001
- A link type of TCPIP
- A buffer size of 4096 (default)

TSO SPOC input:

```

|
| CRE MSPLINK NAME(PLNKT001) SET(TYPE(TCPIP)
|     ICONPLKID(MSC31) IMSCON(HWS3) RMTIMS(IMS2))
|

```

TSO SPOC output:

```

| MSPLink MbrName   CC
| PLNKT001 IMS1     0
|

```

OM API input:

```

| CMD(CRE MSPLINK NAME(PLNKT001)
| SET(TYPE(TCPIP) ICONPLKID(MSC31) IMSCON(HWS3) RMTIMS(IMS2)))
|

```

OM API output:

```

| <ctl>
| <omname>OM10M </omname>
| <omvsn>1.7.0</omvsn>
| <xmlvsn>20 </xmlvsn>
| <stime>2014.274 15:53:59.158782</stime>
| <stotime>2014.274 15:53:59.159591</stotime>
| <staseq>CDD6ECACDDFFEC16</staseq>
| <stoseq>CDD6ECACDE327A16</stoseq>
| <rqsttkn1>USRT011 10085359</rqsttkn1>
| <rc>00000000</rc>
| <rsn>00000000</rsn>
| </ctl>
| <cmd>
| <master>IMS1 </master>
| <userid>USRT011 </userid>
| <verb>CRE </verb>
| <kwd>MSPLINK </kwd>
| <input>CREATE MSPLINK NAME(PLNKT001)
|     SET(TYPE(TCPIP),ICONPLKID(MSC31),IMSCON(HWS3),RMTIMS(IMS2)) </input>
| </cmd>
| <cmdrsphdr>
| <hdr s1bl="MSCP" l1bl="MSPLink" scope="LCL" sort="a" key="1" scroll="
| no" len="8" dtype="CHAR" align="left" />
| <hdr s1bl="MBR" l1bl="MbrName" scope="LCL" sort="a" key="2" scroll="no"
| len="8" dtype="CHAR" align="left" />
| <hdr s1bl="CC" l1bl="CC" scope="LCL" sort="n" key="0" scroll="yes"
| len="4" dtype="INT" align="right" skipb="no" />
| <hdr s1bl="CCTXT" l1bl="CCText" scope="LCL" sort="n" key="0"
| scroll="yes" len="*" dtype="CHAR" skipb="yes" align="left" />
| </cmdrsphdr>
| <cmdrspdata>
| <rsp>MSCP(PLNKT001) MBR(IMS1 ) CC( 0) </rsp>
| </cmdrspdata>
| </imsout>
|

```

**Explanation:** The CREATE MSPLINK command is issued to create one MSC TCP/IP physical link on IMS1. MSC31 is the name that is used by IMS Connect to identify this physical link. HWS3 is the IMSplex name of the local IMS Connect instance that is providing TCP/IP support. IMS2 is the ID of the remote, partner IMS system.

## VTAM example

The following example of the CREATE MSPLINK command defines an MSC physical link with the following attributes:

- A name of PLNKT001
- A link type of VTAM
- A buffer size of 4096 (default)
- A partner IMS system that is defined to VTM as node L6APPL1

- A maximum session number of 1 (default).

TSO SPOC input:

```
CRE MSPLINK NAME(PLNKV001) SET(TYPE(VTAM) NODE(L6APPL1))
```

TSO SPOC output:

```
MSPLink MbrName CC
PLNKV001 IMS1 0
```

OM API input:

```
CMD(CRE MSPLINK NAME(PLNKV001) SET(TYPE(VTAM) NODE(L6APPL1)))
```

OM API output:

```
<imsout>
<ctl>
<omname>OM10M </omname>
<omvsn>1.7.0</omvsn>
<xmlvsn>20 </xmlvsn>
<statime>2014.274 14:45:08.904435</statime>
<stotime>2014.274 14:45:08.913582</stotime>
<staseq>CDD6DD49F31F3F8A</staseq>
<stoseq>CDD6DD49F55AE188</stoseq>
<rqsttkn1>USRT011 10074508</rqsttkn1>
<rc>00000000</rc>
<rsn>00000000</rsn>
</ctl>
<cmd>
<master>IMS1 </master>
<userid>USRT011 </userid>
<verb>CRE </verb>
<kwd>MSPLINK </kwd>
<input>CREATE MSPLINK NAME(PLNKV001) SET(TYPE(VTAM),NODE(L6APPL1))
</input>
</cmd>
<cmdrsphdr>
<hdr s1b1="MSCP" l1b1="MSPLink" scope="LCL" sort="a" key="1" scroll="
no" len="8" dtype="CHAR" align="left" />
<hdr s1b1="MBR" l1b1="MbrName" scope="LCL" sort="a" key="2" scroll="no"
len="8" dtype="CHAR" align="left" />
<hdr s1b1="CC" l1b1="CC" scope="LCL" sort="n" key="0" scroll="yes"
len="4" dtype="INT" align="right" skipb="no" />
<hdr s1b1="CCTXT" l1b1="CCText" scope="LCL" sort="n" key="0"
scroll="yes" len="*" dtype="CHAR" skipb="yes" align="left" />
</cmdrsphdr>
<cmdrspdata>
<rsp>MSCP(PLNKV001) MBR(IMS1 ) CC( 0) </rsp>
</cmdrspdata>
</imsout>
```

**Explanation:** The CREATE MSPLINK command is issued to create one MSC VTAM physical link on IMS1 with remote VTAM node (APPLID) name of L6APPL1.

---

## CREATE OTMADESC command

Use the CREATE OTMADESC command to create a OTMA destination descriptor without restarting IMS.

Subsections:

- “Environment” on page 251
- “Syntax” on page 251

- “Keywords” on page 252
- “Usage notes” on page 256
- “Output fields” on page 256
- “Return, reason, and completion codes” on page 257
- “Examples” on page 259

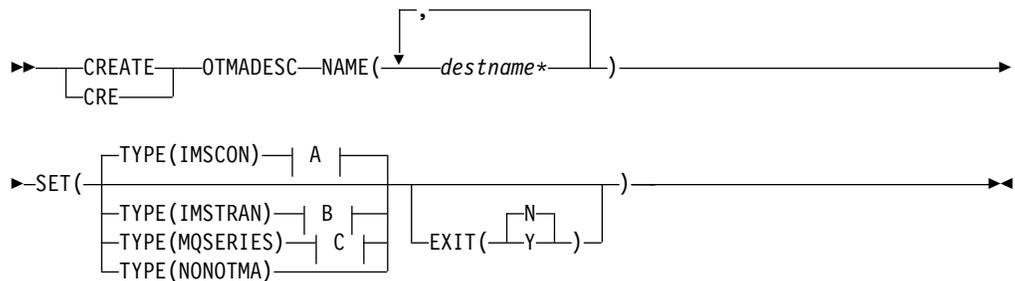
## Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the commands and keywords.

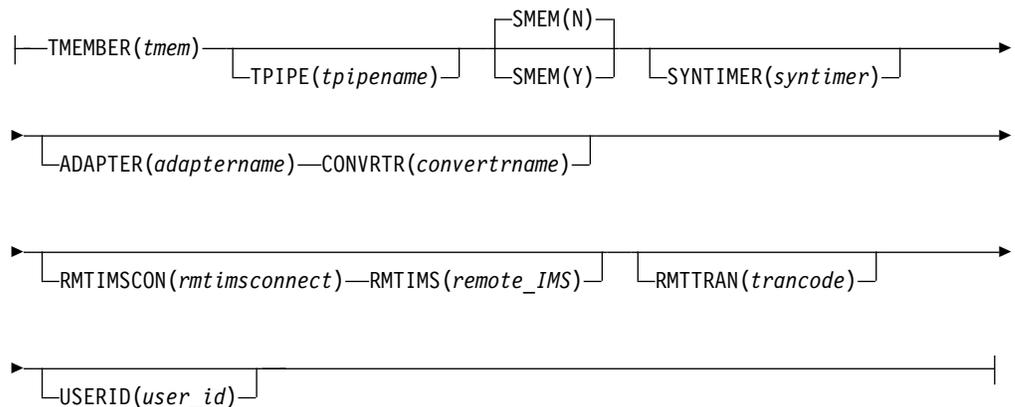
Table 148. Valid environments for the CREATE OTMADESC command and keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
CREATE OTMADESC	X		X
NAME	X		X
SET	X		X

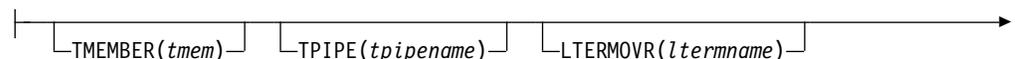
## Syntax

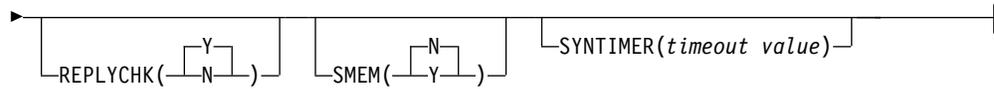


### A (TYPE=IMSCON):

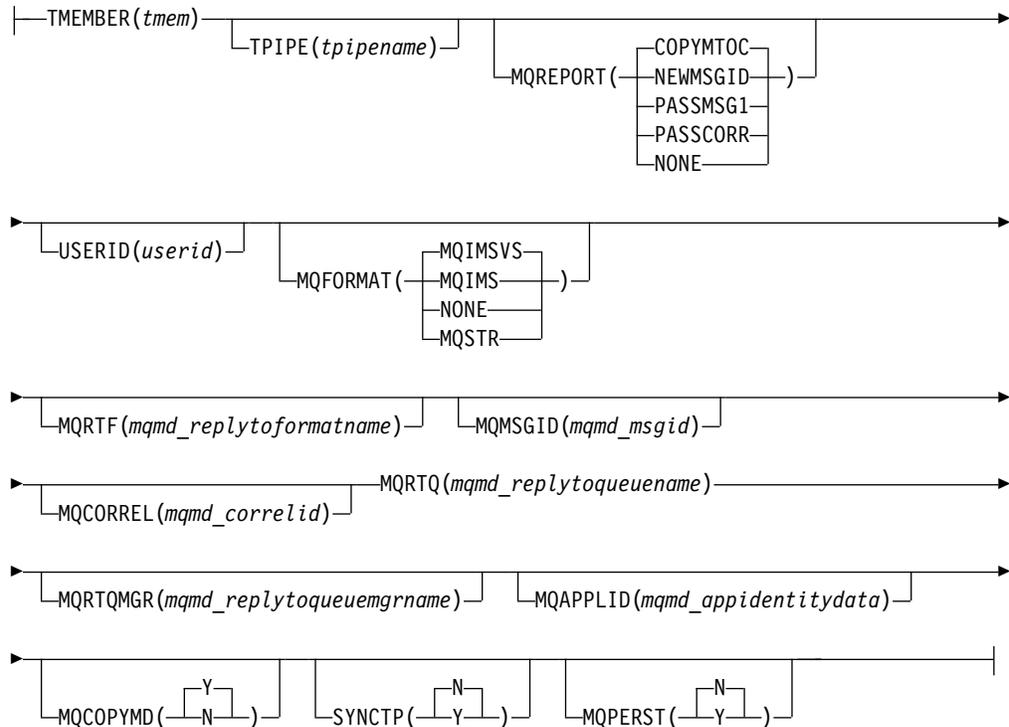


### B (TYPE=IMSTRAN):





### C (TYPE=MQSERIES):



## Keywords

The following keywords are valid for the CREATE OTMADESC command:

### ADAPTER(*adaptername*)

An optional keyword for TYPE(IMSCON). The *adaptername* is a 1- to 8-character name that identifies the IMS Connect adapter. If TYPE(NONOTMA) is specified, you cannot specify this keyword.

### CONVRTR(*convrtrname*)

A required parameter only if the ADAPTER keyword is specified. The *convrtrname* is a 1- to 8-character converter name used by the adapter. Coding this keyword without the ADAPTER keyword or with TYPE(NONOTMA) will cause an error.

### EXIT(N|Y)

When TYPE=IMSTRAN, this is an optional parameter used to indicate whether the IMS user exit (DFSCMUX0) can override the descriptor routing information for late messages.

For other descriptor types, this is an optional parameter used to specify whether the OTMA routing exits (DFSYPX0 and DFYDRU0) can override the descriptor switch destination from ALT IOPCB.

### LTERMVR(*ltermname*)

When TYPE=IMSTRAN, this is an optional parameter used to specify a logical terminal name that overrides the name specified in the application I/O PCB. If

an override name is also specified in the AIBRSNM2 field of the AIB, that name is used instead of the name in the descriptor. If neither the descriptor or the AIB contains an override name, the IMS application terminal symbolic (PSTSYMBO) is used as the default logical terminal name for the target transaction.

**MQAPPLID**(*mqmd\_appidentitydata*)

When TYPE=MQSERIES, this is an optional parameter that specifies a 1- to 32-character value for the MQMD\_APPLIDENTITYDATA field. This field is used to pass data to the receiving IBM MQ application. The default is 32 blanks.

**MQCOPYMD**(Y | N)

When TYPE=MQSERIES, this is an optional parameter that specifies whether the original input values of the MQMD data structure from the MQ IMS Bridge are copied to the output message.

**MQCORREL**(*mqmd\_correlid*)

When TYPE=MQSERIES, this is an optional parameter that specifies a 1- to 24-character value for the MQMD\_CORRELID field. This value is passed directly to the receiving IBM MQ application if the value of the MQREPORT parameter is PASSCORR. The default is 0.

**MQFORMAT**(MQIMSVS | MQIMS | NONE | MQSTR)

When TYPE=MQSERIES, this is an optional parameter that specifies value for the MQMD\_FORMAT field. The default value is MQIMSVS.

**MQMSGID**(*mqmd\_msgid*)

When TYPE=MQSERIES, this is an optional parameter that specifies a 1- to 24-character value for the MQMD\_MSGID field. This value is passed directly to the receiving IBM MQ application if the value of the MQREPORT parameter is PASSMSG1.

**MQPERST**(N | Y)

When TYPE=MQSERIES, this is an optional parameter that specifies whether messages that use this descriptor are persistent on a SYNC TPIPE. Setting this parameter to Y is analogous to setting X'40' for the output flag value of the DFSYDRU0 exit routine.

**MQREPORT**(COPYMTOC | NEWMSGID | PASSMSG1 | PASSCORR | NONE)

When TYPE=MQSERIES, this is an optional parameter that specifies the MQMD\_REPORT field value for messages that use the descriptor. The default value is COPYMTOC. Each supported value corresponds to an equivalent value in the MQMD data structure:

**Parameter value**

**Equivalent MQMD structure value**

**NEWMSGID**

MQRO\_NEW\_MSG\_ID

**PASSMSG1**

MQRO\_PASS\_MSG\_ID

**COPYMTOC**

MQRO\_COPY\_MSG\_ID\_TO\_CORREL\_ID

**PASSCORR**

MQRO\_PASS\_CORREL\_ID

**PASSMSCO**

MQRO\_PASS\_MSG\_ID and MQRO\_PASS\_CORREL\_ID

NONE  
MQRO\_NONE

**MQRTF(*mqmd\_replytoformat*)**

When TYPE=MQSERIES, this is a 1- to 8-character value for the MQMD\_REPLYTOFORMAT field. The default is 0.

**MQRTQ(*mqmd\_replytoqueue*)**

When TYPE=MQSERIES, this is a required parameter that specifies a 1- to 48-character value for the MQMD\_REPLYTOQ field.

**MQRTQMGR(*mqmd\_replytoqueue*)**

When TYPE=MQSERIES, this is an optional parameter that specifies a 1- to 48-character value for the MQMD\_REPLYTOQMGR field. If this parameter is not specified, MQSeries® will use the local queue manager.

**NAME(*destname*\*)**

A required keyword parameter that specifies a 1- to 8-character destination name. The destination name can have an asterisk (\*) at the end to mask groups of names. An asterisk by itself is a valid name and is an entry in the table of destination routing descriptors. A masked name that encompasses another masked name does not have to be created in the order from most specific to most generic. However, the search order starts from the most specific to the most generic.

Creating a destination name that already exists in the system is an error.

**REPLYCHK(Y|N)**

Specifies whether an IMS application replies to the IOPCB for a synchronous program switch request. When REPLYCHK=YES and the ICAL target application does not reply to the IOPCB or switch to another transaction, IMS gives the error return code X'0100', with reason code X'0110', and extended reason code X'0014' instead of a timeout to the ICAL call. The default is Y.

**RMTIMS**

An optional 1- to 8-character name of a remote, destination IMS system for ALTPCB output messages. When RMTIMS is specified, you must also specify the RMTIMSCON parameter. The RMTIMS value must match the value specified on the ID parameter of a DATASTORE statement in the configuration member of the remote IMS Connect instance. This parameter is valid only when TYPE(IMSCON) is specified.

**RMTIMSCON**

An optional 1- to 8-character name of a connection to a remote IMS Connect instance, as defined on ID parameter of the RMTIMSCON statement in the HWSCFGxx PROCLIB member of the local IMS Connect. When RMTIMSCON is specified, you must also specify RMTIMS. This parameter is valid only when TYPE(IMSCON) is specified.

**RMTTRAN**

An optional 1- to 8-character name of the transaction to use at the remote, destination IMS system. When this parameter is specified with RMTIMSCON and RMTIMS parameters, OTMA passes the transaction code to IMS Connect, which inserts it into the outgoing message. This parameter is valid only when TYPE(IMSCON) is specified.

**SMEM(N | Y)**

An optional parameter that can either be a Y or N value to indicate whether the TMEMBER name specified in the TMEMBER parameter is a super member.

If the TMEMBER name is a super member, the length of the TMEMBER name has a maximum of 4 characters. SMEM and TYPE(NONOTMA) are mutually exclusive.

#### **SYNCTP(N | Y)**

When TYPE=MQSERIES, this parameter specifies whether a SYNC TPIPE must be created for messages that use this descriptor. Setting this parameter to Y is analogous to setting X'80' for the output flag value of the DFSYDRU0 exit routine.

#### **SYNTIMER(*syntimer*)**

An optional parameter that specifies the timeout value that when reached, the synchronous callout reply expires because an ACK/NAK or reply from the client has not been received. The value *syntimer*, which is expressed in hundredths of a second, must be numeric and within a range of 0 through 999999, inclusive. When the parameter is coded, a numeric value must be specified. If SYNTIMER(0) is specified, the value defaults to 1000, which is equivalent to 10 seconds.

If TYPE(IMSTRAN) is specified, this parameter is the timeout value for synchronous program switch ICAL calls. If a timeout value is also specified in the AIB interface, IMS will compare the timeout values and use the lower value.

#### **TMEMBER(*tmem*)**

A required parameter for TYPE(IMSCON). The *tmem* value is a 1- to 16-character OTMA TMEMBER name. Coding this keyword with TYPE(NONOTMA) causes an error. If SMEM(Y) is specified, *tmem* is a maximum of 4-character super member name.

An optional parameter for TYPE(IMSTRAN). The *tmem* value is a 1- to 16-character OTMA TMEMBER name. IMS queues late responses from a synchronous program switch to this OTMA TMEMBER. The destination name is used as the default OTMA TPIPE name if the TPIPE parameter is not specified in the descriptor.

#### **TPIPE(*tpipename*)**

An optional parameter that is a 1- to 8-character TPIPE name when TYPE(IMSCON) is specified. If this keyword is not coded, the TPIPE name is the destination name specified in the NAME keyword. This keyword is not valid if TYPE(NONOTMA) is specified. If the TPIPE is not coded and the NAME is a mask, the TPIPE will be a specific destination name that matched the mask in the table of destination routing descriptors and not the NAME keyword.

An optional parameter for TYPE(IMSTRAN) that specifies a 1- to 8-character TPIPE name. This parameter defaults to the destination name. IMS queues late responses from a synchronous program switch to this OTMA TPIPE. A value for the TMEMBER parameter is required when TPIPE is specified.

#### **TYPE(IMSCON | IMSTRAN | MQSERIES | NONOTMA)**

Specifies the type of destination:

##### **IMSCON**

An IMS Connect client. This is the default value.

##### **IMSTRAN**

An IMS transaction.

##### **MQSERIES**

IBM MQ.

### **NONOTMA**

A non-OTMA destination. Only the EXIT parameter is valid with this descriptor type.

### **USERID(*userid*)**

For TYPE=IMSCON, this is an optional 1- to 8-character RACF user ID. When this parameter is specified with RMTIMSCON, RMTTRAN, and RMTIMS parameters, a remote, destination IMS system uses the USERID value to perform transaction authorization. The value of USERID specified in the OTMA destination descriptor overrides the user ID provided by the IMS application program that issued the ISRT call to the OTMA ALTPCB. This parameter is valid only when TYPE(IMSCON) is specified.

For TYPE=MQSERIES, this is a 1- to 8-character user ID that corresponds with the MQMD\_USERIDENTIFIER field in the MQMD data structure. This field allows the OTMA descriptor to override the default user ID in the MQMD structure. Specify the value \*USERID\* to use the IMS user ID value from the PST.

## **Usage notes**

The CREATE OTMADESC command is used to facilitate the creation of OTMA destination descriptors while IMS is up and running. The user can add new routing descriptors dynamically in addition to the statically defined destination routing descriptors in DFSYDTx of IMS.PROCLIB. The order in which the destination routing descriptors are coded is unrestricted and need not be coded in the order from most specific to most generic destination routing descriptor name. However, the search order in the table of destination routing descriptors will be performed from most specific to most generic.

**Note:** Note the distinction between the coding in DFSYDTx and the search order that is performed internally in the table of destination routing descriptors. In DFSYDTx, the rule of coding does not require the order from most specific to most generic; however, the search order in the table of destination routing descriptor is performed from most specific to most generic.

The destination routing descriptors are part of the checkpoint records used for persistency on IMS restart. Also, all CREATE OTMADESC commands will generate log records. The checkpoint log record is X4035 and the log record for CREATE, DELETE, and UPDATE is X221B. During an IMS warm restart, the destination routing descriptors will be rebuilt based on the checkpoint records. In case of an ERE restart, the log records will be used to completely rebuild the destination routing descriptors after the checkpoint records have been processed. Because of the rebuilding process, the routing descriptors in member DFSYDTx in PROCLIB will not be read and processed. However, during cold start processing, the routing descriptors in member DFSYDTx will be read and processed.

The CREATE OTMADESC command is not valid in an XRF alternate, RSR tracking, or an FDBR region environment. The checkpoint and log records will be used to update the destination routing descriptors in these environments. The CREATE OTMADESC command becomes a valid command only when the XRF alternate or the RSR tracking environment becomes the active IMS environment.

## **Output fields**

The following table shows the CREATE OTMADESC output fields. The columns in the table are:

**Short label**

Contains the short label generated in the XML output.

**Long label**

Contains the column heading for the output field in the formatted output.

**Keyword**

Identifies the keyword on the command that caused the field to be generated. N/A appears for output fields that are always returned. *error* appears for output fields that are returned only in case of an error.

**Meaning**

Provides a brief description of the output field.

Table 149. Output fields for the CREATE OTMADESC command

Short label	Long label	Keyword	Meaning
CC	CC	N/A	Completion code for the line of output. The completion code indicates whether IMS was able to process the command for the specified resource. See "Return, reason, and completion codes" for more information. The completion code is always returned.
CCTXT	CCText	N/A	Completion code text that briefly explains the meaning of the non-zero completion code. This field is returned only for an error completion code.
DEST	DestName	NAME	Destination name.
MBR	MbrName	N/A	Member name.

## Return, reason, and completion codes

An IMS return and reason code is returned to OM by the CREATE OTMADESC command. The OM return and reason codes that might be returned as a result of the CREATE OTMADESC command are standard for all commands entered through the OM API.

The following table includes the return and reason codes and a brief explanation of the codes. The return or reason code returned for the command might also indicate an error from a CSL request.

Table 150. Return and reason codes for the CREATE OTMADESC command

Return code	Reason code	Meaning
X'00000000'	X'00000000'	Command completed successfully. The command output contains a line for each resource, accompanied by its completion code. See the completion codes table for details.
X'02000008'	X'00002000'	The command contains an invalid verb or no client is registered for the verb.
X'02000008'	X'00002004'	The command contains an invalid primary keyword or no client is registered for the keyword.
X'02000008'	X'00002028'	The command contains an invalid keyword.
X'02000008'	X'0000202C'	The command contains an unknown positional parameter.
X'02000008'	X'00002034'	The command contains an incomplete keyword parameter.
X'02000008'	X'00002038'	The command is missing a required parameter.

Table 150. Return and reason codes for the CREATE OTMADESC command (continued)

Return code	Reason code	Meaning
X'02000008'	X'0000203C'	The command contains an invalid keyword parameter value.

The following table includes an explanation of the completion codes. Errors unique to the processing of this command are returned as completion codes. A completion code is returned for each action against an individual resource.

Table 151. Completion codes for the CREATE OTMADESC command

Completion code	Completion code text	Meaning
0	Command completed successfully	The CREATE OTMADESC command completed successfully for the resource.
153	Adapter blank, Convrtr not blank	Both ADAPTER() and CONVRTR() keywords must be either both blanks or both contain valid names.
154	Adapter not blank, Convrtr blank	Both ADAPTER() and CONVRTR() keywords must be either both blanks or both contain valid names.
155	NONOTMA, other parms specified	For TYPE(NONOTMA), no other keywords must be specified, such as TYPE(), TMEMBER(), SMEM(), ADAPTER(), or CONVRTR().
156	SMEM=Y, super mbr name GT 4 char	For SMEM(Y), the TMEMBER value must not be greater than 4 characters.
157	SMEM=Y, Tmember parm not specfd	The TMEMBER keyword is a required parameter when SMEM(Y) is specified.
159	TMEMBER is required for IMSCON	For TYPE(IMSCON), a value for the TMEMBER keyword is required and must be specified.
162	Descriptor already exist in tbl	The descriptor being created already exists in the table of destination routing descriptors.
167	SYNTIMER must have numeric value	The timeout value must have a value expressed in numbers within parenthesis.
169	SYNTIMER has nonnumeric value	The value must not contain alphabetic characters or any character that is not numeric.
16A	RmtIMSCon blank, RmtIMS not blank	The RMTIMS and RMTIMSCON keywords were specified on the CREATE OTMADESC command, but no value was entered for RMTIMSCON. When a value is specified on the RMTIMS keyword, a value must also be specified on the RMTIMSCON keyword.

Table 151. Completion codes for the CREATE OTMADESC command (continued)

Completion code	Completion code text	Meaning
16B	RmtIMSCon not blank, RmtIMS blank	The RMTIMS and RMTIMSCON keywords were specified on the CREATE OTMADESC command, but no value was entered for RMTIMS. When a value is specified on the RMTIMSCON keyword, a value must also be specified on the RMTIMS keyword.

## Examples

The following are examples of the CREATE OTMADESC command:

### *Example 1 for CREATE OTMADESC command*

TSO SPOC input:

```
CREATE OTMADESC NAME(OTMACL*) SET(TYPE(IMSCON) TMEMBER(HWS2))
```

TSO SPOC output:

DestName	MbrName	CC
OTMACL*	IMSA	0

**Explanation:** In the example, the CREATE OTMADESC command is issued with NAME(OTMACL\*) and the descriptor OTMACL\* is included in the table of destination routing descriptors.

### *Example 2 for CREATE OTMADESC command*

TSO SPOC input:

```
CREATE OTMADESC NAME(OTMACL99) SET(TYPE(IMSCON) TMEMBER(HWS1) TPIPE(HWS1TP01))
```

TSO SPOC output:

DestName	MbrName	CC
OTMACL99	IMSA	0

**Explanation:** If the CREATE OTMADESC command is issued with the TPIPE keyword, the descriptor OTMACL99 is included in the table of destination routing descriptors. Although not shown in the example, a CREATE command issued for OTMACL\* would be ahead of the CREATE for OTMACL99.

### *Example 3 for CREATE OTMADESC command*

TSO SPOC input:

```
CREATE OTMADESC NAME(OTMACL01) SET(TYPE(IMSCON) TMEMBER(ICON1) RMTIMSCON(ICON2) RMTIMS(IMS2) RMTTRAN(TRAN01) USERID(USER01))
```

TSO SPOC output:

DestName	MbrName	CC
OTMACL01	IMS1	0

**Explanation:** The CREATE OTMADESC command shown in this example creates an OTMA destination descriptor called OTMACL01 that routes ALTPCB output

messages to a remote IMS system via a TCP/IP connection between two instances of IMS Connect. The command also defines the following values on the OTMA destination descriptor:

- The local instance of IMS Connect, as specified by TMEMBER(ICON1).
- The remote IMS Connect connection, as specified by RMTIMSCON(ICON2) in the local IMS Connect configuration member. This value is also specified on the ID parameter of the RMTIMSCON configuration statement of the local IMS Connect instance.
- The remote IMS system, as specified by RMTIMS(IMS2).
- The transaction to schedule at the remote IMS system, as specified by RMTTRAN(TRAN01).
- The user ID to be used for transaction authorization at the remote IMS system, as specified by USERID(USER01).

When the OTMA messages reach IMS2, IMS2 schedules TRAN01 to process the messages. The descriptor OTMACL01 is included in the table of destination routing descriptors.

**Related concepts:**

 [How to interpret CSL request return and reason codes \(System Programming APIs\)](#)

**Related tasks:**

 [Specifying a remote transaction code \(Communications and Connections\)](#)

**Related reference:**

Chapter 2, “Command keywords and their synonyms,” on page 63

---

## CREATE PGM command

Use the CREATE PGM command to define the program resource requirements for application programs that run under the control of the DB/TM environment, as well as for application programs that access databases through DBCTL.

The program resource describes an application program that operates in a message processing region, Fast Path message-driven program region, batch processing region, batch message processing region, or CCTL threads. Program resources combined with transactions define the scheduling and resource requirements for a message-driven application program.

**Subsections:**

- “Environment”
- “Syntax” on page 261
- “Keywords” on page 261
- “Usage notes” on page 265
- “Output fields” on page 266
- “Return, reason, and completion codes” on page 266
- “Examples” on page 269

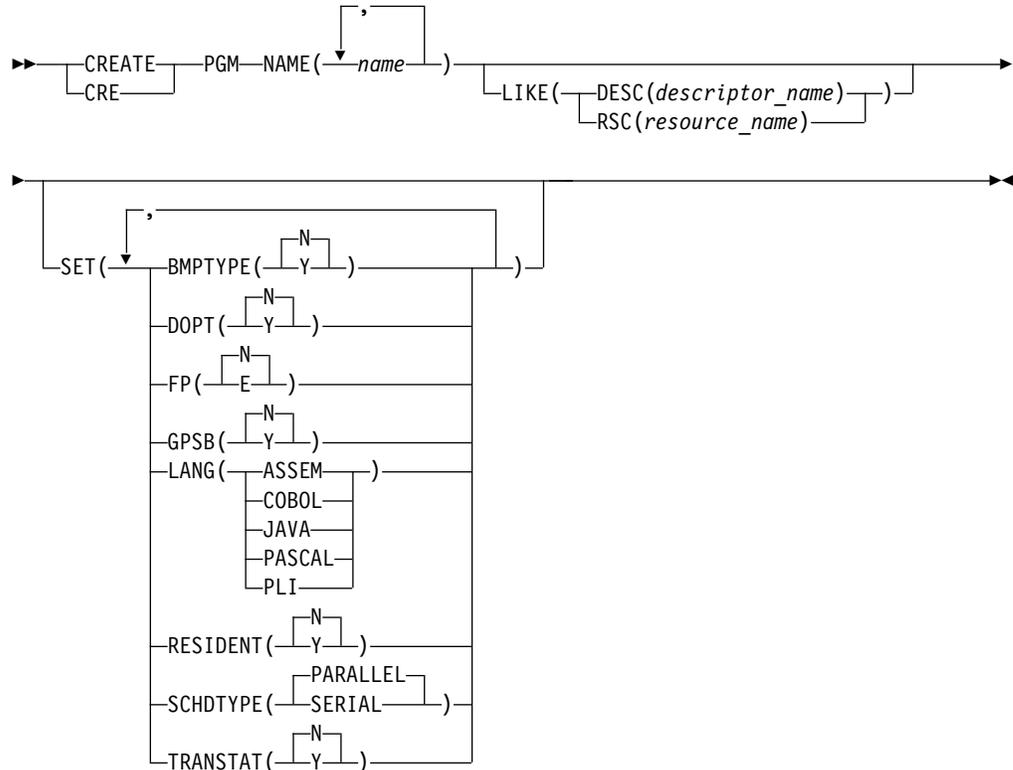
### Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

Table 152. Valid environments for the CREATE PGM command and keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
CREATE PGM	X	X	X
LIKE	X	X	X
NAME	X	X	X
SET	X	X	X

## Syntax



Default values shown in this syntax diagram reflect the default values that are defined in the IMS program descriptor, DFSDSPG1. If you define another descriptor as the default with a CREATE or UPDATE PGMDESC command, the default values shown in this syntax diagram might not match the value defined in the current default descriptor.

## Keywords

The following keywords are valid for the CREATE PGM command:

### LIKE

Specifies that the resource is created using an existing resource or descriptor as a model, instead of the default descriptor. The default descriptor is either the IMS descriptor DFSDSPG1, or user-defined. The model type can either be a descriptor (DESC), or a resource (RSC). The new resource is created with all the same attributes as the model. Attributes set explicitly by the CREATE

command override the model attributes. Later changes to the model are not propagated to resources or descriptors that were created from it.

The QUERY commands can be used to display the model name and model type used to create the resource. The model name and type returned from the QUERY command are provided for informational purposes only. The EXPORT command exports a resource definition, including the model name and model type, to the definition member. The IMPORT command imports a resource definition, including the model name and model type, from the definition member. An IMPORT command that creates a resource saves the model name and model type, but does not use the models attributes.

**DESC(descriptor\_name)**

Specifies the name of the descriptor to use as a model to define this resource.

**RSC(resource\_name)**

Specifies the name of the resource to use as a model to define this resource.

**NAME**

Specifies the 1-8 character name of the program. Names must be alphanumeric (A through Z, 0 through 9, #, \$, and @). All program names must begin with an alphabetic character (A through Z, #, \$, and @) followed by zero to seven alphanumeric characters. Names cannot include a blank, comma, period, hyphen, or equal sign. Names cannot begin with DFS, except for DFSIVPxx and DFSSAMxx. Names cannot be BASICEDT, DBCDM, DBF#FPU0, DBRC, ISCEDT, MSDB, SDB, or WTOR. Each local program name must be unique. If the program is defined with BMPTYPE(N) and FP(N), the program name and associated PSB name must be the same. A program resource and a program descriptor can have the same name.

**SET**

Specifies the attributes of the program to be created. If the LIKE keyword is omitted, attributes not specified take the value defined in the default descriptor, which is either IMS descriptor DFSDSPG1 or user-defined. If the LIKE keyword is specified, attributes not specified take the value defined in the model.

**BMPTYPE()**

BMP type option. Specifies whether the program executes in a BMP type region or not. A BMP type region can be a BMP region or a JBP region. PSBs scheduled by DB2 stored procedures, by programs running under WebSphere® Application Server, and by other users of the ODBA interface may be defined with BMPTYPE Y or N.

**N** The program does not execute in a BMP type region. It might execute in an IMS TM MPP, JMP, or IFP region or it might use either the ODBA or DRA interface. This specification should be used for programs running in IMS TM MPP, JMP, and IFP regions, or PSBs scheduled by CICS® programs using DBCTL and other users of the DRA interface. This is the default.

**Y** The program executes in a BMP type region. It might execute in an IMS BMP region or a JBP region. Any associated transactions are assigned normal and limit priority values of zero.

FP(E) and BMPTYPE(Y) are mutually exclusive.

## DOPT

Specifies the dynamic option.

**N** The PSB associated with this program is not located dynamically. The control blocks for the PSB must exist in an ACBLIB or the IMS catalog, otherwise the program is set to a NOTINIT status, and cannot be scheduled. This is the default.

**Y** The PSB associated with this program is located dynamically. Each time the program that is associated with this PSB is scheduled, the latest copy of the PSB is loaded. When the program terminates, the PSB is deleted from the PSB pool.

Until the PSB is required to process a transaction, the ACB for the PSB does not need to be in the ACBLIB or, if the IMS management of ACBs is enabled, in the IMS catalog. A new version of the PSB can be defined and added to the ACBLIB or the IMS catalog at any time. IMS loads the new version of the ACB the next time the PSB is scheduled.

A DOPT PSB cannot be scheduled if the control blocks for the DBDs that the PSB references are not in the ACBLIB or the IMS catalog.

Keyword combination rules include the following:

- DOPT(Y) and GPSB(Y) are mutually exclusive.
- DOPT(Y) and LANG(JAVA) is a valid combination.
- LANG is invalid with GPSB(N), except if DOPT(Y) and LANG(JAVA).
- LANG(JAVA), DOPT(Y), and GPSB(N) is a valid combination.
- RESIDENT(Y) and DOPT(Y) are mutually exclusive.
- SCHDTYPE(PARALLEL) and DOPT(Y) are mutually exclusive.

**FP** Specifies the Fast Path option.

### E

The program is a Fast Path-exclusive program. This implicitly defines a wait-for-input (WFI) program. Either a transaction or a routing code that can be assigned by the user Input Edit/Routing exit routine must be defined for the Fast Path-exclusive program, in order for this program to be usable.

Fast Path must be defined in the IMS system.

**N** The program is not a Fast Path-exclusive program. This is the default. When FP(N) is specified, any attempt to use Fast Path resources or commands will yield unpredictable results.

Keyword combination rules include the following:

- FP(E) requires that Fast Path be defined.
- LANG(JAVA) and FP(E) are mutually exclusive.
- BMPTYPE(Y) and FP(E) are mutually exclusive.

## GPSB

Specifies the generated PSB option.

**N** The PSB associated with the program is not generated by IMS. The control blocks for the PSB must exist in the ACBLIB or, if the IMS management of ACBs is enabled, in the IMS catalog, otherwise the program is set to a NOTINIT status, and cannot be scheduled.

**Y** The PSB and application control block (ACB) that are associated with

the program are generated by IMS. The scheduling process of all environments generates a PSB that contains an I/O PCB and an alternate modifiable PCB.

You do not need to generate the PSB and ACB or submit DDL to define the PSB. The ACB for the PSB is not loaded from either ACBLIB or, if the IMS management of ACBs is enabled, the IMS catalog. Thus, I/O to retrieve the ACB from either the ACBLIB or the IMS catalog is eliminated.

The generated PSB contains an I/O PCB named IOPCBbbb and a modifiable, alternate PCB named TPPCB1bb. With an alternate modifiable PCB, an application can use the CHNG call to change the output destination and send output to a destination other than the input destination.

Keyword combination rules include the following:

- DOPT(Y) and GPSB(Y) are mutually exclusive.
- DOPT(Y) and LANG(JAVA) is a valid combination.
- GPSB(Y) requires LANG.
- LANG is invalid with GPSB(N), except if DOPT(Y) and LANG(JAVA).
- LANG(JAVA), DOPT(Y), and GPSB(N) is a valid combination.
- RESIDENT(Y) and GPSB(Y) are mutually exclusive.

#### **LANG**

Specifies the language interface of the program for a GPSB, or defines a DOPT(Y) program as using the Java™ language.

In order to define a DOPT program using the Java language, the program must be defined with DOPT(Y) and LANG(JAVA). DOPT PSBs are not loaded at IMS restart, they are loaded every time the program is scheduled. When the program is scheduled for the first time, IMS does not know the language until after the program is scheduled in a region and the PSB is loaded. Unless LANG(JAVA) is defined for the DOPT(Y) program, the program is incorrectly scheduled in a non-Java region.

<b>LANG parameter</b>	<b>Meaning</b>
ASSEM	Assembler
COBOL	COBOL
JAVA	Java
PASCAL	Pascal
PLI	PL/I

Keyword combination rules include the following:

- LANG is invalid with GPSB(N), except if DOPT(Y) and LANG(JAVA).
- DOPT(Y) and LANG(JAVA) is a valid combination.
- LANG(JAVA), DOPT(Y), and GPSB(N) is a valid combination.
- LANG(JAVA) and FP(E) are mutually exclusive.

#### **RESIDENT**

Specifies the resident option. The RESIDENT(N) option takes effect right away. The RESIDENT(Y) option takes effect at the next restart, unless an error is

encountered such as no PSB in ACBLIB for the program, or if the program was created as RESIDENT(Y) after the checkpoint from which this IMS is performing emergency restart.

- N** Use N to specify that the PSB associated with the program is not made resident in storage. The PSB is loaded at scheduling time. This is the default. The RESIDENT(N) option takes effect immediately.
- Y** The PSB associated with the program is made resident in storage at IMS cold start or restart. IMS loads the PSB and initializes it. A resident PSB is accessed in local storage, which avoids I/O to retrieve the associated ACB.

Keyword combination rules include the following:

- DOPT(Y) and RESIDENT(Y) are mutually exclusive.
- GPSB(Y) and RESIDENT(Y) are mutually exclusive.

#### **SCHDTYPE**

Specifies whether this program can be scheduled into more than one message region or batch message region simultaneously.

#### **PARALLEL**

The program can be scheduled in multiple regions simultaneously. This is the default.

#### **SERIAL**

The program can be scheduled in only one region at a time.

Keyword combination rules include the following:

- DOPT(Y) and SCHDTYPE(PARALLEL) are mutually exclusive.

#### **TRANSTAT**

Specifies whether transaction level statistics should be logged. The value specified has meaning only if the program is a JBP or a non-message driven BMP. If Y is specified, transaction level statistics are written to the log in a X'56FA' log record.

- N** Transaction level statistics should not be logged.
- Y** Transaction level statistics should be logged.

The TRANSTAT keyword is optional. If a value is not specified for the TRANSTAT keyword, the system default is used. The system default for the transaction level statistics parameter is set with the TRANSTAT parameter in the DFSDFxxx PROCLIB member. If a value is not specified for the TRANSTAT parameter, the system default is set to N. The TRANSTAT keyword on the CREATE PGM command gives the user the ability to override the system default when creating a program resource.

### **Usage notes**

BLDL is performed on the PSB associated with this program (except DOPT or GPSB). If the PSB is not defined in ACBLIB, the program is created anyway and the command results in a completion code of zero, even though it cannot be scheduled. The program status is NOTINIT when a /DIS PGM or QRY PGM is issued for the program. Online Change must be used to add the PSB to ACBLIB, before the application program can be scheduled.

Resources exist for the life of the IMS unless they are deleted using a DELETE command. The resources and descriptors are recoverable across an IMS warm start

or emergency restart. Resources and descriptors are lost if IMS is cold started, unless cold start imports definitions that were exported while IMS was up.

The CREATE PGM command is similar to local online change (using a /MODIFY command) or global online change (INITIATE OLC) for application program resources in the MODBLKS data set, except that programs are added dynamically.

A CREATE PGM command can be issued only through the Operations Manager API. This command applies to DB/DC, DBCTL and DCCTL systems.

These commands are not valid on the XRF alternate, RSR tracker, or FDBR region. The CREATE command is not valid if online change for MODBLKS is enabled (DFSDFxxx or DFSCGxxx defined with MODBLKS=OLC or MODBLKS not defined).

When a program is created, if the PSB is in the ACBLIB library, the PSB is loaded into the PSB pool at first schedule. If the program is created with RESIDENT(Y), the PSB is made resident at the next IMS restart. If the PSB is not in the ACBLIB library, the program is created anyway, but marked bad because it is unusable. A QRY PGM command specified with SHOW(ALL) or SHOW(STATUS) shows the program status as NOTINIT-40-NOPSB, which means there is no PSB in the ACBLIB library for it. You must use online change for the ACBLIB library to add the PSB before the program can be used

## Output fields

The following table shows the CREATE PGM output fields. The columns in the table are as follows:

### Short label

Contains the short label generated in the XML output.

### Keyword

Identifies keyword on the command that caused the field to be generated. N/A appears for output fields that are always returned. *error* appears for output fields that are returned only in case of an error.

### Meaning

Provides a brief description of the output field.

Table 153. Output fields for the CREATE PGM command

Short label	Keyword	Meaning
CC	N/A	Completion code.
CCTXT	<i>error</i>	Completion code text that briefly explains the meaning of the non-zero completion code.
MBR	N/A	IMSplex member that built the output line.
PGM	PGM	Program name.

## Return, reason, and completion codes

The following table includes the return and reason codes and a brief explanation of the codes. The return or reason code returned for the command might also indicate an error from a CSL request.

Table 154. Return and reason codes for CREATE PGM

Return code	Reason code	Meaning
X'00000000'	X'00000000'	Command completed successfully. The command output contains a line for each resource, accompanied by its completion code. See the completion code table for details.
X'00000008'	X'00002009'	Multiple values specified for the same attribute are invalid. For example, you cannot specify both BMPTYPE(Y) and BMPTYPE(N). This reason code is applicable only when you create a program by using the DFSINSX0 user exit.
X'00000008'	X'00002048'	Invalid SET attribute.
X'00000008'	X'00002106'	DOPT(Y) mutually exclusive with RESIDENT(Y). Either both keywords were explicitly specified, or one keyword was explicitly specified and the other attribute was gotten from the model.
X'00000008'	X'00002107'	DOPT(Y) mutually exclusive with SCHDTYPE(PARALLEL). Either both keywords were explicitly specified, or one keyword was explicitly specified and the other attribute was gotten from the model.
X'00000008'	X'0000210B'	FP(E) mutually exclusive with BMPTYPE(Y). Either both keywords were explicitly specified, or one keyword was explicitly specified and the other attribute was gotten from the model.
X'00000008'	X'0000210D'	FP(E) mutually exclusive with LANG(JAVA). Either both keywords were explicitly specified, or one keyword was explicitly specified and the other attribute was gotten from the model.
X'00000008'	X'00002113'	GPSB(Y) mutually exclusive with DOPT(Y). Either both keywords were explicitly specified, or one keyword was explicitly specified and the other attribute was gotten from the model.
X'00000008'	X'00002114'	GPSB(N) mutually exclusive with LANG.
X'00000008'	X'00002115'	GPSB(Y) mutually exclusive with RESIDENT(Y). Either both keywords were explicitly specified, or one keyword was explicitly specified and the other attribute was gotten from the model.
X'00000008'	X'00002117'	Invalid LIKE descriptor name.
X'00000008'	X'00002118'	Invalid LIKE resource name.
X'00000008'	X'00002131'	GPSB(Y) requires LANG to be specified.
X'00000008'	X'00002132'	DOPT(Y) not supported with LANG except for LANG(JAVA). Either both keywords were explicitly specified, or one keyword was explicitly specified and the other attribute was gotten from the model.
X'0000000C'	X'00003000'	Command was successful for some resources but failed for others. The command output contains a line for each resource, accompanied by its completion code. See the completion code table for details.
X'0000000C'	X'00003004'	Command was not successful for any of the resources. The command output contains a line for each resource, accompanied by its completion code. See the completion code table for details.

Table 154. Return and reason codes for CREATE PGM (continued)

Return code	Reason code	Meaning
X'00000010'	X'0000400C'	Command is not valid on the XRF alternate.
X'00000010'	X'00004014'	Command is not valid on the RSR tracker.
X'00000010'	X'00004024'	No FP defined, so FP(E) invalid.
X'00000010'	X'00004120'	Online change phase is in progress.
X'00000010'	X'00004204'	Model is quiesced. Cannot quiesce model.
X'00000010'	X'00004300'	Command is not allowed because online change for MODBLKS is enabled (DFSDFxxx or DFSCGxxx defined with MODBLKS=OLC, or MODBLKS not defined).
X'00000014'	X'00005004'	DFSOCMD response buffer could not be obtained.
X'00000014'	X'00005008'	DFSPPOOL storage could not be obtained.
X'00000014'	X'0000500C'	AWE could not be obtained.
X'00000014'	X'00005010'	Unable to obtain latch.
X'00000014'	X'00005114'	Resource or descriptor that is specified as a model is in the process of being imported from the change list in the IMSRSC repository or was not imported successfully from the change list at the end of IMS restart. This resource or descriptor cannot be referenced as a model on a CREATE command until it is successfully imported from the repository.

Errors unique to the processing of this command are returned as completion codes. The following table includes an explanation of the completion codes.

Table 155. Completion codes for the CREATE PGM command

Completion code	Completion code text	Meaning
0		Command completed successfully for the program.
11	RESOURCE ALREADY EXISTS	Program exists.
5F	INVALID CHARACTERS IN NAME	Name is invalid because it contains an invalid character.
61	DFSBCB STORAGE ERROR	Request to get BCB storage for PDIR control block failed.
8A	WILDCARD PARAMETER NOT SUPPORTED	CREATE command does not support wildcard parameters. You must explicitly specify the names of the resources you want to create.
93	RESERVED NAME	Name is reserved. For example, name starts with DFS (except for DFSSAMxx or DFSIVPxx), or is an IMS reserved name such as BASICEDT or WTOR.

Table 155. Completion codes for the CREATE PGM command (continued)

Completion code	Completion code text	Meaning
BA	NOT ALLOWED FOR MRQ PSB	CREATE command failed for MRQ PSB because an attribute was specified that is not supported for the MRQ PSB: BMPTYPE(N), DOPT(Y), GPSB(Y), FP(E), RESIDENT(Y), SCHDTYPE(SERIAL)
B8	NO PSB IN ACBLIB	The DFSINSX0 exit attempted specified that the create should fail if no PSB for the program exists in ACBLIB.

## Examples

The following are examples of the CREATE PGM command:

### Example 1 for CREATE PGM command

TSO SPOC input:

```
CREATE PGM NAME(PGM00001) SET(BMPTYPE(Y))
```

TSO SPOC output:

```
PgmName MbrName CC
PGM00001 IMS1 0
```

OM API input:

```
CMD(CREATE PGM NAME(PGM00001) SET(BMPTYPE(Y)))
```

OM API output:

```
<imsout>
<ctl>
<omname>OM10M </omname>
<omvsn>1.3.0</omvsn>
<xmlvsn>20 </xmlvsn>
<statime>2006.254 22:20:06.531446</statime>
<stotime>2006.254 22:20:06.542510</stotime>
<staseq>BF6453E8EAD76197</staseq>
<stoseq>BF6453E8ED8AE98C</stoseq>
<rqsttkn1>USRT011 10152006</rqsttkn1>
<rc>00000000</rc>
<rsn>00000000</rsn>
</ctl>
<cmd>
<master>IMS1 </master>
<userid>USRT011 </userid>
<verb>CRE </verb>
<kwd>PGM </kwd>
<input>CREATE PGM NAME(PGM00001) SET(BMPTYPE(Y)) </input>
</cmd>
<cmdrsphdr>
<hdr s1b1="PGM" l1b1="PgmName" scope="LCL" sort="a" key="1" scroll="no"
len="8" dtype="CHAR" align="left" />
<hdr s1b1="MBR" l1b1="MbrName" scope="LCL" sort="a" key="2" scroll="no"
len="8" dtype="CHAR" align="left" />
<hdr s1b1="CC" l1b1="CC" scope="LCL" sort="n" key="0" scroll="yes"
len="4" dtype="INT" align="right" skipb="no" />
<hdr s1b1="CCTXT" l1b1="CCText" scope="LCL" sort="n" key="0"
```

```

    scroll="yes" len="*" dtype="CHAR" skipb="yes" align="left" />
</cmdrsphdr>
<cmdrspdata>
<rsp>PGM(PGM00001) MBR(IMS1) CC(  0) </rsp>
</cmdrspdata>
</imsout>

```

TSO SPOC input:

```
QRY PGM NAME(PGM00001) SHOW(MODEL,BMPTYPE,DEFNTYPE)
```

TSO SPOC output:

PgmName	MbrName	CC	RgnType	BMPTYPE	ModelName	ModelType	DefnType
PGM00001	IMS1	0	BMP	Y	DFSDSPG1	DESC	CREATE

**Explanation:** Create a batch program called PGM00001. Attributes not specified on the command are assigned the defaults from the program default descriptor. The QRY PGM command shows the model name DFSDSPG1 and model type DESC for the program default descriptor.

### *Example 2 for CREATE PGM command*

TSO SPOC input:

```
CREATE PGM NAME(DFSSAM02,PLAPJK02,PGM00002,WTOR,PGM*)
```

TSO SPOC output:

PgmName	MbrName	CC	CCText
DFSSAM02	IMS1	11	RESOURCE ALREADY EXISTS
PGM*	IMS1	8A	WILDCARD PARAMETER NOT SUPPORTED
PGM00002	IMS1	0	
PLAPJK02	IMS1	11	RESOURCE ALREADY EXISTS
WTOR	IMS1	93	RESERVED NAME

OM API input:

```
CMD(CREATE PGM NAME(DFSSAM02,PLAPJK02,PGM00002,WTOR,PGM*))
```

OM API output:

```

<imsout>
<ctl>
<omname>OM10M </omname>
<omvsn>1.3.0</omvsn>
<xmlvsn>20 </xmlvsn>
<statime>2006.254 23:05:22.692697</statime>
<stotime>2006.254 23:05:22.738636</stotime>
<staseq>BF645E0740259287</staseq>
<stoseq>BF645E074B5CC902</stoseq>
<rqsttkn1>USRT011 10160522</rqsttkn1>
<rc>0200000C</rc>
<rsn>00003008</rsn>
<rsnmsg>CSLN054I</rsnmsg>
<rsntxt>None of the clients were successful.</rsntxt>
</ctl>
<cmderr>
<mbr name="IMS1 ">
<typ>IMS </typ>
<styp>DBDC </styp>
<rc>0000000C</rc>
<rsn>00003000</rsn>
<rsntxt>At least one rqst successful</rsntxt>
</mbr>
</cmderr>
<cmd>

```

```

<master>IMS1 </master>
<userid>USRT011 </userid>
<verb>CRE </verb>
<kwd>PGM </kwd>
<input>CREATE PGM NAME(DFSSAM02,PLAPJK02,PGM00002,WTOR,PGM*) </input>
</cmd>
<cmdrsphdr>
<hdr s1bl="PGM" l1bl="PgmName" scope="LCL" sort="a" key="1" scroll="no"
len="8" dtype="CHAR" align="left" />
<hdr s1bl="MBR" l1bl="MbrName" scope="LCL" sort="a" key="2" scroll="no"
len="8" dtype="CHAR" align="left" />
<hdr s1bl="CC" l1bl="CC" scope="LCL" sort="n" key="0" scroll="yes"
len="4" dtype="INT" align="right" skipb="no" />
<hdr s1bl="CCTXT" l1bl="CCText" scope="LCL" sort="n" key="0"
scroll="yes" len="*" dtype="CHAR" skipb="yes" align="left" />
</cmdrsphdr>
<cmdrspdata>
<rsp>PGM(DFSSAM02) MBR(IMS1) CC( 11) CCTXT(RESOURCE ALREADY EXISTS)
</rsp>
<rsp>PGM(PLAPJK02) MBR(IMS1) CC( 11) CCTXT(RESOURCE ALREADY EXISTS)
</rsp>
<rsp>PGM(PGM00002) MBR(IMS1) CC( 0) </rsp>
<rsp>PGM(WTOR ) MBR(IMS1) CC( 93) CCTXT(RESERVED NAME) </rsp>
<rsp>PGM(PGM* ) MBR(IMS1) CC( 8A) CCTXT(WILDCARD PARAMETER NOT
SUPPORTED) </rsp>
</cmdrspdata>
</imsout>

```

**Explanation:** Several programs fail to be created because of different error conditions. The CREATE commands that fail return a nonzero completion code (CC) and completion code text (CCText) that briefly describes the completion code. The CREATE command fails for programs DFSSAM02 and PLAPJK02, because they already exist. The CREATE command succeeds for program PGM00002. The CREATE command fails for program WTOR, because WTOR is a reserved name that cannot be used as a resource name. The CREATE command fails for program PGM\*, because an asterisk is an invalid character that cannot be used in a resource name.

**Related concepts:**

 [How to interpret CSL request return and reason codes \(System Programming APIs\)](#)

**Related reference:**

Chapter 2, “Command keywords and their synonyms,” on page 63

## CREATE PGMDESC command

Use the CREATE PGMDESC command to create program descriptors. A descriptor is a model that can be used to create descriptors or resources.

A descriptor is created with a value set for every attribute. Attributes not explicitly specified on the CREATE command take the default value. Any program resource or descriptor can be created using this descriptor as a model, by specifying the CREATE command with LIKE(DESC(*descriptor\_name*)).

**Subsections:**

- “Environment” on page 272
- “Syntax” on page 272
- “Keywords” on page 273
- “Usage notes” on page 277

- “Output fields” on page 277
- “Return, reason, and completion codes” on page 277
- “Examples” on page 280

## Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

Table 156. Valid environments for the CREATE PGMDESC command and keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
CREATE PGMDESC	X	X	X
LIKE	X	X	X
NAME	X	X	X
SET	X	X	X

## Syntax



Default values shown in this syntax diagram reflect the default values that are defined in the IMS program descriptor, DFSDSPG1. If you define another descriptor as the default with a CREATE or UPDATE PGMDESC command, the default values shown in this syntax diagram might not match the value defined in the current default descriptor.

## Keywords

The following keywords are valid for the CREATE PGMDESC command:

### LIKE

Specifies that the descriptor is created using an existing resource or descriptor as a model, instead of the default descriptor. The default descriptor is either the IMS descriptor DFSDSPG1, or user-defined. The model type can either be a descriptor (DESC), or a resource (RSC). The descriptor is created with all the same attributes as the model, except for the DEFAULT(Y) value. Attributes set explicitly by the CREATE command override the model attributes. DEFAULT(Y) must be specified explicitly to make a descriptor the default descriptor. Later changes to the model are not propagated to resources or descriptors that were created from it.

The QUERY commands can be used to display the model name and model type used to create the descriptor or resource. The model name and type returned from the QUERY command are provided for informational purposes only. The EXPORT command exports a resource definition including the model name and model type to the definition member. The IMPORT command imports a resource definition including the model name and model type from the definition member. An IMPORT command that creates a resource saves the model name and model type, but does not use the models attributes.

### DESC(*descriptor\_name*)

Specifies the name of the descriptor to use as a model to define this descriptor.

### RSC(*resource\_name*)

Specifies the name of the resource to use as a model to define this descriptor.

### NAME

Specifies the 1-8 character name of the descriptor. Names must be alphanumeric (A through Z, 0 through 9, #, \$, and @). All program names must begin with an alphabetic character (A through Z, #, \$, and @) followed by 0 - 7 alphanumeric characters. Names cannot include a blank, comma, period, hyphen, or equal sign. Names cannot begin with DFS, except for DFSIVPxx and DFSSAMxx. Names cannot be BASICEDT, DBCDM, DBF#FPU0, DBRC, ISCEDT, MSDB, SDB, or WTOR. Each local program name must be unique. If the descriptor is defined with BMPTYPE(N) and FP(N), the program name and associated PSB name must be the same. A program resource and a program descriptor can have the same name.

### SET

Specifies the attributes of the descriptor to be created. If the LIKE keyword is omitted, attributes not specified take the value defined in the default descriptor, which is either IMS descriptor DFSDSPG1 or user-defined. If the LIKE keyword is specified, attributes not specified take the value defined in the model.

### **BMPTYPE()**

BMP type option. Specifies whether the program executes in a BMP type region or not. A BMP type region might be a BMP region or a JBP region.

- N** The program does not execute in a BMP type region. It can execute in an IMS TM MPP, JMP, or IFP region or it can use either the ODBA or DRA interface. This specification should be used for programs running in IMS TM MPP, JMP, and IFP regions, or PSBs scheduled by CICS programs using DBCTL and other users of the DRA interface. This is the default.
- Y** The program executes in a BMP type region. It can execute in an IMS BMP region or a JBP region. Any associated transactions are assigned normal and limit priority values of zero.

### **DEFAULT**

Specifies whether the descriptor is the default.

- N** The descriptor is not the default.
- Y** The descriptor is the default, which resets the existing default descriptor to DEFAULT(N). When a descriptor or resource is created without the LIKE keyword, any attribute not specified on the CREATE command takes the value defined in the default descriptor. Only one descriptor can be defined as the default for a resource type. IMS defines a default program descriptor called DFSDSPG1, where all attributes are defined with the default value. Defining a user-defined descriptor as the default overrides the IMS-defined descriptor. Since only one program descriptor can be the default at one time, only one program name can be specified with DEFAULT(Y).

### **DOPT**

Specifies the dynamic option.

- N** The PSB associated with this program is not located dynamically. The control blocks for the PSB must exist in an ACBLIB or the IMS catalog, otherwise the program is set to a NOTINIT status, and cannot be scheduled. This is the default.
- Y** The PSB associated with this program is located dynamically. Each time the program that is associated with this PSB is scheduled, the latest copy of the PSB is loaded. When the program terminates, the PSB is deleted from the PSB pool.

Until the PSB is required to process a transaction, the ACB for the PSB does not need to be in the ACBLIB or, if the IMS management of ACBs is enabled, in the IMS catalog. A new version of the PSB can be defined and added to the ACBLIB or the IMS catalog at any time. IMS loads the new version of the ACB the next time the PSB is scheduled.

A DOPT PSB cannot be scheduled if the control blocks for the DBDs that the PSB references are not in the ACBLIB or the IMS catalog.

Keyword combination rules include the following:

- DOPT(Y) and GPSB(Y) are mutually exclusive.
- DOPT(Y) and LANG(JAVA) is a valid combination.
- LANG is invalid with GPSB(N), except if DOPT(Y) and LANG(JAVA).
- LANG(JAVA), DOPT(Y), and GPSB(N) is a valid combination.
- RESIDENT(Y) and DOPT(Y) are mutually exclusive.

- SCHDTYPE(PARALLEL) and DOPT(Y) are mutually exclusive.

**FP** Specifies the Fast Path option.

**E** The program is a Fast Path-exclusive program. This implicitly defines a wait-for-input (WFI) program. Either a transaction or a routing code that can be assigned by the user Input Edit/Routing exit routine must be defined for the Fast Path-exclusive program, in order for this program to be usable.

Fast Path must be defined in the IMS system.

**N** The program is not a Fast Path-exclusive program. This is the default. When FP(N) is specified, any attempt to use Fast Path resources or commands will yield unpredictable results.

Keyword combination rules include the following:

- FP(E) requires that Fast Path be defined.
- LANG(JAVA) and FP(E) are mutually exclusive.
- BMPTYPE(N) and FP(E) are mutually exclusive.

### **GPSB**

Specifies the generated PSB option.

**N** The PSB associated with the program is not generated by IMS. The control blocks for the PSB must exist in the ACBLIB or, if the IMS management of ACBs is enabled, in the IMS catalog, otherwise the program is set to a NOTINIT status, and cannot be scheduled.

**Y** The PSB and application control block (ACB) that are associated with the program are generated by IMS. The scheduling process of all environments generates a PSB that contains an I/O PCB and an alternate modifiable PCB.

You do not need to generate the PSB and ACB or submit DDL to define the PSB. The ACB for the PSB is not loaded from either ACBLIB or, if the IMS management of ACBs is enabled, the IMS catalog. Thus, I/O to retrieve the ACB from either the ACBLIB or the IMS catalog is eliminated.

The generated PSB contains an I/O PCB named IOPCBbbb and a modifiable, alternate PCB named TPPCB1bb. With an alternate modifiable PCB, an application can use the CHNG call to change the output destination and send output to a destination other than the input destination.

Keyword combination rules include the following:

- DOPT(Y) and GPSB(Y) are mutually exclusive.
- DOPT(Y) and LANG(JAVA) is a valid combination.
- GPSB(Y) requires LANG.
- LANG is invalid with GPSB(N), except if DOPT(Y) and LANG(JAVA).
- LANG(JAVA), DOPT(Y), and GPSB(N) is a valid combination.
- RESIDENT(Y) and GPSB(Y) are mutually exclusive.

### **LANG**

Specifies the language interface of the program for a GPSB.

<b>LANG parameter</b>	<b>Meaning</b>
ASSEM	Assembler
COBOL	COBOL
JAVA	Java
PASCAL	Pascal
PLI	PL/I

Keyword combination rules include the following:

- LANG is invalid with GPSB(N), except if DOPT(Y) and LANG(JAVA).
- DOPT(Y) and LANG(JAVA) is a valid combination.
- LANG(JAVA), DOPT(Y), and GPSB(N) is a valid combination.
- LANG(JAVA) and FP(E) are mutually exclusive.

#### **RESIDENT**

Specifies the resident option.

- N** The PSB associated with the program is not made resident in storage. The PSB is loaded at scheduling time. This is the default.
- Y** The PSB associated with the program is made resident in storage immediately. IMS loads the PSB and initializes it. A resident PSB is accessed in local storage, which avoids I/O to retrieve the ACB. Resident PSBs are loaded by IMS restart, a CREATE PGM or an UPDATE PGM command that defines the PSB as resident.

Keyword combination rules include the following:

- DOPT(Y) and RESIDENT(Y) are mutually exclusive.
- GPSB(Y) and RESIDENT(Y) are mutually exclusive.

#### **SCHDTYPE**

Specifies whether this program can be scheduled into more than one message region or batch message region simultaneously.

##### **PARALLEL**

The program can be scheduled in multiple regions simultaneously. This is the default.

##### **SERIAL**

The program can only be scheduled in one region at a time.

Keyword combination rules include the following:

- DOPT(Y) and SCHDTYPE(PARALLEL) are mutually exclusive.

#### **TRANSTAT**

Specifies whether transaction level statistics should be logged. The value specified has meaning only if the descriptor is a JBP or a non-message driven BMP. If Y is specified, transaction level statistics are written to the log in a X'56FA' log record.

- N** Transaction level statistics should not be logged.
- Y** Transaction level statistics should be logged.

The TRANSTAT keyword is optional. If a value is not specified for the TRANSTAT keyword, the system default is used. The system default for the transaction level statistics parameter is set with the TRANSTAT parameter in

the DFSDFxxx PROCLIB member. If a value is not specified for the TRANSTAT parameter, the system default is set to N. The TRANSTAT keyword on the CREATE PGM or CREATE PGMDESC command gives the user the ability to override the system default when creating a program or program descriptor.

## Usage notes

Descriptors exist for the life of the IMS unless they are deleted using a DELETE command. The resources and descriptors are recoverable across an IMS warm start or emergency restart. Resources and descriptors are lost if IMS is cold started, unless cold start imports definitions that were exported while IMS was up.

The CREATE PGMDESC command can be issued only through the Operations Manager API. This command applies to DB/DC, DBCTL and DCCTL systems.

This command is not valid on the XRF alternate, RSR tracker, or FDBR region. The CREATE command is not valid if online change for MODBLKS is enabled (DFSDFxxx or DFSCGxxx defined with MODBLKS=OLC or MODBLKS not defined).

## Output fields

The following table shows the CREATE PGMDESC output fields. The columns in the table are as follows:

### Short label

Contains the short label generated in the XML output.

### Keyword

Identifies keyword on the command that caused the field to be generated. N/A appears for output fields that are always returned. *error* appears for output fields that are returned only in case of an error.

### Meaning

Provides a brief description of the output field.

Table 157. Output fields for the CREATE PGMDESC command

Short label	Keyword	Meaning
CC	N/A	Completion code.
CCTXT	<i>error</i>	Completion code text that briefly explains the meaning of the nonzero completion code.
DESC	PGMDESC	Program descriptor name.
MBR	N/A	IMSplex member that built the output line.
OLDDEF	PGMDESC	Old default descriptor name, if this descriptor is made the default by specifying DEFAULT(Y). The old default descriptor is no longer the default.

## Return, reason, and completion codes

The following table includes the return and reason codes and a brief explanation of the codes. The return or reason code returned for the command might also indicate an error from a CSL request.

Table 158. Return and reason codes for CREATE PGMDESC

Return code	Reason code	Meaning
X'00000000'	X'00000000'	Command completed successfully. The command output contains a line for each descriptor, accompanied by its completion code. See the completion code table for details.
X'00000008'	X'00002048'	Invalid SET attribute.
X'00000008'	X'00002106'	DOPT(Y) mutually exclusive with RESIDENT(Y). Either both keywords were explicitly specified, or one keyword was explicitly specified and the other attribute was obtained from the model.
X'00000008'	X'00002107'	DOPT(Y) mutually exclusive with SCHDTYPE(PARALLEL). Either both keywords were explicitly specified, or one keyword was explicitly specified and the other attribute was obtained from the model.
X'00000008'	X'0000210B'	FP(E) mutually exclusive with BMPTYPE(Y). Either both keywords were explicitly specified, or one keyword was explicitly specified and the other attribute was obtained from the model.
X'00000008'	X'0000210D'	FP(E) mutually exclusive with LANG(JAVA). Either both keywords were explicitly specified, or one keyword was explicitly specified and the other attribute was obtained from the model.
X'00000008'	X'00002113'	GPSB(Y) mutually exclusive with DOPT(Y). Either both keywords were explicitly specified, or one keyword was explicitly specified and the other attribute was obtained from the model.
X'00000008'	X'00002114'	GPSB(N) mutually exclusive with LANG.
X'00000008'	X'00002115'	GPSB(Y) mutually exclusive with RESIDENT(Y). Either both keywords were explicitly specified, or one keyword was explicitly specified and the other attribute was obtained from the model.
X'00000008'	X'00002117'	Invalid LIKE descriptor name.
X'00000008'	X'00002118'	Invalid LIKE resource name.
X'00000008'	X'00002131'	GPSB(Y) requires LANG to be specified.
X'00000008'	X'00002132'	DOPT(Y) not supported with LANG except for LANG(JAVA). Either both keywords were explicitly specified, or one keyword was explicitly specified and the other attribute was obtained from the model.
X'00000008'	X'00002133'	Multiple name parameters were specified with DEFAULT(Y). Only one descriptor can be the default at one time.
X'0000000C'	X'00003000'	Command was successful for some descriptors but failed for others. The command output contains a line for each descriptor, accompanied by its completion code. See the completion code table for details.
X'0000000C'	X'00003004'	Command was not successful for any of the descriptors. The command output contains a line for each descriptor, accompanied by its completion code. See the completion code table for details.

Table 158. Return and reason codes for CREATE PGMDESC (continued)

Return code	Reason code	Meaning
X'00000010'	X'0000400C'	Command is not valid on the XRF alternate.
X'00000010'	X'00004014'	Command is not valid on the RSR tracker.
X'00000010'	X'00004024'	No FP defined, so FP(E) invalid.
X'00000010'	X'00004120'	Online change phase is in progress.
X'00000010'	X'00004204'	Model is quiesced. Cannot quiesce model.
X'00000010'	X'00004300'	Command is not allowed because online change for MODBLKS is enabled (DFSDFxxx or DFSCGxxx defined with MODBLKS=OLC, or MODBLKS not defined).
X'00000014'	X'00005004'	DFSOCMD response buffer could not be obtained.
X'00000014'	X'00005008'	DFSPOOL storage could not be obtained.
X'00000014'	X'0000500C'	AWE could not be obtained.
X'00000014'	X'00005010'	Unable to obtain latch.
X'00000014'	X'00005114'	Resource or descriptor that is specified as a model is in the process of being imported from the change list in the IMSRSC repository or was not imported successfully from the change list at the end of IMS restart. This resource or descriptor cannot be referenced as a model on a CREATE command until it is successfully imported from the repository.

Errors unique to the processing of this command are returned as completion codes. The following table includes an explanation of the completion codes.

Table 159. Completion codes for the CREATE PGMDESC command

Completion code	Completion code text	Meaning
0		Command completed successfully for program descriptor.
11	RESOURCE ALREADY EXISTS	Program descriptor already exists.
5F	INVALID CHARACTERS IN NAME	Name is invalid because it contains an invalid character.
61	DFSBCB STORAGE ERROR	Request to get BCB storage for PDIR control block failed.
8A	WILDCARD PARAMETER NOT SUPPORTED	The CREATE command does not support wildcard parameters. You must explicitly specify the names of the descriptors you want to create.
93	RESERVED NAME	Name is reserved. For example, name starts with DFS (except for DFSSAMxx or DFSIVPxx), or is an IMS reserved name such as BASICEDT or WTOR.

## Examples

The following are examples of the CREATE PGMDESC command:

### *Example 1 for CREATE PGMDESC command*

TSO SPOC input:

```
CREATE PGMDESC NAME(FPEDESC) SET(DOPT(Y) FP(E) BMPTYPE(N),SCHDTYPE(SERIAL))
```

TSO SPOC output:

```
DescName MbrName CC  
FPEDESC IMS1 0
```

OM API input:

```
CMD(CREATE PGMDESC NAME(FPEDESC) SET(DOPT(Y) FP(E) BMPTYPE(N),SCHDTYPE(SERIAL)))
```

OM API output:

```
<imsout>  
<ctl>  
<omname>OM10M </omname>  
<omvsn>1.3.0</omvsn>  
<xmlvsn>20 </xmlvsn>  
<statime>2006.254 22:51:06.632728</statime>  
<stotime>2006.254 22:51:06.633257</stotime>  
<staseq>BF645AD6D921890E</staseq>  
<stoseq>BF645AD6D9429A8E</stoseq>  
<rqsttkn1>USRT011 10155106</rqsttkn1>  
<rc>00000000</rc>  
<rsn>00000000</rsn>  
</ctl>  
<cmd>  
<master>IMS1 </master>  
<userid>USRT011 </userid>  
<verb>CRE </verb>  
<kwd>PGMDESC </kwd>  
<input>CREATE PGMDESC NAME(FPEDESC) SET(DOPT(Y) FP(E)  
BMPTYPE(N),SCHDTYPE(SERIAL)) </input>  
</cmd>  
<cmdrspHdr>  
<hdr s1b1="DESC" l1b1="DescName" scope="LCL" sort="a" key="1"  
scroll="no" len="8" dtype="CHAR" align="left" />  
<hdr s1b1="MBR" l1b1="MbrName" scope="LCL" sort="a" key="2" scroll="no"  
len="8" dtype="CHAR" align="left" />  
<hdr s1b1="CC" l1b1="CC" scope="LCL" sort="n" key="0" scroll="yes"  
len="4" dtype="INT" align="right" skipb="no" />  
<hdr s1b1="CCTXT" l1b1="CCText" scope="LCL" sort="n" key="0"  
scroll="yes" len="*" dtype="CHAR" skipb="yes" align="left" />  
</cmdrspHdr>  
<cmdrspdata>  
<rsp>DESC(FPEDESC ) MBR(IMS1) CC( 0) </rsp>  
</cmdrspdata>  
</imsout>
```

**Explanation:** Creates a descriptor with attributes appropriate for a Fast Path exclusive program. Subsequent Fast Path exclusive programs can be created more easily by using this descriptor.

**Related concepts:**

 [How to interpret CSL request return and reason codes \(System Programming APIs\)](#)

**Related reference:**

## CREATE RTC command

Use the CREATE RTC command to create a Fast Path routing code that can be used by the Fast Path Input Edit/Routing Exit Routine (DBFHAGU0) to route a transaction to a different application program within the same load balancing group for processing.

Subsections:

- "Environment"
- "Syntax"
- "Keywords" on page 282
- "Usage notes" on page 283
- "Output fields" on page 283
- "Return, reason, and completion codes" on page 284
- "Examples" on page 285

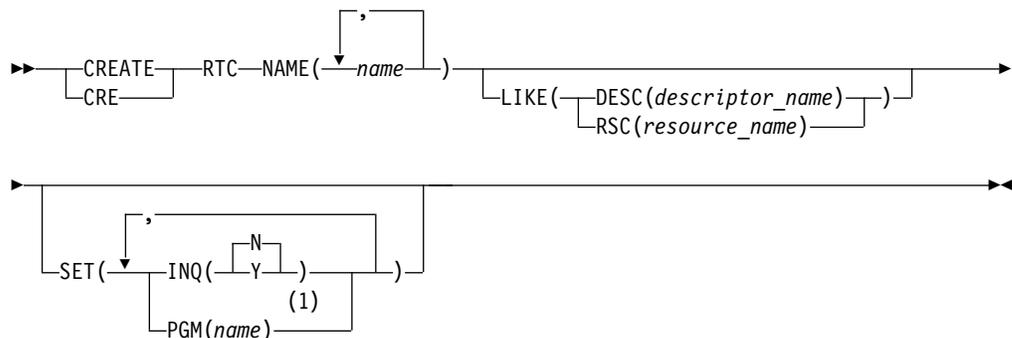
### Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

Table 160. Valid environments for the CREATE RTC command and keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
CREATE RTC	X		X
LIKE	X		X
NAME	X		X
SET	X		X

### Syntax



### Notes:

- 1 The PGM() keyword is required if the default descriptor or other descriptor from which the resource is being created does not already have a program defined.

Default values shown in this syntax diagram reflect the default values that are defined in the IMS routing code descriptor, DBFDSRT1. If you define another descriptor as the default with a CREATE or UPDATE RTCDESC command, the default values shown in this syntax diagram might not match the value defined in the current default descriptor.

## Keywords

The following keywords are valid for the CREATE RTC command:

### LIKE

Specifies that the resource is created using an existing resource or descriptor as a model, instead of the default descriptor. The default descriptor is either the IMS descriptor DBFDSRT1, or user-defined. The model type can either be a descriptor (DESC), or a resource (RSC). The new resource is created with all the same attributes as the model, except for the DEFAULT(Y) value. Attributes set explicitly by the CREATE command override the model attributes. Later changes to the model are not propagated to resources or descriptors that were created from it.

QUERY commands can be used to display the model name and model type used to create the resource. The model name and type returned from the QUERY command are provided for informational purposes only. The EXPORT command exports a resource definition, including the model name and model type, to the definition member. The IMPORT command imports a resource definition, including the model name and model type, from the definition member. An IMPORT command that creates a resource saves the model name and model type, but does not use the models attributes.

When LIKE is used on a CREATE command, the models current attributes are used to set the attributes of the newly created resource.

#### **DESC(descriptor\_name)**

Specifies the name of the descriptor to use as a model to define this resource.

#### **RSC(resource\_name)**

Specifies the name of the resource to use as a model to define this resource.

### NAME

Specifies the one- to eight-character alphanumeric routing code. Names must be alphanumeric (A through Z, 0 through 9, #, \$, and @). Names cannot include a blank, comma, period, hyphen, or equal sign. Names cannot begin with DFS, except for DFSIVPxx and DFSSAMxx. Names cannot be BASICEDT, DBCDM, DBFDSRT1, DBRC, ISCEDT, MSDB, SDB, or WTOR. Routing codes can be duplicates of either transaction names or logical terminal names, but each must be unique within the set of routing codes. A set of routing codes is the group of all routing codes associated with the same program. A routing code resource and a routing code descriptor can have the same name.

### SET

Specifies the attributes of the routing code to be created. If the LIKE keyword is omitted, attributes not specified take the value defined in the default descriptor, which is either IMS descriptor DBFDSRT1 or user-defined. If the LIKE keyword is specified, attributes not specified take the value defined in the model.

### INQ

- N This is not an inquiry transaction. This is the default.
- Y This is an inquiry transaction. Any message associated with the routing code is an inquiry transaction. This should be specified only for transactions that do not cause a change to a database. Programs are prohibited from issuing Insert, Delete, or Replace calls to a database when processing a transaction defined as INQ(Y).

**PGM**

Specifies the name of the program associated with the routing code. For a CREATE RTC command, the program must exist and be defined with a BMPTYPE(N). The PGM() keyword is required if the default descriptor or other descriptor from which the resource is being created does not already have a program defined.

**Usage notes**

Resources exist for the life of the IMS unless they are deleted using a DELETE command. The resources are recoverable across an IMS warm start or emergency restart. Resources are lost if IMS is cold started, unless cold start imports definitions that were exported while IMS was up.

The CREATE RTC command is similar to local online change (using a /MODIFY command) or global online change (INITIATE OLC) for routing code resources in the MODBLKS data set, except that routing codes are added dynamically.

The CREATE RTC command can be issued only through the Operations Manager API. Fast Path must be installed on the system. This command applies to DB/DC and DCCTL systems.

This command is not valid on the XRF alternate, RSR tracker, or FDBR region. The CREATE command is not valid if online change for MODBLKS is enabled (DFSDFxxx or DFSCGxxx defined with MODBLKS=OLC or MODBLKS not defined).

**Output fields**

The following table shows the CREATE RTC output fields. The columns in the table are as follows:

**Short label**

Contains the short label generated in the XML output.

**Keyword**

Identifies keyword on the command that caused the field to be generated. N/A appears for output fields that are always returned. *error* appears for output fields that are returned only in case of an error.

**Meaning**

Provides a brief description of the output field.

*Table 161. Output fields for the CREATE RTC command*

Short label	Keyword	Meaning
CC	N/A	Completion code.
CCTXT	<i>error</i>	Completion code text that briefly explains the meaning of the non-zero completion code.
MBR	N/A	IMSplex member that built the output line.

Table 161. Output fields for the CREATE RTC command (continued)

Short label	Keyword	Meaning
RTC	RTC	Routing code name.

## Return, reason, and completion codes

The following table includes the return and reason codes and a brief explanation of the codes. The return or reason code returned for the command might also indicate an error from a CSL request.

Table 162. Return and reason codes for CREATE RTC

Return code	Reason code	Meaning
X'00000000'	X'00000000'	Command completed successfully. The command output contains a line for each resource, accompanied by its completion code. See the completion code table for details.
X'00000008'	X'00002048'	Invalid SET attribute.
X'00000008'	X'00002117'	Invalid LIKE descriptor name.
X'00000008'	X'00002118'	Invalid LIKE resource name.
X'00000008'	X'00002120'	No PGM() defined.
X'00000008'	X'00002123'	Invalid PGM name. PGM was specified explicitly, or gotten from the model. If the IMS-defined descriptor DBFDSRT1 was used, the command failed because DBFDSRT1 defines no program name.
X'00000008'	X'00002133'	Multiple name parameters were specified with DEFAULT(Y). Only one descriptor can be the default at one time.
X'0000000C'	X'00003000'	Command was successful for some resources but failed for others. The command output contains a line for each resource, accompanied by its completion code. See the completion code table for details.
X'0000000C'	X'00003004'	Command was not successful for any of the resources. The command output contains a line for each resource, accompanied by its completion code. See the completion code table for details.
X'00000010'	X'0000400C'	Command is not valid on the XRF alternate.
X'00000010'	X'00004014'	Command is not valid on the RSR tracker.
X'00000010'	X'00004024'	No Fast Path defined.
X'00000010'	X'00004120'	Online change phase is in progress.
X'00000010'	X'00004204'	Model is quiesced. Cannot quiesce model.
X'00000010'	X'00004300'	Command is not allowed because online change for MODBLKS is enabled (DFSDFxxx or DFSCGxxx defined with MODBLKS=OLC, or MODBLKS not defined).
X'00000010'	X'0000431C'	Program is quiesced. Cannot quiesce program.
X'00000014'	X'00005004'	DFSOCMD response buffer could not be obtained.
X'00000014'	X'00005008'	DFSPPOOL storage could not be obtained.
X'00000014'	X'0000500C'	AWE could not be obtained.
X'00000014'	X'00005010'	Unable to obtain latch.

Table 162. Return and reason codes for CREATE RTC (continued)

Return code	Reason code	Meaning
X'00000014'	X'00005114'	Resource or descriptor that is specified as a model is in the process of being imported from the change list in the IMSRSC repository or was not imported successfully from the change list at the end of IMS restart. This resource or descriptor cannot be referenced as a model on a CREATE command until it is successfully imported from the repository.

Errors unique to the processing of this command are returned as completion codes. The following table includes an explanation of the completion codes.

Table 163. Completion codes for the CREATE RTC command

Completion code	Completion code text	Meaning
0		Command completed successfully for routing code.
11	RESOURCE ALREADY EXISTS	Routing code already exists.
17	ANOTHER CMD IN PROGRESS	Another command (such as DELETE PGM or UPDATE PGM) is in progress for the program referenced by this routing code.
5F	INVALID CHARACTERS IN NAME	Name is invalid because it contains an invalid character.
61	DFSBCB STORAGE ERROR	Request to get BCB storage for RCTE control block failed.
7A	RTC/FP=N PGM CONFLICT	Routing code to be created conflicts with program defined as non-Fast Path FP(N).
8A	WILDCARD PARAMETER NOT SUPPORTED	The CREATE command does not support wildcard parameters. You must explicitly specify the names of the resources you want to create.
90	INTERNAL ERROR	Internal error.
93	RESERVED NAME	Name is reserved. For example, name starts with DFS (except for DFSSAMxx or DFSIVPxx), or is an IMS reserved name such as BASICEDT or WTOR.

## Examples

The following are examples of the CREATE RTC command:

### Example 1 for CREATE RTC command

TSO SPOC input:

```
CRE RTC (NAME(RTC00001)) LIKE(RSC(TXCDRN14))
```

TSO SPOC output:

```
RtcName MbrName CC
RTC00001 IMS1 0
```

#### OM API input:

```
CMD(CRE RTC (NAME(RTC00001)) LIKE(RSC(TXCDRN14)))
```

#### OM API output:

```
<imsout>
<ctl>
<omname>OM10M </omname>
<omvsn>1.3.0</omvsn>
<xmlvsn>20 </xmlvsn>
<statime>2006.254 23:36:59.043090</statime>
<stotime>2006.254 23:36:59.043761</stotime>
<staseq>BF646517C0512E05</staseq>
<stoseq>BF646517C07B1705</stoseq>
<rqsttkn1>USRT011 10163658</rqsttkn1>
<rc>00000000</rc>
<rsn>00000000</rsn>
</ctl>
<cmd>
<master>IMS1 </master>
<userid>USRT011 </userid>
<verb>CRE </verb>
<kwd>RTC </kwd>
<input>CRE RTC (NAME(RTC00001)) LIKE(RSC(TXCDRN14)) </input>
</cmd>
<cmdrsphdr>
<hdr s1b1="RTC" l1b1="RtcName" scope="LCL" sort="a" key="1" scroll="no"
  len="8" dtype="CHAR" align="left" />
<hdr s1b1="MBR" l1b1="MbrName" scope="LCL" sort="a" key="2" scroll="no"
  len="8" dtype="CHAR" align="left" />
<hdr s1b1="CC" l1b1="CC" scope="LCL" sort="n" key="0" scroll="yes"
  len="4" dtype="INT" align="right" />
</cmdrsphdr>
<cmdrspdata>
<rsp>RTC(RTC00001) MBR(IMS1) CC( 0) </rsp>
</cmdrspdata>
</imsout>
```

#### TSO SPOC input:

```
QRY RTC (NAME(RTC0*)) SHOW(MODEL, INQ, PGM, DEFNTYPE)
```

#### TSO SPOC output:

```
Rtcode MbrName CC PgmName Inq ModelName ModelType DefnType
RTC00001 IMS1 0 DDLTRN14 N TXCDRN14 RSC CREATE
```

**Explanation:** A CREATE RTC command is issued to create routing code RTC00001 like an already existing routing code called TXCDRN14. The QRY RTC command shows the model name TXCDRN14, specified by the LIKE keyword, and model type RSC, for routing code TXCDRN14.

#### Related concepts:

 How to interpret CSL request return and reason codes (System Programming APIs)

#### Related reference:

Chapter 2, "Command keywords and their synonyms," on page 63

## CREATE RTCDESC command

Use the CREATE RTCDESC command to create Fast Path routing code descriptors.

A descriptor is a model that can be used to create routing code descriptors or resources. A descriptor is created with a value set for every attribute. Attributes not explicitly specified on the CREATE command take the default value. Any routing code resource or descriptor can be created using this descriptor as a model, by specifying the CREATE command with LIKE(DESC(descriptor\_name)).

Subsections:

- "Environment"
- "Syntax"
- "Keywords" on page 288
- "Usage notes" on page 289
- "Output fields" on page 289
- "Return, reason, and completion codes" on page 290
- "Examples" on page 292

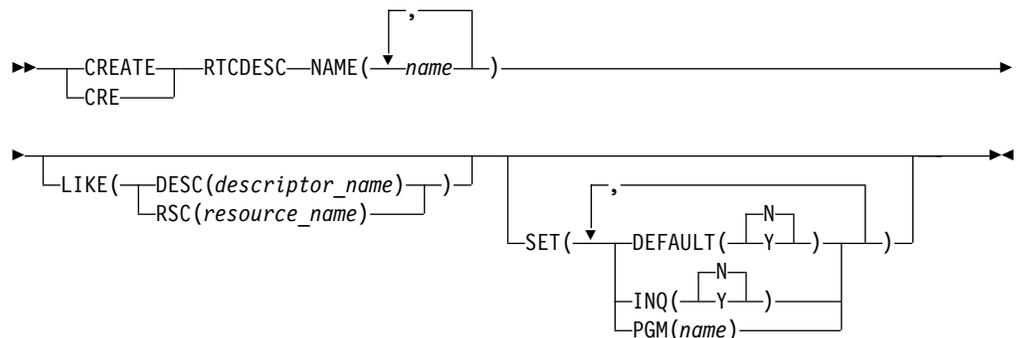
### Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

Table 164. Valid environments for the CREATE RTCDESC command and keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
CREATE RTCDESC	X		X
LIKE	X		X
NAME	X		X
SET	X		X

### Syntax



Default values shown in this syntax diagram reflect the default values that are defined in the IMS routing code descriptor, DBFDSRT1. If you define another descriptor as the default with a CREATE or UPDATE RTCDESC command, the

default values shown in this syntax diagram might not match the value defined in the current default descriptor.

## Keywords

The following keywords are valid for the CREATE RTCDESC command:

### LIKE

Specifies that the descriptor is created using an existing resource or descriptor as a model, instead of the default descriptor. The default descriptor is either the IMS descriptor DBFDSRT1, or user-defined. The model type can either be a descriptor (DESC), or a resource (RSC). The descriptor is created with all the same attributes as the model, except for the DEFAULT(Y) value. Attributes set explicitly by the CREATE command override the model attributes. DEFAULT(Y) must be specified explicitly to make a descriptor the default descriptor. Later changes to the model are not propagated to resources or descriptors that were created from it.

QUERY commands can be used to display the model name and model type used to create the descriptor or resource. The model name and type returned from the QUERY command are provided for informational purposes only. The EXPORT command exports a resource definition including the model name and model type to the definition member. The IMPORT command imports a resource definition including the model name and model type from the definition member. An IMPORT command that creates a resource saves the model name and model type, but does not use the model's attributes.

When LIKE is used on a CREATE command, the model's current attributes are used to set the attributes of the newly-created resource or descriptor, except for the default attribute.

### DESC(*descriptor\_name*)

Specifies the name of the descriptor to use as a model to define this descriptor.

### RSC(*resource\_name*)

Specifies the name of the resource to use as a model to define this descriptor.

### NAME

Specifies the one- to eight-character alphanumeric routing code descriptor. Names must be alphanumeric (A through Z, 0 through 9, #, \$, and @). Names cannot include a blank, comma, period, hyphen, or equal sign. Names cannot begin with DFS, except for DFSIVPxx and DFSSAMxx. Names cannot be BASICEDT, DBCDM, DBFDSRT1, DBRC, ISCEDT, MSDB, SDB, or WTOR. Routing codes can be duplicates of either transaction names or logical terminal names, but each must be unique within the set of routing codes. A set of routing codes is the group of all routing codes associated with the same program. A routing code resource and a routing code descriptor can have the same name.

### SET

Specifies the attributes of the routing code descriptor to be created. If the LIKE keyword is omitted, attributes not specified take the value defined in the default descriptor, which is either IMS descriptor DBFDSRT1 or user-defined. If the LIKE keyword is specified, attributes not specified take the value defined in the model.

**DEFAULT**

Specifies whether the descriptor is the default.

- N** The descriptor is not the default.
- Y** The descriptor is the default, which resets the existing default descriptor to DEFAULT(N). When a descriptor or resource is created without the LIKE keyword, any attribute not specified on the CREATE command takes the value defined in the default descriptor. Only one descriptor can be defined as the default for a resource type. IMS defines a default routing code descriptor called DBFDSRT1, where all attributes are defined with the default value. Defining a user-defined descriptor as the default overrides the IMS-defined descriptor. Since only one routing code descriptor can be the default at one time, only one routing code parameter may be specified with DEFAULT(Y).

**INQ**

- N** This is not an inquiry transaction. This is the default.
- Y** This is an inquiry transaction. Any message associated with the routing code descriptor is an inquiry transaction. This should be specified only for transactions that do not cause a change to a database. Programs are prohibited from issuing Insert, Delete, or Replace calls to a database when processing a transaction defined as INQ(Y).

**PGM**

Specifies the name of the program associated with the routing code descriptor.

**Usage notes**

Descriptors exist for the life of the IMS unless they are deleted using a DELETE command. The resources and descriptors are recoverable across an IMS warm start or emergency restart. Descriptors are lost if IMS is cold started, unless cold start imports definitions that were exported while IMS was up.

The CREATE RTCDESC command can only be issued through the Operations Manager API. Fast Path must be installed on the system. This command applies to DB/DC and DCCTL systems.

This command is not valid on the XRF alternate, RSR tracker, or FDBR region. The CREATE command is not valid if online change for MODBLKS is enabled (DFSDFxxx or DFSCGxxx defined with MODBLKS=OLC or MODBLKS not defined).

**Output fields**

The following table shows the CREATE RTCDESC output fields. The columns in the table are as follows:

**Short label**

Contains the short label generated in the XML output.

**Keyword**

Identifies keyword on the command that caused the field to be generated. N/A appears for output fields that are always returned. *error* appears for output fields that are returned only in case of an error.

## Meaning

Provides a brief description of the output field.

Table 165. Output fields for the CREATE RTCDESC command

Short label	Keyword	Meaning
CC	N/A	Completion code.
CCTXT	<i>error</i>	Completion code text that briefly explains the meaning of the non-zero completion code.
DESC	RTCDESC	Routing code descriptor name.
MBR	N/A	IMSplex member that built the output line.
OLDDEF	RTCDESC	Old default descriptor name, if this descriptor is made the default by specifying DEFAULT(Y). The old default descriptor is no longer the default.

## Return, reason, and completion codes

The following table includes the return and reason codes and a brief explanation of the codes. The return or reason code returned for the command might also indicate an error from a CSL request.

Table 166. Return and reason codes for CREATE RTCDESC

Return code	Reason code	Meaning
X'00000000'	X'00000000'	Command completed successfully. The command output contains a line for each descriptor, accompanied by its completion code. See the completion code table for details.
X'00000008'	X'00002048'	Invalid SET attribute.
X'00000008'	X'00002117'	Invalid LIKE descriptor name.
X'00000008'	X'00002118'	Invalid LIKE resource name.
X'00000008'	X'00002123'	Invalid PGM name. PGM was specified explicitly, or gotten from the model. If the IMS-defined descriptor DBFDSRT1 was used, the command failed because DBFDSRT1 defines no program name.
X'00000008'	X'00002137'	Multiple name parameters specified with DEFAULT(Y) is invalid. Only one descriptor can be the default at one time.
X'0000000C'	X'00003000'	Command was successful for some descriptors but failed for others. The command output contains a line for each descriptor, accompanied by its completion code. See the completion code table for details.
X'0000000C'	X'00003004'	Command was not successful for any of the descriptors. The command output contains a line for each descriptor, accompanied by its completion code. See the completion code table for details.
X'00000010'	X'0000400C'	Command is not valid on the XRF alternate.
X'00000010'	X'00004014'	Command is not valid on the RSR tracker.
X'00000010'	X'00004024'	No Fast Path defined.
X'00000010'	X'00004120'	Online change phase is in progress.
X'00000010'	X'00004204'	Model is quiesced. Cannot quiesce model.

Table 166. Return and reason codes for CREATE RTCDESC (continued)

Return code	Reason code	Meaning
X'00000010'	X'00004300'	Command is not allowed because online change for MODBLKS is enabled (DFSDFxxx or DFSCGxxx defined with MODBLKS=OLC, or MODBLKS not defined).
X'00000014'	X'00005004'	DFSOCMD response buffer could not be obtained.
X'00000014'	X'00005008'	DFSPOOL storage could not be obtained.
X'00000014'	X'0000500C'	AWE could not be obtained.
X'00000014'	X'00005010'	Unable to obtain latch.
X'00000014'	X'00005114'	Resource or descriptor that is specified as a model is in the process of being imported from the change list in the IMSRSC repository or was not imported successfully from the change list at the end of IMS restart. This resource or descriptor cannot be referenced as a model on a CREATE command until it is successfully imported from the repository.

Errors unique to the processing of this command are returned as completion codes. The following table includes an explanation of the completion codes.

Table 167. Completion codes for the CREATE RTCDESC command

Completion code	Completion code text	Meaning
0		Command completed successfully for routing code descriptor.
11	RESOURCE ALREADY EXISTS	Routing code descriptor already exists.
17	ANOTHER CMD IN PROGRESS	Another command (such as DELETE PGM or UPDATE PGM) is in progress for the program referenced by this routing code descriptor.
5F	INVALID CHARACTERS IN NAME	Name is invalid because it contains an invalid character.
61	DFSBCB STORAGE ERROR	Request to get BCB storage for RCTE control block failed.
7A	RTC/FP=N PGM CONFLICT	Routing code to be created conflicts with program defined as non-Fast Path FP(N).
8A	WILDCARD PARAMETER NOT SUPPORTED	CREATE command does not support wildcard parameters. You must explicitly specify the names of the resources you want to create.
90	INTERNAL ERROR	Internal error
93	RESERVED NAME	Name is reserved. For example, name starts with DFS (except for DFSSAMxx or DFSIVPxx), or is an IMS reserved name such as BASICEDT or WTOR.

## Examples

The following are examples of the CREATE RTCDESC command:

### *Example 1 for CREATE RTCDESC command*

TSO SPOC input:

```
CRE RTCDESC NAME(DRCTR02,DRCTR03) LIKE(RSC(FPTRN01)) SET(PGM(FPPSB03))
```

TSO SPOC output:

DescName	MbrName	CC
DRCTR02	IMS1	0
DRCTR03	IMS1	0

**Explanation:** A CREATE RTCDESC command is issued to create two routing code descriptors at IMS1 in an IMSplex. Routing code descriptors DRCTR02 and DRCTR03 are defined using the existing routing code resource called FPTRN01. Routing code FPTRN01 is defined with INQ(Y). Descriptors DRCTR02 and DRCTR03 are defined using routing code FPTRN01 as a model for the INQ(Y) attribute, but overriding the program name by explicitly specifying PGM(FPPSB03) on the command.

### **Related concepts:**

 [How to interpret CSL request return and reason codes \(System Programming APIs\)](#)

### **Related reference:**

Chapter 2, “Command keywords and their synonyms,” on page 63

---

## CREATE TRAN command

Use the CREATE TRAN command to create a transaction code that causes the application program associated with the program resource defined by PGM() to be scheduled for execution in an IMS message processing region.

The transaction provides IMS with information that influences the application program scheduling algorithm. The program resource defined by PGM() keyword must exist for the CREATE TRAN command to complete successfully.

Subsections:

- “Environment”
- “Syntax” on page 293
- “Keywords” on page 295
- “Usage notes” on page 309
- “Output fields” on page 309
- “Return, reason, and completion codes” on page 310
- “Examples” on page 315

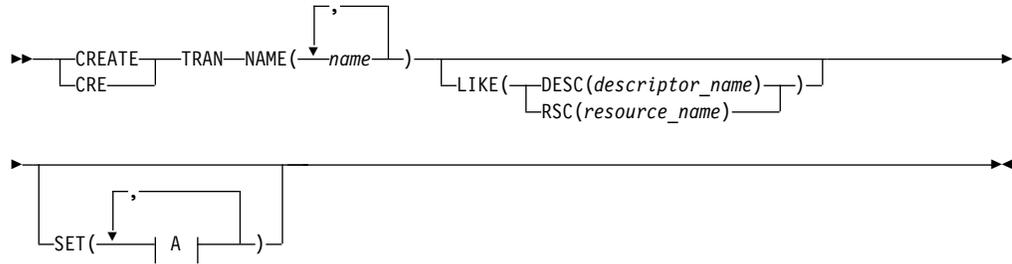
### **Environment**

The following table lists the environments (DB/BC, DBCTL, and DCCTL) from which the command keywords can be issued.

Table 168. Valid environments for the CREATE TRAN command and keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
CREATE TRAN	X		X
LIKE	X		X
NAME	X		X
SET	X		X

## Syntax



**A:**

AOCMD ( <input type="checkbox"/> N <input type="checkbox"/> CMD <input type="checkbox"/> TRAN <input type="checkbox"/> Y )
CLASS ( <input type="checkbox"/> 1 <input type="checkbox"/> class )
CMTMODE ( <input type="checkbox"/> SINGL <input type="checkbox"/> MULT )
CONV ( <input type="checkbox"/> N <input type="checkbox"/> Y )
DCLWA ( <input type="checkbox"/> N <input type="checkbox"/> Y )
DIRROUTE ( <input type="checkbox"/> N <input type="checkbox"/> Y )
EDITRTN ( <i>name</i> )
EDITUC ( <input type="checkbox"/> Y <input type="checkbox"/> N )
EMHBSZ ( <i>size</i> )
EXPRTIME ( <i>seconds</i> )
FP ( <input type="checkbox"/> N <input type="checkbox"/> E <input type="checkbox"/> P )
INQ ( <input type="checkbox"/> N <input type="checkbox"/> Y )
LCT ( <input type="checkbox"/> 65535 <input type="checkbox"/> value )
LPRI ( <input type="checkbox"/> 1 <input type="checkbox"/> value )
MAXRGN ( <input type="checkbox"/> 0 <input type="checkbox"/> number )
MSGTYPE ( <input type="checkbox"/> MULTSEG <input type="checkbox"/> SINGLSEG )
MSNAME ( <i>name</i> )
NPRI ( <input type="checkbox"/> 1 <input type="checkbox"/> value )
PARLIM ( <input type="checkbox"/> 65535 <input type="checkbox"/> value )
PGM ( <i>name</i> )
PLCT ( <input type="checkbox"/> 65535 <input type="checkbox"/> value )
PLCTTIME ( <input type="checkbox"/> 6553500 <input type="checkbox"/> hundredths of seconds )
RECOVER ( <input type="checkbox"/> Y <input type="checkbox"/> N )
REMOTE ( <input type="checkbox"/> N <input type="checkbox"/> Y )
RESP ( <input type="checkbox"/> N <input type="checkbox"/> Y )
SEGNO ( <input type="checkbox"/> 0 <input type="checkbox"/> number )
SEGSZ ( <input type="checkbox"/> 0 <input type="checkbox"/> size )
SERIAL ( <input type="checkbox"/> N <input type="checkbox"/> Y )
SIDL ( <i>localsysid</i> )
SIDR ( <i>remotesysid</i> )
SPASZ ( <i>size</i> )
SPATRUNC ( <input type="checkbox"/> R <input type="checkbox"/> S )
TRANSTAT ( <input type="checkbox"/> N <input type="checkbox"/> Y )
WFI ( <input type="checkbox"/> N <input type="checkbox"/> Y )

Default values shown in this syntax diagram reflect the default values that are defined in the IMS transaction descriptor, DFSDSTR1. If you define another descriptor as the default with a CREATE or UPDATE TRANDESC command, the default values shown in this syntax diagram might not match the value defined in the current default descriptor.

## Keywords

The following keywords are valid for the CREATE TRAN command:

### LIKE

Specifies that the resource is created using an existing resource or descriptor as a model, instead of the default descriptor. The default descriptor is either the IMS descriptor DFSDSTR1, or user-defined. The model type can either be a descriptor (DESC), or a resource (RSC). The new resource is created with all the same attributes as the model. Attributes set explicitly by the CREATE command override the model attributes. Later changes to the model are not propagated to resources or descriptors that were created from it.

The QUERY commands can be used to display the model name and model type used to create the resource. The model name and type returned from the QUERY command are provided for informational purposes only. The EXPORT command exports a resource definition, including the model name and model type, to the definition member. The IMPORT command imports a resource definition, including the model name and model type, from the definition member. An IMPORT command that creates a resource saves the model name and model type, but does not use the models attributes.

### DESC(*descriptor\_name*)

Specifies the name of the descriptor to use as a model to define this resource.

### RSC(*resource\_name*)

Specifies the name of the resource to use as a model to define this resource.

### NAME

Specifies the 1-8 character name of the transaction. Names must be alphanumeric (A through Z, 0 through 9, #, \$, and @). Names cannot include a blank, comma, period, hyphen, or equal sign. Names cannot begin with DFS, except for DFSCONE, DFSIVPxx, and DFSSAMxx. Names cannot be BASICEDT, DBCDM, DBRC, ISCEDT, MSDB, SDB, or WTOR. A transaction resource and a transaction descriptor can have the same name. A transaction cannot have the same name as an item.

### SET

Specifies the attributes of the transaction to be created. If the LIKE keyword is omitted, attributes not specified take the value defined in the default descriptor, which is either the IMS descriptor DFSDSTR1, or user-defined. If the LIKE keyword is specified, attributes not specified take the value defined in the model. A transaction must have the same characteristics in all systems where it is defined when it is shared. These characteristics include:

- Nonconversational/conversational
- SPA size if conversational
- Single-/multi-segment messages
- Noninquiry/inquiry

- Recoverable/nonrecoverable
- Response mode/non-response mode
- Fast Path exclusive/Fast Path potential/non-Fast Path

#### **AOCMD**

Specifies the AOI option that indicates whether the transaction can issue the type-1 AOI CMD call or the type-2 AOI ICMD call. If AOCMD is defined as CMD, TRAN, or Y, and the AOI1 execute parameter is defined as AOI1=N, no authorization checking is done, and the transaction is permitted to issue CMD and ICMD calls.

**N** Indicates that the transaction is not permitted to issue type-1 AOI CMD calls. The transaction is permitted to issue type-2 AOI ICMD calls.

#### **CMD**

Indicates that the transaction is permitted to issue type-1 AOI CMD calls and type-2 AOI ICMD calls. If the AOI1 execute parameter is defined as C, R, or A, authorization checking is based on which transactions can issue a particular command. In this case, the commands (or the first three characters of the commands) need to be defined to RACF or an equivalent product as a user. The type-1 AOI transactions must be defined as profiles under the TIMS class, and for each transaction, the commands it can issue must be specified. Defining AOCMD(CMD) requires you to create fewer user IDs than you need to create for the AOCMD(TRAN) definition. However, defining AOCMD(CMD) requires you to create or modify a larger number of resource profiles.

#### **TRAN**

Indicates that the transaction is permitted to issue type-1 AOI CMD calls and type-2 AOI ICMD calls. If the AOI1 execute parameter is defined as C, R, or A, the transaction code is used for authorization. The first authorization check results in the accessor environment element (ACEE) being built. This environment is kept for use by future authorization checks. The type-1 AOI transaction needs to be defined to RACF or an equivalent product as a user. The transactions will then be specified on RACF PERMIT statements for each command they are allowed to issue from a type-1 AOI transaction. Specifying AOI transactions as users to RACF might conflict with the name of a user already defined to RACF. If this occurs, then either the transaction name or the existing user name needs to be changed.

**Y** Indicates that the transaction is permitted to issue type-1 AOI CMD calls and type-2 AOI ICMD calls. If the AOI1 execute parameter is defined as C, R, or A, the user ID or the program name is used for authorization. For some environments, if a Get Unique call has not yet occurred, the program name is used for authorization.

#### **CLASS**

Specifies the transaction class, which is an attribute used to select a transaction for scheduling. A transaction can be scheduled if there is a message processing region available for that class. The value can be a number from 1 to 999. The default is 1. This value must not exceed the value given (by specification or default) on the MAXCLAS= keyword of the IMSCTRL macro.

Define CPI-C transactions with a different message class from that used for non-CPI-C transactions. IMS handles all CPI-C transactions as priority zero within the transaction class.

#### **CMTMODE**

Specifies when database updates and non-express output messages are committed. This operand affects emergency restart.

#### **MULT**

Database updates and non-express output messages are committed only when the application program terminates normally, when the processing limit count has been reached, or, in the case of a pseudo-WFI dependent region, when there are no more messages on the queue. For example, if five transactions are processed during a single schedule of a program, all five are committed only when the fifth one is completed and the program terminates. Until a transaction has been committed, locks for updated database records are not released and non-express output messages are not queued for output. If an application ends abnormally before committing its messages, emergency restart requeues all the messages that were processed within the commit scope and makes them available for reprocessing.

If the transaction results in the application calling an external subsystem, such as DB2, the Commit Verify exit provided by the external subsystem can determine whether CMTMODE(MULT) is supported. See documentation under the Commit Verify exit routine in *IMS Version 14 Exit Routines*.

#### **SNGL**

Database updates and non-express output messages are committed when the application program completes processing each transaction. IMS invokes commit processing either when the application program requests the next message (issues a GU to the IO-PCB), or when the application program terminates. If an application ends abnormally before committing its message, emergency restart requeues the message that was in process before the abend and makes it available for reprocessing. This is the default.

**Note:** The TRANSACT macro MODE keyword uses a default of MULT. The CREATE TRAN command CMTMODE keyword uses a default of SNGL.

Keyword combination rules include the following:

- CMTMODE(MULT) and CONV(Y) are mutually exclusive.
- CMTMODE(MULT) and WFI(Y) are mutually exclusive.

#### **CONV**

Specifies the conversation option.

**N** The transaction is not conversational. This is the default.

**Y** The transaction is conversational. The transaction message is destined for a conversational program. A conversational program processes transactions made up of several steps. A conversational program receives a message from a terminal, replies to the terminal, but saves the data from the transaction in a scratchpad area (SPA). When the person at the terminal enters more data, the program has the data it saved from the last message in the SPA, so it can continue processing the request without the person at the terminal having to enter the data again.

Keyword combination rules include the following:

- CMTMODE(MULT) and CONV(Y) are mutually exclusive.
- FP(E) and CONV(Y) are mutually exclusive.
- CONV(Y) requires SPASZ and SPATRUNC.

- RECOVER(N) and CONV(Y) are mutually exclusive.
- SPASZ and CONV(N) are mutually exclusive.
- SPATRUNC and CONV(N) are mutually exclusive.

#### **DCLWA**

Specifies the log write-ahead option. If not specified and the default descriptor is the IMS-defined descriptor DFSDSTR1, the value is defined to be the DFSDFxxx PROCLIB member DCLWA parameter at IMS cold start. Changing the DFSDFxxx DCLWA value across a restart has no effect on the default descriptor DCLWA value. If DCLWA is not defined in the DFSDFxxx PROCLIB member, the default is DCLWA=Y.

**N** IMS should not perform log write-ahead. Specify N if input message integrity and the consistency of output messages with associated database updates is not required. DCLWA does not apply to response mode or Fast Path input processing, and is ignored during IMS execution.

**Y**

IMS should perform log write-ahead for recoverable, nonresponse input messages and transaction output messages. This ensures the following:

- A nonresponse input transaction is made recoverable across IMS failures before IMS acknowledges receipt of the input.
- Database changes are made recoverable before IMS sends associated output reply messages.
- Information in the log buffers is written to the IMS log, before the associated input acknowledgment or output reply is sent to the terminal.

Define DCLWA(Y) for all VTAM terminal types.

#### **DIRROUTE**

Specifies the MSC directed routing option.

**N** The application program processing a transaction is not informed of the system which originated the transaction. The name of the originating LTERM is placed in the I/O PCB. This is the default.

**Y** The application program processing a transaction is informed of the system which originated the transaction, if MSC directed routing is used in a multiple IMS system configuration. An MSNAME corresponding to a logical path back to the originating system is placed in the I/O PCB.

#### **EDITRTN**

Specifies the 1- to 8-character name of your transaction input edit routine that edits messages before the program receives the message. This name must be alphanumeric (A through Z, 0 through 9, #, \$, and @). It must begin with an alphabetic character (A through Z, #, \$, @). It cannot include a blank, comma, period, hyphen, or equal. It cannot include the wildcard characters \* or %.

The specified edit routine can either be an edit routine defined during the system definition process with the EDIT= parameter on the TRANSACT macro or it can be a new routine. If the routine is a new routine, the routine must reside in one of the RESLIB concatenated data sets. A maximum of 255 input edit routines are supported.

If the edit routine specified on the command is new to IMS, IMS attempts to load the routine from RESLIB. If the load fails, the command is rejected. If the edit routine specified is already defined to IMS, a decision is made whether to load a new copy of the routine or to use the existing copy. If the routine was

generated into the system, but there are no transactions referencing the routine, IMS first attempts to load a new copy of the module from RESLIB. If the load is successful, the dynamic copy of the edit routine is used. The generated copy cannot be used again unless IMS is cold started. If the load of the dynamic routine fails, the generated copy is used. If the edit routine was generated into the system and other transactions reference it, the generated copy of the module is used.

EDITRTN is used for a Fast Path potential transaction when the transaction is routed to IMS.

For input from LU 6.2 devices, the user edit exit routine DFSLUEE0 is called instead of the transaction input edit routine specified in EDITRTN.

Keyword combination rules include the following:

- FP(E) and EDITRTN are mutually exclusive.

#### **EDITUC**

Specifies the edit to uppercase option.

**N** The input data is not translated to uppercase. It can consist of upper and lowercase characters as entered from the terminal.

**Y** The input data is to be translated to uppercase before it is presented to the processing program. If FP(E) or FP(P), the transaction is to be translated to uppercase before being presented to the edit/routing exit routine. This is the default.

Specifying EDITUC(Y) for VTAM terminals prevents the transmission of embedded device control characters.

#### **EMHBSZ**

Specifies the EMH buffer size required to run the Fast Path transaction. This overrides the EMHL execution parameter. If EMHBSZ is not specified, the EMHL execution parameter value is used. The value can be a number from 0 to 30 720.

Keyword combination rules include the following:

- EMHBSZ>0 requires Fast Path to be defined.
- FP(N) and EMHBSZ>0 are mutually exclusive.

#### **EXPTIME**

Specifies the elapsed time in seconds that IMS can use to cancel the input transaction. After a transaction is submitted to IMS, the transaction could be delayed for processing because of a stopped transaction or a potential system slow down. In that case, the user or client application could time out before the transaction is processed. When IMS eventually schedules and processes the transaction, the response message is no longer wanted. With the elapsed time specified for the transaction, IMS can flag the input transaction as expired so that the system does not waste CPU cycles to process the unwanted transaction.

The value can be a number, in seconds, which can range from 0 to 65535. The default is 0, which means that no expiration time is set for this transaction. The transaction expiration attribute is supported by all of the IMS TM interfaces.

**Restriction:** The transaction expiration checking is not performed at the GU time for Fast Path transactions, IMS conversational transactions, and program-to-program switch transactions.

**FP** Specifies the Fast Path option.

- E** The transaction is processed exclusively as Fast Path. A Fast Path routing code by the same name is created. The program must be defined as Fast Path exclusive.
- N** The transaction is not a candidate for Fast Path processing. The program identified by PGM() must be defined as not Fast Path. This is the default.
- P** The transaction is a potential candidate for Fast Path processing. Fast Path-potential transactions must be able to run under two programs: a Fast Path exclusive program a non-Fast Path program. This CREATE TRAN command defines the non-Fast Path program with PGM(). A Fast Path exclusive program should be defined to which this transaction can be routed. Fast Path-potential transactions must be processed by a user edit/routing exit to determine whether the transaction is actually to be processed by IMS Fast Path. If it is to be processed by IMS Fast Path, the edit/routing exit routine associates the transaction with a routing code. This routing code identifies which Fast Path program is to process the transaction.

The program defined by PGM() must not be defined as Fast Path exclusive.

Keyword combination rules include the following:

- EDITRTN and FP(E) are mutually exclusive.
- FP(P) and FP(E) require Fast Path to be defined.
- CONV(Y) and FP(E) are mutually exclusive.
- MSGTYPE(MULTSEG) and FP(E) are mutually exclusive.
- MSGTYPE(MULTSEG) and FP(P) are mutually exclusive.
- MSNAME and FP(E) are mutually exclusive.
- MSNAME and FP(P) are mutually exclusive.
- RECOVER(N) and FP(E) are mutually exclusive.
- RECOVER(N) and FP(P) are mutually exclusive.
- RESP(N) and FP(E) are mutually exclusive.
- RESP(N) and FP(P) are mutually exclusive.
- SIDL and FP(E) are mutually exclusive.
- SIDL and FP(P) are mutually exclusive.
- SIDR and FP(E) are mutually exclusive.
- SIDR and FP(P) are mutually exclusive.

#### **INQ**

Specifies the inquiry option.

- N** This is not an inquiry transaction. This is the default.
- Y** This is an inquiry transaction. If INQ(Y) is specified, you can also specify whether this transaction should be recovered during an IMS emergency or normal restart using the RECOVER() parameter.

This option should be specified only for those transactions that, when entered, do not cause a change in any database. Programs are prohibited from issuing ISRT, DLET, or REPL calls to a database when scheduled to process a transaction defined as INQ(Y).

An application program cannot do an SQL INSERT, DELETE, or UPDATE when the IMS transaction is defined with INQ(Y).

Keyword combination rules include the following:

- RECOVER(N) and INQ(N) are mutually exclusive.

### **LCT**

Specifies the limit count. This is the number that, when compared to the number of input transactions queued and waiting to be processed, determines whether the normal or limit priority value is assigned to this transaction. The value can be a number from 1 to 65535. The default is 65535.

The limit count value is ignored for a transaction processed by a BMP.

The limit count value is ignored in a shared-queues environment.

The limit count value does not apply to FP exclusive transactions and is ignored.

### **LPRI**

Specifies the limit priority. This is the scheduling priority to which this transaction is raised when the number of input transactions enqueued and waiting to be processed is equal to or greater than the limit count value. The scheduling priority is an attribute used to select a transaction for scheduling. A transaction of higher priority is scheduled before a lower priority one, if they are defined with the same class. The value can be a number from 0 through 14. The default is 1.

When the limit priority is used and the scheduling priority is raised to the limit priority, the priority is not reduced to the normal priority until all messages enqueued for this transaction name are processed. If you do not want the limit priority for this transaction, define equal values for the normal priority and limit priority, and a limit count of 65535.

When a transaction is processed exclusively by a batch message program (BMP), define the limit priority as 0. If the program specified by PGM() is defined with a program type of batch, the current priority is forced to be 0. However, a batch message processing region (BMP) can process transactions with current scheduling priorities other than 0.

This priority also controls the priority of messages created by this transaction and sent to a destination in a remote system. See also the discussion on MSC priorities under the NPRI definition.

The limit priority value is ignored for a transaction processed by a BMP.

Scheduling ignores the limit priority value for batch programs. If the transaction refers to a batch program and a non-zero LPRI value is specified, it is set to 0.

The limit priority value is ignored in a shared queues environment.

### **MAXRGN**

Specifies the maximum region count. This count value limits the number of message processing program (MPP) regions that can be concurrently scheduled to process a transaction. When the number of MPP regions is not limited, one transaction might monopolize all available regions. The value can be a number between 0 and the value specified on the MAXPST= control region parameter. MAXRGN(0) means that no limit is imposed. If you define the program with a scheduling type of SERIAL, omit the MAXRGN parameter or define it as 0.

Keyword combination rules include the following:

- PARLIM(65535) and MAXRGN>0 are mutually exclusive.
- SERIAL(Y) and MAXRGN>0 are mutually exclusive.

## **MSGTYPE**

Specifies the message type (single segment or multiple segment). It specifies the time at which an incoming message is considered complete and available to be routed to an application program for subsequent processing.

If MSC-directed routing is used in a multiple IMS system configuration, IMS does not ensure that both the message and the transaction destined to process that message are either single segment or multiple segments.

### **MULTSEG**

The incoming message can be more than one segment in length. It is not eligible for scheduling to an application program until an end-of-message indication is received, or a complete message is created by MFS. This is the default.

### **SNGLSEG**

The incoming message is one segment in length. It becomes eligible for scheduling when the terminal operator indicates end-of-segment.

Keyword combination rules include the following:

- FP(E) and MSGTYPE(MULTSEG) are mutually exclusive.
- FP(P) and MSGTYPE(MULTSEG) are mutually exclusive.

## **MSNAME**

Specifies the one- to eight-character name of the logical link path in a multiple IMS system configuration (MSC). A logical link path is a path between any two IMS systems. The IMS systems are identified by the remote system ID and the local system ID associated with the logical link path. The remote system ID identifies the system in which messages using this path are to be processed. The local system ID identifies this system. If this transaction is exported, the local and remote system IDs are exported, not the msname. For a CREATE TRAN command adding a remote transaction, the MSNAME must already be defined.

Keyword combination rules include the following:

- FP(E) and MSNAME are mutually exclusive.
- FP(P) and MSNAME are mutually exclusive.
- SIDL and MSNAME are mutually exclusive.
- SIDR and MSNAME are mutually exclusive.

## **NPRI**

Specifies the normal scheduling priority. The scheduling priority is an attribute used to select a transaction for scheduling. A transaction of higher priority is scheduled before a lower priority one, if they are defined with the same class. When a transaction resource is created, the current priority (CPRI), or the scheduling priority, is initialized to the normal priority (NPRI). The normal priority is also assigned to a transaction as the current priority when the number of input transactions enqueued and waiting to be processed is less than the limit count value. The value can be a number from 0 through 14. The default is 1.

This priority also controls the priority of messages created by this transaction and sent to a destination in a remote system.

When a transaction is processed exclusively by a batch message program (BMP), code the normal priority as 0.

When a transaction is processed exclusively by a batch message program (BMP), define the limit priority as 0. If the program specified by PGM() is defined with a program type of batch, the current priority is forced to be 0. However, a batch message processing region (BMP) can process transactions with current scheduling priorities other than 0.

For remote transactions, the priority used to send the transaction to the processing system, which is termed the MSC link message priority. The three MSC link message priority groups are:

- Low
- Medium
- High

The low priority group consists of primary requests in the input terminal system. This group is assigned remote transaction priorities from 0 to 6. The medium group consists of secondary requests, responses, primary requests in an intermediate system, and primary requests in the input terminal system. This group is assigned a remote transaction priority of 7. The high group consists of primary requests in the input terminal system. Messages in this group are assigned remote transaction priorities from 8 to 14. Within each group, messages have a priority based on the current priority value of the transaction or remote transaction in the input terminal system for primary requests, and on the latest processing system for secondary requests and responses.

In an MSC configuration, the transaction priority determines the priority used to send messages inserted by this transaction across an MSC link. If the transaction inserts multiple messages to the same destination (for example, pages to a printer) and these messages must be sent in the order inserted, the normal and limit priority values should be the same. If the normal and limit priority values are not identical, messages inserted at a higher priority than previously inserted messages could arrive at their destination first. (This restriction does not apply to multiple segments of the same message.)

The normal priority value is ignored for a transaction processed by a BMP.

Scheduling ignores the normal priority value for batch programs. If the transaction refers to a batch program and a non-zero NPRI value is specified, it is set to 0.

## **PARLIM**

Specifies the parallel processing limit count. This is the maximum number of messages that can currently be queued, but not yet processed, by each active message region currently scheduled for this transaction. This is the threshold value to be used when the associated program is defined with a scheduling type of parallel. An additional region is scheduled whenever the current transaction enqueue count exceeds the PARLIM value multiplied by the number of regions currently scheduled for this transaction.

The value can be a number from 0 to 32767 or 65535. PARLIM(0) indicates that any input message can cause a new region to be scheduled because the scheduling condition is always being met (the number of messages are greater than zero). If you specify PARLIM(0), you should specify a MAXRGN value to limit the number of regions that can be scheduled to process a particular transaction. PARLIM(65535) means that parallel processing is disabled and IMS allows the transaction to be scheduled in only one region at a time. PARLIM(65535) is the default.

The value specified for PARLIM applies to message processing programs (MPPs) only; it is not supported for batch message processing programs (BMPs).

If you define the transaction as serial or the program with a scheduling type as SERIAL, define PARLIM(65535).

In a shared-queues environment (when the scheduling type is PARALLEL), any PARLIM value other than 65535 causes a new region to be scheduled whenever the successful consecutive GU count exceeds the PARLIM value multiplied by the number of regions currently scheduled for this transaction. For shared-queues environments, the successful consecutive GU count is used instead of the queue count. New regions continue to be scheduled up to the maximum number of regions specified on MAXRGN.

The PARLIM keyword is not applicable to FPE transactions and should be allowed to default to 65535. A PARLIM value can be specified for FPE transactions, but it is ignored by scheduling. Specifying a PARLIM value that is not the default will result in a BAL status shown for an FPE transaction on commands such as /DISPLAY TRAN or QUERY TRAN. The PARLIM and BAL status can be ignored for FPE transactions.

Keyword combination rules include the following:

- MAXRGN>0 and PARLIM(65535) are mutually exclusive.
- SERIAL(Y) and PARLIM value 0 - 32767 are mutually exclusive.

#### **PGM**

Specifies the name of the program associated with the transaction.

For a CREATE TRAN command specified with REMOTE(N), the program must exist. PGM() is required, except for remote transactions defined with REMOTE(Y).

#### **PLCT**

Specifies the processing limit count. This is maximum number of messages sent to the application program by IMS for processing without reloading the application program. The value must be a number from 0 through 65535. PLCT(0) means the maximum number of messages sent to the application is one and the application program is reloaded before receiving a subsequent message. PLCT(65535) means that no limit is to be placed upon the number of messages processed at a single program load. Values 1 through 65535 are eligible for quick reschedule processing. The default is 65535.

The value is used to determine how many messages an application program is allowed to process in a single scheduling cycle. When the application program requests, and receives, the number of messages indicated, any subsequent requests result in one of two actions.

- IMS indicates “no more messages exist” if any of the following conditions is true:
  - The region is not an MPP.
  - The currently scheduled mode is not CMTMODE(SNGL).
  - Equal or higher priority transactions are enqueued for the region.

IMS might have other messages enqueued for the application program. It is the responsibility of the application program to terminate when it receives an indicator that no more messages are available. Termination of the application program makes the region it occupied available for rescheduling. This feature makes it possible for IMS to enable scheduling of higher priority transactions that entered the system while the previous transactions

were in process. In addition, if any equal-priority transactions are enqueued, they become eligible for scheduling on a first-in, first-out (FIFO) basis.

- The region goes through quick reschedule and returns the next message to the application if all of the following conditions are true:
  - The region is an MPP.
  - The transaction is CMTMODE(SNGL).
  - No equal or higher transactions are enqueued.
  - Messages are still enqueued for the application program.

### **PLCTTIME**

Specifies the processing limit count time. This is the amount of time (in hundredths of seconds) allowable to process a single transaction (or message). The number specifies the maximum CPU time allowed for each message to be processed in the message processing region.

Batch Message Programs (BMPs) are not affected by this setting.

The value can be a number, in hundredths of seconds, that can range from 1 to 6553500. A value of 6553500 means no time limit is placed on the application program. The default is 6553500.

If Fast Path is used, this keyword specifies, for a given transaction name, the amount of time (in hundredths of seconds) the program is allowed to process a single transaction message. The time represents real time that elapses during transaction processing (not accumulated task time). Real time is used because the input terminal is in response mode and cannot enter another transaction until the response is sent. In this case PLCT() is ignored.

The value controls application program looping. You are not required to optimize the value for program-transaction execution time. However, the time value assigned should not be less than the expected per-transaction execution time. If the scheduled application program exceeds the product of PLCTTIME() and PLCT(), the application program ends abnormally. If an IMS STIMER value of 2 is specified on the DFSMPR procedure, the region does not abend until completion of the DL/I call.

The application program must not use STIMER timer services. IMS uses STIMER timer services to time the execution of transactions. If an application program issues an MVS STIMER macro, it cancels the STIMER timer services set by IMS. Use the STIMERM macro instead for application program timer requests.

### **RECOVER**

Specifies the recovery option.

- Y** The transaction is recovered during IMS emergency or normal restart. This is the default.
- N** The transaction is not recovered.

Keyword combination rules include the following:

- CONV(Y) and RECOVER(N) are mutually exclusive.
- FP(E) and RECOVER(N) are mutually exclusive.
- FP(P) and RECOVER(N) are mutually exclusive.
- INQ(N) and RECOVER(N) are mutually exclusive.

### **REMOTE**

Specifies the remote option.

- N** The transaction is not remote. The transaction is local and runs on the local system. This is the default.
- Y** The transaction is remote. The transaction runs on a remote system.

Keyword combination rules include the following:

- REMOTE(Y) requires MSNAME or SIDR and SIDL.

### **RESP**

Specifies the response-mode option.

- N** The transaction is not response mode. For terminals specifying or accepting a default of OPTIONS=TRANRESP, input should not stop after this transaction is entered. This is the default.
- Y** The transaction is response mode. The terminal from which the transaction is entered is held and prevents further input until a response is received. For terminals specifying or accepting a default of OPTIONS=TRANRESP, no additional messages are to be allowed after this transaction is entered until this transaction sends a response message back to the terminal. Response mode can be forced or negated by individual terminal definition. RESP(Y) is ignored during online processing for all terminals that do not operate in response mode.

Keyword combination rules include the following:

- FP(E) and RESP(N) are mutually exclusive.
- FP(P) and RESP(N) are mutually exclusive.

### **SEGNO**

Specifies the segment number. This is the maximum number of application program output segments that are allowed into the message queues per Get Unique (GU) call from the application program. The value can be a number from 0 through 65535. The default is 0. If SEGNO(0) is defined, the number of segments is not checked by the online system at execution time.

### **SEGSZ**

Specifies the segment size. This is the maximum number of bytes allowed in any one output segment. The value can be a number from 0 through 65535. The default is 0. If SEGSZ(0) is defined, the segment size is not checked by the online system at execution time.

The maximum output message segment to an LU 6.2 device is 32767. If a transaction is expected to send output to an LU 6.2 device, the SEGSIZE parameter should be no greater than 32767. However, this is not enforced during processing of the command, because IMS cannot determine the device type for the message destination until output time.

### **SERIAL**

Specifies the serial option.

- N** Messages for the transaction are not processed serially. Message processing can be processed in parallel. Messages are placed on the suspend queue after a U3303 pseudoabend. Scheduling continues until repeated failures result in the transaction being stopped with a USTOP. This is the default.
- Y** Messages for the transaction are processed serially. U3303 pseudoabends do not cause the message to be placed on the suspend queue but rather on the front of the transaction message queue, and the transaction is stopped

with a USTOP. The USTOP of the transaction is removed when the transaction or the class is started with a /START or UPD TRAN command.

Keyword combination rules include the following:

- MAXRGN>0 and SERIAL(Y) are mutually exclusive.
- PARLIM value 0 - 32767 and SERIAL(Y) are mutually exclusive.

### **SIDL**

Specifies the system identification (SYSID) of the local system in a multiple-IMS system (MSC) configuration. The local system is the originating system to which responses are returned. The value can be a number from 1 to 2036, if MSC is enabled, or 0, if MSC is not enabled. The local SYSID can be defined in any or all of the MSNAMEs or transactions.

For a CREATE TRAN command creating a remote transaction by specifying SET(MSNAME()) or SET(SIDR(),SIDL()), where SIDR is different from SIDL, the MSNAME that represents that SIDL/SIDR pair must already be defined.

For a local transaction where SIDL is not specified, SIDL is defined with a value of the lowest system ID when MSC is enabled on this system, or SIDL is defined with a value of 0 when MSC is not enabled on this system.

The local system ID (SIDL) and the remote system ID (SIDR) are the same for local transactions. When creating a local transaction, if the SIDL and SIDR values are not defined as local to this IMS, the lowest system ID is used for the SIDL and SIDR values.

The SIDL parameter is independent of the link type (CTC, MTM, TCP/IP, VTAM) specified on the TYPE= keyword of the MSPLINK macro statement.

Keyword combination rules include the following:

- FP(E) and SIDL are mutually exclusive, unless SIDL and SIDR are specified as a pair and are equal to the local system ID of this IMS.
- FP(P) and SIDL are mutually exclusive.
- MSNAME and SIDL are mutually exclusive.
- SIDL value must be defined to this IMS.

### **SIDR**

Specifies the system identification (SYSID) of the remote system in a multiple-IMS system (MSC) configuration. The remote system is the system on which the application program executes. The value can be a number from 1 to 2036 if MSC is enabled, or 0 if MSC is not enabled. The remote SYSID specified must also be defined for an MSNAME.

For a CREATE TRAN command creating a remote transaction by specifying SET(MSNAME()) or SET(SIDR(),SIDL()), where SIDR is different from SIDL, the MSNAME that represents that SIDL/SIDR pair must already be defined.

Local transactions must have the SIDR value set equal to the SIDL value. Both values must be set to the same local SYSID.

The SIDR parameter is independent of the link type (CTC, MTM, TCP/IP, VTAM) specified on the TYPE= keyword of the MSPLINK macro statement.

Keyword combination rules include the following:

- FP(E) and SIDR are mutually exclusive, unless SIDL and SIDR are specified as a pair and are equal to the local system ID of this IMS.
- FP(P) and SIDR are mutually exclusive.

- MSNAME and SIDR are mutually exclusive.
- SIDR value must be defined to this IMS.

### **SPASZ**

Specifies the scratchpad area (SPA) size, in bytes, for a conversational transaction. The value can be a number from 16 and 32767.

Keyword combination rules include the following:

- CONV(N) and SPASZ are mutually exclusive.
- FP(E) and SPASZ are mutually exclusive.

### **SPATRUNC**

Specifies the scratchpad area (SPA) truncation option of a conversational transaction. This keyword defines whether the SPA data should be truncated or preserved across a program switch to a transaction that is defined with a smaller SPA. If this keyword is not specified for a conversational transaction, the value is obtained from the model (RSC, DESC, or default descriptor). If the SPATRUNC value is not specified or set in the model, the SPATRUNC value is set based on the value specified for the TRUNC parameter in the DFSDCxxx PROCLIB member at IMS cold start. Changing the DFSDCxxx TRUNC value across a restart has no effect on the default descriptor SPATRUNC value.

When a conversation initially starts, and when a program is switched, the SPATRUNC option is checked and set or reset as specified. When the option is set, it remains set for the life of the conversation, or until a program switch occurs to a transaction that specifies the option is to be reset.

When a program switch occurs, the truncated data option for the new transaction is first checked, and that specification is set for the conversation and is used for the SPA inserted into the output message. If the option is not specified for the new transaction, the option currently in effect for the conversation is used.

**S** IMS preserves all of the data in the SPA, even when a program switch is made to a transaction that is defined with a smaller SPA. The transaction with the smaller SPA does not see the truncated data, but when the transaction switches to a transaction with a larger SPA, the truncated data is used.

**R** The truncated data is not preserved.

Keyword combination rules include the following:

- CONV(N) and SPATRUNC are mutually exclusive.
- FP(E) and SPATRUNC are mutually exclusive.

### **TRANSTAT**

Specifies whether transaction level statistics are to be logged for message driven programs. If Y is specified, transaction level statistics are written to the log in a X'56FA' log record.

**N** Transaction level statistics are not logged.

**Y** Transaction level statistics are logged.

The TRANSTAT keyword is optional. If not specified and the default descriptor is the IMS-defined descriptor DFSDSTR1, the value is defined to be the DFSDFxxx PROCLIB member TRANSTAT parameter at IMS cold start. Changing the DFSDFxxx TRANSTAT value across a restart has no effect on the default descriptor TRANSTAT value. You can use the TRANSTAT keyword on

the CREATE TRAN command or the CREATE TRANDESC command to override the system default when you create a transaction or transaction descriptor.

#### **WFI**

Specifies the wait-for-input option. This attribute does not apply to Fast Path transactions, which always behave as wait-for-input transactions.

**N** This is not a wait-for-input transaction. This is the default.

**Y** This is a wait-for-input transaction. A message processing or batch processing application program that processes WFI transactions is scheduled and invoked normally. If the transaction to be processed is defined as WFI, the program is allowed to remain in main storage after it has processed the available input messages. The QC status code (no more messages) is returned to the program if the processing limit count (PLCT) is reached; a command is entered to change the status of the scheduled transaction, database, program, or class; commands relating to the databases used by the transaction are entered, or IMS is terminated with a checkpoint shutdown.

Keyword combination rules include the following:

- CMTMODE(MULT) and WFI(Y) are mutually exclusive.

### **Usage notes**

In an IMSplex where an RM and a resource structure are defined, the transaction names are saved in the resource structure.

Resources exist for the life of the IMS unless they are deleted using a DELETE command. Resources are recoverable across an IMS warm start or emergency restart. Resources are lost if IMS is cold started, unless cold start imports definitions that were exported while IMS was up.

The CREATE TRAN command is similar to local online change (using a /MODIFY command) or global online change (using an INITIATE OLC command) for transaction resources in the MODBLKS data set, except that transactions are added dynamically.

The CREATE TRAN command can be issued only through the Operations Manager API. These commands apply to DB/DC and DCCTL systems.

This command is not valid on the XRF alternate, RSR tracker, or FDBR region. The CREATE command is not valid if online change for MODBLKS is enabled (DFSDFxxx or DFSCGxxx defined with MODBLKS=OLC or MODBLKS not defined).

### **Output fields**

The following table shows the CREATE TRAN output fields. The columns in the table are as follows:

#### **Short label**

Contains the short label generated in the XML output.

#### **Keyword**

Identifies keyword on the command that caused the field to be generated. N/A appears for output fields that are always returned. The value *error* appears for output fields that are returned only in case of an error.

## Meaning

Provides a brief description of the output field.

Table 169. Output fields for the CREATE TRAN command

Short label	Keyword	Meaning
CC	N/A	Completion code.
CCTXT	<i>error</i>	Completion code text that briefly explains the meaning of the non-zero completion code.
MBR	N/A	IMSplex member that built the output line.
TRAN	TRAN	Transaction name.

## Return, reason, and completion codes

An IMS return and reason code is returned to OM by the CREATE TRAN command. The OM return and reason codes that might be returned as a result of the CREATE TRAN command are standard for all commands entered through the OM.

The following table includes the return and reason codes and a brief explanation of the codes. The return or reason code returned for the command might also indicate an error from a CSL request.

Table 170. Return and reason codes for the CREATE TRAN command

Return code	Reason code	Meaning
X'00000000'	X'00000000'	Command completed successfully. The command output contains a line for each resource, accompanied by its completion code. See the completion code table for details.
X'00000008'	X'00002009'	Multiple values specified for the same attribute are invalid. For example, you cannot specify both RESP(Y) and RESP(N). This reason code is applicable only when you create a transaction using the DFSINSX0 user exit.
X'00000008'	X'0000204C'	The class value is invalid.
X'00000008'	X'00002050'	The current priority CPRI value is invalid.
X'00000008'	X'00002054'	The limit count LCT value is invalid.
X'00000008'	X'00002058'	The limit priority LPRI is invalid.
X'00000008'	X'0000205C'	The maximum region MAXRGN is invalid.
X'00000008'	X'00002060'	The normal priority NPRI value is invalid.
X'00000008'	X'00002064'	The parallel limit count PARLIM value is invalid.
X'00000008'	X'00002068'	The processing limit count PLCT is invalid.
X'00000008'	X'0000206C'	The segment number SEGNO value is invalid.
X'00000008'	X'00002070'	The segment size SEGSZ value is invalid.
X'00000008'	X'00002100'	CMTMODE(MULT) mutually exclusive with WFI(Y). Either both keywords were explicitly specified, or one keyword was explicitly specified and the other attribute was obtained from the model.

Table 170. Return and reason codes for the CREATE TRAN command (continued)

Return code	Reason code	Meaning
X'00000008'	X'00002101'	CONV(Y) mutually exclusive with CMTMODE(MULT). Either both keywords were explicitly specified, or one keyword was explicitly specified and the other attribute was obtained from the model.
X'00000008'	X'00002103'	CONV(N) mutually exclusive with SPASZ>0 or SPATRUNC. Either both keywords were explicitly specified, or one keyword was explicitly specified and the other attribute was obtained from the model.
X'00000008'	X'00002104'	CONV(Y) mutually exclusive with RECOVER(N). Either both keywords were explicitly specified, or one keyword was explicitly specified and the other attribute was obtained from the model.
X'00000008'	X'00002105'	CONV(Y) requires SPASZ and SPATRUNC. Either SPASZ was not explicitly specified or defined in the model, or SPATRUNC was not explicitly specified or defined in the model. CONV(Y) was either explicitly specified or defined in the model.
X'00000008'	X'00002108'	Invalid EDITRTN name. EDITRTN was specified explicitly or obtained from the model.
X'00000008'	X'00002109'	Maximum 255 edit routines exceeded. EDITRTN was specified explicitly or obtained from the model.
X'00000008'	X'0000210A'	Invalid EMHBSZ. One of the following situations occurred: <ul style="list-style-type: none"> <li>• The EMHB size specified is either greater than the maximum size, which is 30720</li> <li>• The EMHB size specified plus the length of the X'5901' log record prefix is greater than the log buffer size</li> </ul>
X'00000008'	X'0000210C'	FP(E) mutually exclusive with EDITRTN. Either both keywords were explicitly specified, or one keyword was explicitly specified and the other attribute was obtained from the model.
X'00000008'	X'0000210E'	FP(E) or FP(P) mutually exclusive with MSC keyword, either MSNAME or SIDR and SIDL. Either both keywords were explicitly specified, or one keyword was explicitly specified and the other attribute was obtained from the model.
X'00000008'	X'0000210F'	FP(E) or FP(P) mutually exclusive with MSGTYPE(MULTSEG). Either both keywords were explicitly specified, or one keyword was explicitly specified and the other attribute was obtained from the model.
X'00000008'	X'00002110'	FP(N) mutually exclusive with EMHBSZ > 0. Either both keywords were explicitly specified, or one keyword was explicitly specified and the other attribute was obtained from the model.
X'00000008'	X'00002111'	FP(E) or FP(P) mutually exclusive with RECOVER(N). Either both keywords were explicitly specified, or one keyword was explicitly specified and the other attribute was obtained from the model.

Table 170. Return and reason codes for the CREATE TRAN command (continued)

Return code	Reason code	Meaning
X'00000008'	X'00002112'	FP(E) or FP(P) mutually exclusive with RESP(N). Either both keywords were explicitly specified, or one keyword was explicitly specified and the other attribute was obtained from the model.
X'00000008'	X'00002116'	INQ(N) mutually exclusive with RECOVER(N). Either both keywords were explicitly specified, or one keyword was explicitly specified and the other attribute was obtained from the model.
X'00000008'	X'00002117'	Invalid LIKE descriptor name.
X'00000008'	X'00002118'	Invalid LIKE resource name.
X'00000008'	X'00002119'	MSC keyword MSNAME or SIDR and SIDL are mutually exclusive with application program defined as Fast Path exclusive (FP(E)) associated with this transaction.
X'00000008'	X'0000211A'	Invalid MSNAME name.
X'00000008'	X'0000211B'	MSNAME mutually exclusive with SIDR and SIDL.
X'00000008'	X'0000211D'	MAXRGN>0 mutually exclusive with PARLIM(65535). Either both keywords were explicitly specified, or one keyword was explicitly specified and the other attribute was obtained from the model.
X'00000008'	X'0000211E'	MAXRGN>0 mutually exclusive with SERIAL(Y). Either keywords were explicitly specified, or one keyword was explicitly specified and the other attribute was obtained from the model.
X'00000008'	X'00002120'	PGM() definition missing. PGM must be defined for local transactions either explicitly or in the model.
X'00000008'	X'00002121'	PARLIM value mutually exclusive with SERIAL(Y). Either both keywords were explicitly specified, or one keyword was explicitly specified and the other attribute was obtained from the model.
X'00000008'	X'00002123'	Invalid PGM name. PGM was specified explicitly, or obtained from the model. If the IMS-defined descriptor DFSDSTR1 was used, the command failed because DFSDSTR1 defines no program name.
X'00000008'	X'00002125'	If REMOTE(Y) is specified, the SIDR value must be a remote SYSID and the SIDL value must be a local SYSID. If neither the MSNAME keyword nor the SIDR and SIDL keywords are specified explicitly, the SIDR and SIDL values are obtained from the model. If REMOTE(N) is specified, the SIDR value must be equal to the SIDL value. The SIDR and SIDL values can be specified explicitly or obtained from the model. The MSNAME keyword is not allowed with REMOTE(N). The REMOTE value can be specified explicitly or obtained from the model.
X'00000008'	X'00002126'	Invalid SIDL value.
X'00000008'	X'00002127'	SIDL/SIDR must be specified as a pair. Either SIDL was specified alone or SIDR was specified alone.
X'00000008'	X'00002128'	Invalid SIDR value.

Table 170. Return and reason codes for the CREATE TRAN command (continued)

Return code	Reason code	Meaning
X'00000008'	X'0000212A'	The CREATE TRAN command is rejected for the transaction, because it specifies a batch program and an attribute value that is not allowed for a batch program such as a PARLIM value other than 65535.
X'00000008'	X'00002135'	Parlim/schdtype(serial) conflict. If PARLIM is specified and defined with a value other than 65535 and the program is defined as schdtype(serial), the command is rejected.
X'00000008'	X'00002140'	Fast Path exclusive FP(E) is mutually exclusive with any conversation keyword, including CONV(Y), SPASZ, and SPATRUNC. An FP exclusive transaction cannot be defined as conversational.
X'0000000C'	X'00003000'	Command was successful for some resources but failed for others. The command output contains a line for each resource, accompanied by its completion code. See the completion code table for details.
X'0000000C'	X'00003004'	Command was not successful for any of the resources. The command output contains a line for each resource, accompanied by its completion code. See the completion code table for details.
X'00000010'	X'0000400C'	Command is not valid on the XRF alternate.
X'00000010'	X'00004014'	Command is not valid on the RSR tracker.
X'00000010'	X'00004024'	No Fast Path defined, so FP(E), FP(P), or EMHBSZ > 0 is invalid.
X'00000010'	X'00004120'	Online change phase is in progress.
X'00000010'	X'00004204'	Model is quiesced. Cannot quiesce model.
X'00000010'	X'00004300'	Command is not allowed because online change for MODBLKS is enabled (DFSDFxxx or DFSCGxxx defined with MODBLKS=OLC, or MODBLKS not defined).
X'00000010'	X'00004310'	Storage could not be obtained for the Transaction Input edit routine table. A cold start is required to fix this error.
X'00000010'	X'00004314'	The Transaction Input edit routine could not be loaded.
X'00000010'	X'00004318'	A new Transaction Input edit routine could not be added. The maximum of 255 routines has already been reached.
X'00000014'	X'00005004'	DFSOCMD response buffer could not be obtained.
X'00000014'	X'00005008'	DFSPOOL storage could not be obtained.
X'00000014'	X'0000500C'	AWE could not be obtained.
X'00000014'	X'00005010'	Unable to obtain latch.
X'00000014'	X'00005114'	Resource or descriptor that is specified as a model is in the process of being imported from the change list in the IMSRSC repository or was not imported successfully from the change list at the end of IMS restart. This resource or descriptor cannot be referenced as a model on a CREATE command until it is successfully imported from the repository.

The following table includes an explanation of the completion codes.

Table 171. Completion codes for the CREATE TRAN command

Completion code	Completion code text	Meaning
0		Command completed successfully for transaction.
11	RESOURCE ALREADY EXISTS	Transaction already exists.
17	ANOTHER CMD IN PROGRESS	Another command (such as DELETE PGM or UPDATE PGM) is in progress for the program referenced by this transaction.
22	CPIC TRAN ALREADY EXISTS	CPI-C transaction already exists by that name.
36	FP=E/FP=N PGM CONFLICT	Transaction to be created as Fast Path exclusive FP(E) conflicts with program already defined as non-Fast Path FP(N).
3E	FP=N/FP=E PGM CONFLICT	Transaction to be created as non-Fast Path FP(N) conflicts with program already defined as Fast Path exclusive FP(E).
3F	FP=P/BMPTYPE=Y CONFLICT	Transaction to be created as Fast Path potential FP(P) conflicts with the program defined as batch BMPTYPE(Y).
4E	LTERM ALREADY EXISTS	LTERM already exists by that name.
5B	MSNAME ALREADY EXISTS	MSNAME already exists by that name.
5F	INVALID CHARACTERS IN NAME	Name is invalid because it contains an invalid character.
61	DFSBCB STORAGE ERROR	Request to get BCB storage for SMB control block failed.
6A	FP=P/FP=E PGM CONFLICT	Transaction to be created as Fast Path potential FP(P) conflicts with program already defined as Fast Path exclusive FP(E).
78	RM CONFLICT	LTERM, CPIC transaction, MSNAME, or descriptor already exists in RM by that name.
8A	WILDCARD PARAMETER NOT SUPPORTED	The CREATE command does not support wildcard parameters. You must explicitly specify the names of the resources you want to create.
90	INTERNAL ERROR	An IMS system service failure.
93	RESERVED NAME	Name is reserved. For example, name starts with DFS (except for DFSSAMxx or DFSIVPxx), or is an IMS reserved name such as BASICEDT or WTOR.

Table 171. Completion codes for the CREATE TRAN command (continued)

Completion code	Completion code text	Meaning
B5	ROUTING CODE ALREADY EXISTS	Transaction creation failed for a Fast Path exclusive FP(E) transaction. The routing code could not be created, because one already exists with the same name as the transaction to be created.
B6	LATCH ERROR	Unable to obtain latch.

## Examples

The following are examples of the CREATE TRAN command:

### Example 1 for CREATE TRAN command

TSO SPOC input:

```
CREATE TRAN NAME(TRN00001,TRN00002) LIKE(RSC(APOL18))
SET(NPRI(6),LPRI(10),TRANSTAT(Y),CLASS(2))
```

TSO SPOC output:

```
Response for: CREATE TRAN NAME(TRN00001,TRN00002) LIKE(RSC(APO...
Trancode MbrName CC
TRN00001 IMS1 0
TRN00002 IMS1 0
```

OM API input:

```
CMD(CRE TRAN NAME(TRN00001,TRN00002) LIKE(RSC(APOL18))
SET(NPRI(6),LPRI(10),TRANSTAT(Y),CLASS(2))
```

OM API output:

```
<imsout>
<ctl>
<omname>OM10M </omname>
<omvsn>1.3.0</omvsn>
<xm1vsn>20 </xm1vsn>
<statime>2006.307 19:23:15.095304</statime>
<stotime>2006.307 19:23:15.096942</stotime>
<staseq>BFA6CF3A7E308C1C</staseq>
<stoseq>BFA6CF3A7E96EA52</stoseq>
<rqsttkn1>USRT011 10112315</rqsttkn1>
<rc>00000000</rc>
<rsn>00000000</rsn>
</ctl>
<cmd>
<master>IMS1 </master>
<userid>USRT011 </userid>
<verb>CRE </verb>
<kwd>TRAN </kwd>
<input>CREATE TRAN NAME(TRN00001,TRN00002) LIKE(RSC(APOL18))
SET(NPRI(6),LPRI(10),TRANSTAT(Y),CLASS(2)) </input>
</cmd>
<cmdrsphdr>
<hdr s1b1="TRAN" l1b1="Trancode" scope="LCL" sort="a" key="1"
scroll="no" len="8" dtype="CHAR" align="left" />
<hdr s1b1="MBR" l1b1="MbrName" scope="LCL" sort="a" key="2" scroll="no"
len="8" dtype="CHAR" align="left" />
<hdr s1b1="CC" l1b1="CC" scope="LCL" sort="n" key="0" scroll="yes"
```

```

len="4" dtype="INT" align="right" skipb="no" />
<hdr s1b1="CCTXT" l1b1="CCText" scope="LCL" sort="n" key="0"
scroll="yes" len="*" dtype="CHAR" skipb="yes" align="left" />
</cmdrsphdr>
<cmdrspdata>
<rsp>TRAN(TRN00001) MBR(IMS1) CC( 0) </rsp>
<rsp>TRAN(TRN00002) MBR(IMS1) CC( 0) </rsp>
</cmdrspdata>
</imsout>

```

**Explanation:** Transactions named TRN00001 and TRN00002 are created like transaction APOL18, with several attributes specified with the SET keyword to override those attributes obtained from transaction APOL18.

**Related concepts:**

 [How to interpret CSL request return and reason codes \(System Programming APIs\)](#)

**Related reference:**

Chapter 2, “Command keywords and their synonyms,” on page 63

 [Commit Verify exit routine \(Exit Routines\)](#)

## CREATE TRANDESC command

Use the CREATE TRANDESC command to create transaction descriptors. A descriptor is a model that can be used to create descriptors or resources.

A descriptor is created with a value set for every attribute. Attributes not explicitly specified on the CREATE command take the default value. Any transaction descriptor can be created using this descriptor as a model, by specifying the CREATE command with LIKE(DESC(descriptor\_name)).

Subsections:

- “Environment”
- “Syntax” on page 317
- “Keywords” on page 318
- “Usage notes” on page 332
- “Output fields” on page 332
- “Return, reason, and completion codes” on page 333
- “Examples” on page 337

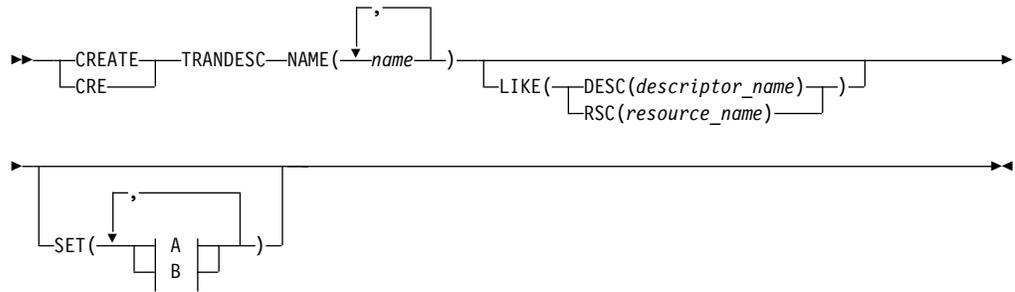
### Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

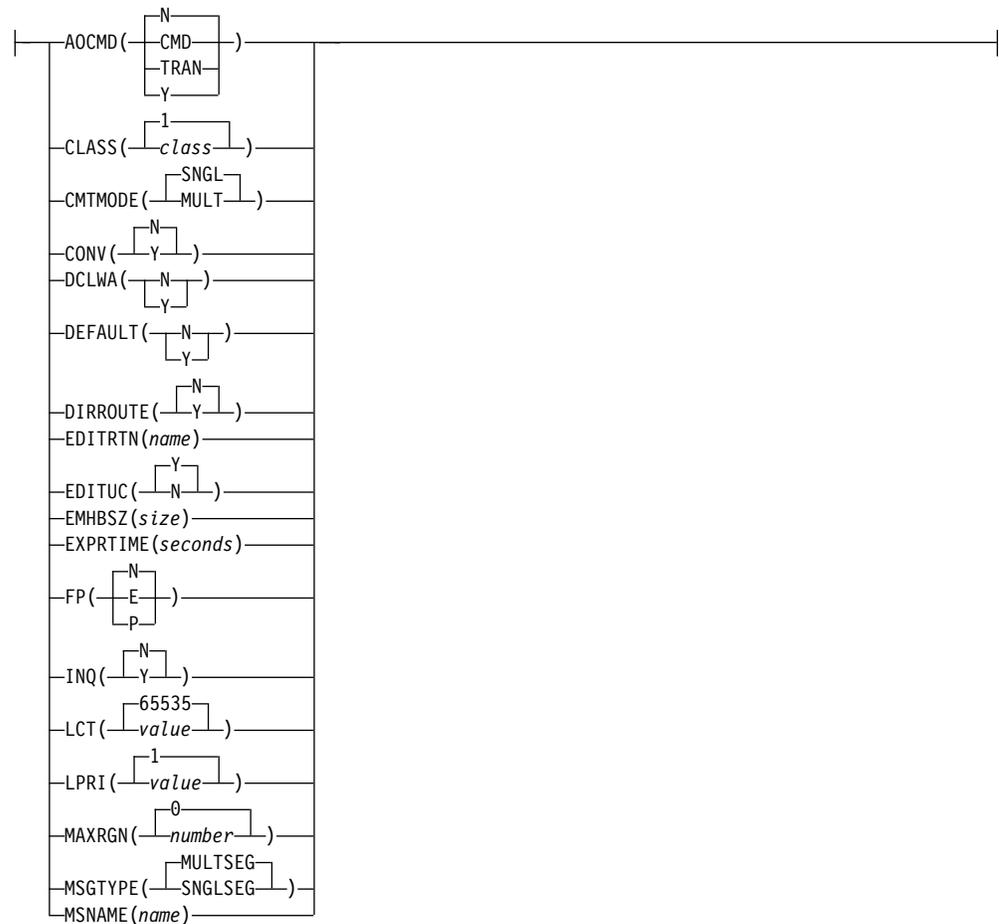
*Table 172. Valid environments for the CREATE TRANDESC command and keywords*

Command / Keywords	DB/DC	DBCTL	DCCTL
CREATE TRANDESC	X		X
LIKE	X		X
NAME	X		X
SET	X		X

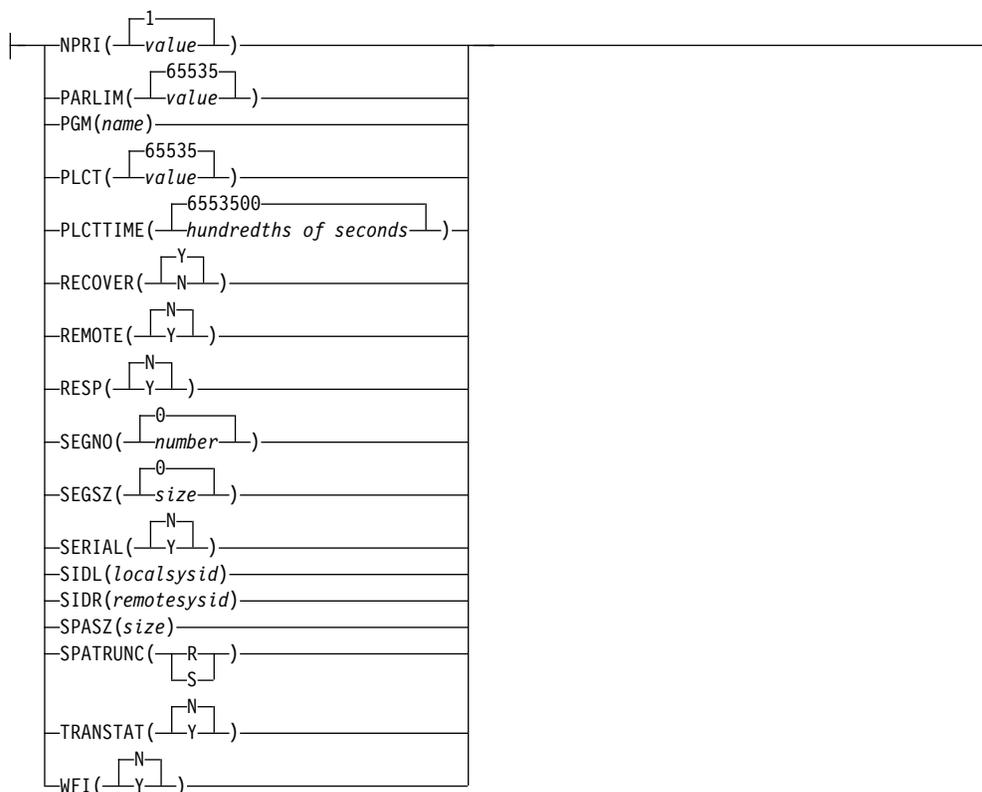
## Syntax



### A:



### B:



Default values shown in this syntax diagram reflect the default values that are defined in the IMS transaction descriptor, DFSDSTR1. If you define another descriptor as the default with a CREATE or UPDATE TRANDESC command, the default values shown in this syntax diagram might not match the value defined in the current default descriptor.

## Keywords

The following keywords are valid for the CREATE TRANDESC command:

### LIKE

Specifies that the descriptor is created using an existing descriptor as a model, instead of the default descriptor. The default descriptor is either the IMS descriptor DFSDSTR1, or user-defined. The model type can either be a descriptor (DESC), or a resource (RSC). The descriptor is created with all the same attributes as the model, except for the DEFAULT(Y) value and the CPRI value, which is a runtime scheduling value that does not apply to transaction descriptors. Attributes set explicitly by the CREATE command override the model attributes. DEFAULT(Y) must be specified explicitly to make a descriptor the default descriptor. Later changes to the model are not propagated to resources or descriptors that were created from it.

The QUERY commands can be used to display the model name and model type used to create the descriptor or resource. The model name and type returned from the QUERY command are provided for informational purposes only. The EXPORT command exports a resource definition including the model name and model type to the definition member. The IMPORT command imports a resource definition including the model name and model type from

the definition member. An IMPORT command that creates a resource saves the model name and model type, but does not use the models attributes.

**DESC(descriptor\_name)**

Specifies the name of the descriptor to use as a model to define this descriptor.

**RSC(resource\_name)**

Specifies the name of the resource to use as a model to define this descriptor.

**NAME**

Specifies the 1-8 character name of the transaction. Names must be alphanumeric (A through Z, 0 through 9, #, \$, and @). Names cannot include a blank, comma, period, hyphen, or equal sign. Names cannot begin with DFS, except for DFSIVPxx and DFSSAMxx. Names cannot be BASICEDT, DBCDM, DBRC, ISCEDT, MSDB, SDB, or WTOR. A transaction resource and a transaction descriptor can have the same name. A transaction cannot have the same name as a logical terminal.

**SET**

Specifies the attributes of the transaction to be created. If the LIKE keyword is omitted, attributes not specified take the value defined in the default descriptor, which is either the IMS descriptor DFSDSTR1, or user-defined. If the LIKE keyword is specified, attributes not specified take the value defined in the model. A transaction must have the same characteristics in all systems where it is defined when it is shared. These characteristics include:

- Nonconversational or conversational
- SPA size if conversational
- Single- or multi-segment messages
- Noninquiry or inquiry
- Recoverable or nonrecoverable
- Response mode or non-response mode
- Fast Path exclusive, Fast Path potential, or non-Fast Path

**AOCMD**

Specifies the AOI option indicating whether the transaction can issue the type-1 AOI CMD call or the type-2 AOI ICMD call. If AOCMD is defined as CMD, TRAN, or Y, and the AOI1 execute parameter is defined as AOI1=N, no authorization checking is done, and the transaction is permitted to issue CMD and ICMD calls.

**N** Indicates that the transaction is not permitted to issue type-1 AOI CMD calls. The transaction is permitted to issue type-2 AOI ICMD calls.

**CMD**

Indicates that the transaction is permitted to issue type-1 AOI CMD calls and type-2 AOI ICMD calls. If the AOI1 execute parameter is defined as C, R, or A, authorization checking is based on which transactions can issue a particular command. In this case, the commands (or the first three characters of the commands) need to be defined to RACF or an equivalent product as a user. The type-1 AOI transactions must be defined as profiles under the TIMS class, and for each transaction, the commands it can issue must be specified. Defining AOCMD(CMD) requires you to create fewer

user IDs than you need to create for the AOCMD(TRAN) definition. However, defining AOCMD(CMD) requires you to create or modify a larger number of resource profiles.

#### **TRAN**

Indicates that the transaction is permitted to issue type-1 AOI CMD calls and type-2 AOI ICMD calls. If the AOI1 execute parameter is defined as C, R, or A, the transaction code is used for authorization. The first authorization check results in the accessor environment element (ACEE) being built. This environment is kept for use by future authorization checks. The type-1 AOI transaction needs to be defined to RACF or an equivalent product as a user. The transactions will then be specified on RACF PERMIT statements for each command they are allowed to issue from a type-1 AOI transaction. Specifying AOI transactions as users to RACF might conflict with the name of a user already defined to RACF. If this occurs, then either the transaction name or the existing user name needs to be changed.

**Y** Indicates that the transaction is permitted to issue type-1 AOI CMD calls and type-2 AOI ICMD calls. If the AOI1 execute parameter is defined as C, R, or A, the user ID or the program name is used for authorization. For some environments, if a Get Unique call has not yet occurred, the program name is used for authorization.

#### **CLASS**

Specifies the transaction class, which is an attribute used to select a transaction for scheduling. A transaction can be scheduled if there is a message processing region available for that class. The value can be a number from 1 to 999. The default is 1. This value must not exceed the value given (by specification or default) on the MAXCLAS= keyword of the IMSCTRL macro.

Define CPI-C transactions with a different message class from that used for non-CPI-C transactions. IMS handles all CPI-C transactions as priority zero within the transaction class.

#### **CMTMODE**

Specifies when database updates and non-express output messages are committed. This operand affects emergency restart.

#### **MULT**

Database updates and non-express output messages are committed only when the application program terminates normally, when the processing limit count has been reached, or, in the case of a pseudo-WFI dependent region, when there are no more messages on the queue. For example, if five transactions are processed during a single schedule of a program, all five are committed only when the fifth one is completed and the program terminates. Until a transaction has been committed, locks for updated database records are not released and non-express output messages are not queued for output. If an application ends abnormally before committing its messages, emergency restart requeues all the messages that were processed within the commit scope and makes them available for reprocessing.

If the transaction results in the application calling an external subsystem, such as DB2, the Commit Verify exit provided by the external subsystem can determine whether CMTMODE(MULT) is supported. See documentation under the Commit Verify exit routine in *IMS Version 14 Exit Routines*.

## SNGL

Database updates and non-express output messages are committed when the application program completes processing each transaction. IMS invokes commit processing either when the application program requests the next message (issues a GU to the IO-PCB), or when the application program terminates. If an application ends abnormally before committing its message, emergency restart requeues the message that was in process before the abend and makes it available for reprocessing. This is the default.

The TRANSACT macro MODE keyword uses a default of MULT. The CRE TRAN command CMTMODE keyword uses a default of SNGL.

Keyword combination rules include the following:

- CONV(Y) and CMTMODE(MULT) are mutually exclusive.
- CMTMODE(MULT) and WFI(Y) are mutually exclusive.

## CONV

Specifies the conversation option.

**N** The transaction is not conversational. This is the default.

**Y** The transaction is conversational. The transaction message is destined for a conversational program. A conversational program processes transactions made up of several steps. A conversational program receives a message from a terminal, replies to the terminal, but saves the data from the transaction in a scratchpad area (SPA). When the person at the terminal enters more data, the program has the data it saved from the last message in the SPA, so it can continue processing the request without the person at the terminal having to enter the data again.

Keyword combination rules include the following:

- CMTMODE(MULT) and CONV(Y) are mutually exclusive.
- CONV(Y) requires SPASZ and SPATRUNC.
- INQ(Y) and CONV(Y) are mutually exclusive.
- RECOVER(N) and CONV(Y) are mutually exclusive.
- SPASZ and CONV(N) are mutually exclusive.
- SPATRUNC and CONV(N) are mutually exclusive.

## DCLWA

Specifies the log write-ahead option. If not specified, the value is defined to be the DCLWA parameter in the DFSDFxxx PROCLIB member. If DCLWA is not defined in the DFSDFxxx PROCLIB member, the default is DCLWA=Y.

**N** IMS should not perform log write-ahead. Specify N if input message integrity and the consistency of output messages with associated database updates is not required. DCLWA does not apply to response mode or Fast Path input processing, and is ignored during IMS execution.

**Y**

IMS should perform log write-ahead for recoverable, nonresponse input messages and transaction output messages. This ensures the following:

- A nonresponse input transaction is made recoverable across IMS failures before IMS acknowledges receipt of the input.
- Database changes are made recoverable before IMS sends associated output reply messages.

- Information in the log buffers is written to the IMS log, before the associated input acknowledgment or output reply is sent to the terminal.

Define DCLWA(Y) for all VTAM terminal types.

#### **DEFAULT**

Specifies whether the descriptor is the default.

- N** The descriptor is not the default.
- Y** The descriptor is the default, which resets the existing default descriptor to DEFAULT(N). When a descriptor or resource is created without the LIKE keyword, any attribute not specified on the CREATE command takes the value defined in the default descriptor. Only one descriptor can be defined as the default for a resource type. IMS defines a default transaction descriptor called DFSDSTR1, where all attributes are defined with the default value. Defining a user-defined descriptor as the default overrides the IMS-defined descriptor. Since only one transaction descriptor can be the default at one time, only one transaction name can be specified with DEFAULT(Y).

#### **DIRROUTE**

Specifies the MSC directed routing option.

- N** The application program processing a transaction is not informed of the system which originated the transaction. The name of the originating LTERM is placed in the I/O PCB. This is the default.
- Y** The application program processing a transaction is informed of the system which originated the transaction, if MSC directed routing is used in a multiple IMS system configuration. An MSNAME corresponding to a logical path back to the originating system is placed in the I/O PCB.

#### **EDITRTN**

Specifies the one- to eight-character name of your transaction input edit routine that edits messages before the program receives the message. This name must begin with an alphabetic character. The specified edit routine (load module) must reside on the USERLIB data set. This routine cannot be the same as the one defined by the system definition TYPE EDIT= parameter. A maximum of 255 input edit routines are supported.

EDITRTN is used for a Fast Path potential transaction when the transaction is routed to IMS.

For input from LU 6.2 devices, the user edit exit routine DFSLUEE0 is called instead of the transaction input edit routine specified in EDITRTN.

Keyword combination rules include the following:

- FP(E) and EDITRTN are mutually exclusive.

#### **EDITUC**

Specifies the edit to uppercase option.

- N** The input data is not translated to uppercase. It can consist of upper and lowercase characters as entered from the terminal.
- Y** The input data is to be translated to uppercase before it is presented to the processing program. If FP(E) or FP(P), the transaction is to be translated to uppercase before being presented to the edit/routing exit routine. This is the default.

Specifying EDITUC(Y) for VTAM terminals prevents the transmission of embedded device control characters.

## EMHBSZ

Specifies the EMH buffer size required to run the Fast Path transaction. This overrides the EMHL execution parameter. If EMHBSZ is not specified, the EMHL execution parameter value is used. The value can be a number from 12 to 30720.

Keyword combination rules include the following:

- EMHBSZ>0 requires Fast Path to be defined.
- FP(N) and EMHBSZ>0 are mutually exclusive.

## EXPRTIME

Specifies the elapse time in seconds that IMS can use to cancel the input transaction. After a transaction is submitted to IMS, the transaction could be delayed for processing because of a stopped transaction or a potential system slow down. In that case, the user or client application could time out before the transaction is processed. When IMS eventually schedules and processes the transaction, the response message is no longer wanted. With the elapse time specified for the transaction, IMS can flag the input transaction as expired so that the system does not waste CPU cycles to process the unwanted transaction.

The value can be a number, in seconds, which can range from 0 to 65535. The default is 0, which means that no expiration time is set for this transaction. The transaction expiration attribute is supported by all of the IMS TM interfaces.

**Restriction:** The transaction expiration checking is not performed at the GU time for Fast Path transactions, IMS conversational transactions, and program-to-program switch transactions.

**FP** Specifies the Fast Path option.

- E** The transaction is processed exclusively as Fast Path. A Fast Path routing code by the same name is created. The program must be defined as Fast Path exclusive.
- N** The transaction is not a candidate for Fast Path processing. The program identified by PGM() must be defined as not Fast Path. This is the default.
- P** The transaction is a potential candidate for Fast Path processing. Fast Path-potential transactions must be able to run under two programs: a Fast Path exclusive program a non-Fast Path program. This CREATE TRAN command defines the non-Fast Path program with PGM(). A Fast Path exclusive program should be defined to which this transaction can be routed. Fast Path-potential transactions must be processed by a user edit/routing exit to determine whether the transaction is actually to be processed by IMS Fast Path. If it is to be processed by IMS Fast Path, the edit/routing exit routine associates the transaction with a routing code. This routing code identifies which Fast Path program is to process the transaction.

The program defined by PGM() must not be defined as Fast Path exclusive.

Keyword combination rules include the following:

- EDITRTN and FP(E) are mutually exclusive.
- FP(P) and FP(E) require Fast Path to be defined.
- MSGTYPE(MULTSEG) and FP(E) are mutually exclusive.
- MSGTYPE(MULTSEG) and FP(P) are mutually exclusive.
- MSNAME and FP(E) are mutually exclusive.

- MSNAME and FP(P) are mutually exclusive.
- RECOVER(N) and FP(E) are mutually exclusive.
- RECOVER(N) and FP(E) are mutually exclusive.
- RESP(N) and FP(E) are mutually exclusive.
- RESP(N) and FP(P) are mutually exclusive.
- SIDL and FP(E) are mutually exclusive.
- SIDL and FP(P) are mutually exclusive.
- SIDR and FP(E) are mutually exclusive.
- SIDR and FP(P) are mutually exclusive.

#### **INQ**

Specifies the inquiry option.

**N** This is not an inquiry transaction. This is the default.

**Y** This is an inquiry transaction. If INQ(Y) is specified, you can also specify whether this transaction should be recovered during an IMS emergency or normal restart using the RECOVER() parameter.

This should be specified only for those transactions that, when entered, do not cause a change in any database. Programs are prohibited from issuing ISRT, DLET, or REPL calls to a database when scheduled to process a transaction defined as INQ(Y).

An application program cannot do an SQL INSERT, DELETE, or UPDATE when the IMS transaction is defined with INQ(Y).

Keyword combination rules include the following:

- CONV(Y) and INQ(Y) are mutually exclusive.
- RECOVER(N) and INQ(N) are mutually exclusive.

#### **LCT**

Specifies the limit count. This is the number that, when compared to the number of input transactions queued and waiting to be processed, determines whether the normal or limit priority value is assigned to this transaction. The value can be a number from 1 to 65535. The default is 65535.

The limit count value is ignored for a transaction processed by a BMP.

The limit count value is ignored in a shared queues environment.

#### **LPRI**

Specifies the limit priority. This is the scheduling priority to which this transaction is raised when the number of input transactions enqueued and waiting to be processed is equal to or greater than the limit count value. The scheduling priority is an attribute used to select a transaction for scheduling. A transaction of higher priority is scheduled before a lower priority one, if they are defined with the same class. The value can be a number from 0 through 14. The default is 1.

When the limit priority is used and the scheduling priority is raised to the limit priority, the priority is not reduced to the normal priority until all messages enqueued for this transaction name are processed. If you do not want the limit priority for this transaction, define equal values for the normal priority and limit priority, and a limit count of 65535.

When a transaction is processed exclusively by a batch message program (BMP), define the limit priority as 0. If the program specified by PGM() is

defined with a program type of batch, the current priority is forced to be 0. However, a batch message processing region (BMP) can process transactions with current scheduling priorities other than 0.

This priority also controls the priority of messages created by this transaction and sent to a destination in a remote system. See also the discussion on MSC priorities under the NPRI definition.

The limit priority value is ignored for a transaction processed by a BMP.

The limit priority value is ignored in a shared-queues environment.

#### **MAXRGN**

Specifies the maximum region count. This count value limits the number of message processing program (MPP) regions that can be concurrently scheduled to process a transaction. When the number of MPP regions is not limited, one transaction might monopolize all available regions. The value can be a number from 0 to the value specified on the MAXPST= control region parameter. The default of MAXRGN(0) means that no limit is imposed.

If you define the program with a scheduling type of SERIAL, omit the MAXRGN keyword or define it as 0.

The following keyword combinations are mutually exclusive:

- PARLIM(65535) and MAXRGN value greater than 0
- SERIAL(Y) and MAXRGN value greater than 0

#### **MSGTYPE**

Specifies the message type (single segment or multiple segment). It specifies the time at which an incoming message is considered complete and available to be routed to an application program for subsequent processing.

If MSC-directed routing is used in a multiple IMS system configuration, IMS does not ensure that both the message and the transaction destined to process that message are either single segment or multiple segments.

#### **MULTSEG**

The incoming message can be more than one segment in length. It is not eligible for scheduling to an application program until an end-of-message indication is received, or a complete message is created by MFS. This is the default.

#### **SNGLSEG**

The incoming message is one segment in length. It becomes eligible for scheduling when the terminal operator indicates end-of-segment.

Keyword combination rules include the following:

- FP(E) and MSGTYPE(MULTSEG) are mutually exclusive.
- FP(P) and MSGTYPE(MULTSEG) are mutually exclusive.

#### **MSNAME**

Specifies the one- to eight-character name of the logical link path in a multiple IMS system configuration (MSC). A logical link path is a path between any two IMS systems. The IMS systems are identified by the remote system ID and the local system ID associated with the logical link path. The remote system ID identifies the system in which messages using this path are to be processed. The local system ID identifies this system. If this transaction is exported, the

local and remote system IDs are exported, not the msname. For a CREATE TRAN command adding a remote transaction, the MSNAME must already be defined.

Keyword combination rules include the following:

- FP(E) and MSNAME are mutually exclusive.
- FP(P) and MSNAME are mutually exclusive.
- SIDL and MSNAME are mutually exclusive.
- SIDR and MSNAME are mutually exclusive.

#### **NPRI**

Specifies the normal scheduling priority. The scheduling priority is an attribute used to select a transaction for scheduling. A transaction of higher priority is scheduled before a lower priority one, if they are defined with the same class. When a transaction resource is created, the current priority (CPRI), or the scheduling priority, is initialized to the normal priority (NPRI). The normal priority is also assigned to a transaction as the current priority when the number of input transactions enqueued and waiting to be processed is less than the limit count value. The value can be a number from 0 through 14. The default is 1.

This priority also controls the priority of messages created by this transaction and sent to a destination in a remote system.

When a transaction is processed exclusively by a batch message program (BMP), code the normal priority as 0.

When a transaction is processed exclusively by a batch message program (BMP), define the limit priority as 0. If the program specified by PGM() is defined with a program type of batch, the current priority is forced to be 0. However, a batch message processing region (BMP) can process transactions with current scheduling priorities other than 0.

For remote transactions, the priority used to send the transaction to the processing system, which is termed the MSC link message priority. The three MSC link message priority groups are:

- Low
- Medium
- High

The low priority group consists of primary requests in the input terminal system. This group is assigned remote transaction priorities from 0 to 6. The medium group consists of secondary requests, responses, primary requests in an intermediate system, and primary requests in the input terminal system. This group is assigned a remote transaction priority of 7. The high group consists of primary requests in the input terminal system. Messages in this group are assigned remote transaction priorities from 8 to 14. Within each group, messages have a priority based on the current priority value of the transaction or remote transaction in the input terminal system for primary requests, and on the latest processing system for secondary requests and responses.

In an MSC configuration, the transaction priority determines the priority used to send messages inserted by this transaction across an MSC link. If the transaction inserts multiple messages to the same destination (for example, pages to a printer) and these messages must be sent in the order inserted, the normal and limit priority values should be the same. If the normal and limit priority values are not identical, messages inserted at a higher priority than

previously inserted messages could arrive at their destination first. (This restriction does not apply to multiple segments of the same message.)

The normal priority value is ignored for a transaction processed by a BMP.

#### **PARLIM**

Specifies the parallel processing limit count. This is the maximum number of messages that can currently be queued, but not yet processed, by each active message region currently scheduled for this transaction. This is the threshold value to be used when the associated program is defined with a scheduling type of parallel. An additional region is scheduled whenever the current transaction enqueue count exceeds the PARLIM value multiplied by the number of regions currently scheduled for this transaction.

The value can be a number from 0 to 32767 or 65535. PARLIM(0) indicates that any input message can cause a new region to be scheduled because the scheduling condition is always being met (the number of messages are greater than zero). If you specify PARLIM(0), you should specify a MAXRGN value to limit the number of regions that can be scheduled to process a particular transaction. PARLIM(65535) means that parallel processing is disabled and IMS allows the transaction to be scheduled in only one region at a time. PARLIM(65535) is the default.

The value specified for PARLIM applies to message processing programs (MPPs) only; it is not supported for batch message processing programs (BMPs).

If you define the transaction as serial or the program with a scheduling type as SERIAL, define PARLIM(65535).

In a shared-queues environment (when the scheduling type is PARALLEL), any PARLIM value other than 65535 causes a new region to be scheduled whenever the successful consecutive GU count exceeds the PARLIM value multiplied by the number of regions currently scheduled for this transaction. For shared-queues environments, the successful consecutive GU count is used instead of the queue count. New regions continue to be scheduled up to the maximum number of regions specified on MAXRGN.

Keyword combination rules include the following:

- MAXRGN>0 and PARLIM(65535) are mutually exclusive.
- SERIAL(Y) and PARLIM value 0 - 32767 are mutually exclusive.

#### **PGM**

Specifies the name of the program associated with the transaction.

For a CREATE TRANDESC command, the program does not need to exist until a CREATE TRAN command creates a transaction using the transaction descriptor.

#### **PLCT**

Specifies the processing limit count. This is maximum number of messages sent to the application program by IMS for processing without reloading the application program. The value must be a number from 0 through 65535. PLCT(0) means the maximum number of messages sent to the application is one and the application program is reloaded before receiving a subsequent message. PLCT(65535) means that no limit is to be placed upon the number of messages processed at a single program load. Values 1 through 65535 are eligible for quick reschedule processing. The default is 65535.

The value is used to determine how many messages an application program is allowed to process in a single scheduling cycle. When the application program requests, and receives, the number of messages indicated, any subsequent requests result in one of two actions.

- IMS indicates “no more messages exist” if any of the following conditions is true:
    - The region is not an MPP.
    - The currently scheduled mode is not CMTMODE(SNGL).
    - Equal or higher priority transactions are enqueued for the region.
- IMS might have other messages enqueued for the application program. It is the responsibility of the application program to terminate when it receives an indicator that no more messages are available. Termination of the application program makes the region it occupied available for rescheduling. This feature makes it possible for IMS to enable scheduling of higher priority transactions that entered the system while the previous transactions were in process. In addition, if any equal-priority transactions are enqueued, they become eligible for scheduling on a first-in, first-out (FIFO) basis.
- The region goes through quick reschedule and returns the next message to the application if all of the following conditions are true:
    - The region is an MPP.
    - The transaction is CMTMODE(SNGL).
    - No equal or higher transactions are enqueued.
    - Messages are still enqueued for the application program.

#### PLCTTIME

Specifies the processing limit count time. This is the amount of time (in hundredths of seconds) allowable to process a single transaction (or message). The number specifies the maximum CPU time allowed for each message to be processed in the message processing region.

Batch Message Programs (BMPs) are not affected by this setting.

The value can be a number, in hundredths of seconds, that can range from 1 to 6553500. A value of 6553500 means no time limit is placed on the application program. The default is 6553500.

If Fast Path is used this specifies, for a given transaction name, the amount of time (in hundredths of seconds) the program is allowed to process a single transaction message. The time represents real time that elapses during transaction processing (not accumulated task time). Real time is used because the input terminal is in response mode and cannot enter another transaction until the response is sent. In this case PLCT() is ignored.

The value controls application program looping. You are not required to optimize the value for program-transaction execution time. However, the time value assigned should not be less than the expected per-transaction execution time. If the scheduled application program exceeds the product of PLCTTIME() and PLCT(), the application program ends abnormally. If an IMS STIMER value of 2 is specified on the DFSMPR procedure, the region does not abend until completion of the DL/I call.

**Important:** The application program must not use STIMER timer services. IMS uses STIMER timer services to time the execution of transactions. If an

application program issues an MVS STIMER macro, it cancels the STIMER timer services set by IMS. Use the STIMERM macro instead for application program timer requests.

#### **RECOVER**

Specifies the recovery option.

- N** The transaction should not be recovered.
- Y** The transaction should be recovered during IMS emergency or normal restart. This is the default.

Keyword combination rules include the following:

- CONV(Y) and RECOVER(N) are mutually exclusive.
- FP(E) and RECOVER(N) are mutually exclusive.
- FP(P) and RECOVER(N) are mutually exclusive.
- INQ(N) and RECOVER(N) are mutually exclusive.

#### **REMOTE**

Specifies the remote option.

- N** The transaction is not remote. The transaction is local and runs on the local system. This is the default.
- Y** The transaction is remote. The transaction runs on a remote system.

Keyword combination rules include the following:

- REMOTE(Y) requires MSNAME or SIDR and SIDL.

#### **RESP**

Specifies the response-mode option.

- N** The transaction is not response mode. For terminals specifying or accepting a default of OPTIONS=TRANRESP, input should not stop after this transaction is entered. This is the default.
- Y** The transaction is response mode. The terminal from which the transaction is entered is held and prevents further input until a response is received. For terminals specifying or accepting a default of OPTIONS=TRANRESP, no additional messages are to be allowed after this transaction is entered until this transaction sends a response message back to the terminal. Response mode can be forced or negated by individual terminal definition. RESP(Y) is ignored during online processing for all terminals that do not operate in response mode.

Keyword combination rules include the following:

- FP(E) and RESP(N) are mutually exclusive.
- FP(P) and RESP(N) are mutually exclusive.

#### **SEGNO**

Specifies the segment number. This is the maximum number of application program output segments that are allowed into the message queues per Get Unique (GU) call from the application program. The value can be a number from 0 through 65535. The default is 0. If SEGNO(0) is defined, the number of segments is not checked by the online system at execution time.

#### **SEGSZ**

Specifies the segment size. This is the maximum number of bytes allowed in any one output segment. The value can be a number from 0 through 65535. The default is 0. If SEGSZ(0) is defined, the segment size is not checked by the online system at execution time.

The maximum output message segment to an LU 6.2 device is 32767. If a transaction is expected to send output to an LU 6.2 device, the SEGSIZE parameter should be no greater than 32767. However, this is not enforced during processing of the command, because IMS cannot determine the device type for the message destination until output time.

### **SERIAL**

Specifies the serial option.

- N** Messages for the transaction are not processed serially. Message processing can be processed in parallel. Messages are placed on the suspend queue after a U3303 pseudoabend. Scheduling continues until repeated failures result in the transaction being stopped with a USTOP. This is the default.
- Y** Messages for the transaction are processed serially. U3303 pseudoabends do not cause the message to be placed on the suspend queue but rather on the front of the transaction message queue, and the transaction is stopped with a USTOP. The USTOP of the transaction is removed when the transaction or the class is started with a /START or UPD TRAN command.

Keyword combination rules include the following:

- MAXRGN>0 and SERIAL(Y) are mutually exclusive.
- PARLIM value 0 - 32767 and SERIAL(Y) are mutually exclusive.

### **SIDL**

Specifies the system identification (SYSID) of the local system in a multiple-IMS system (MSC) configuration. The local system is the originating system to which responses are returned. The value can be a number from 1 to 2036. The local SYSID can be defined in any or all of the MSNAMEs or transactions.

For a local transaction where SIDL is not specified, SIDL is defined with a value of the lowest system ID when MSC is enabled on this system, or SIDL is defined with a value of 0 when MSC is not enabled on this system. The local system ID and remote system ID are the same for local transactions.

The SIDL parameter is independent of the link type (CTC, MTM, TCP/IP, VTAM) specified on the TYPE= keyword of the MSPLINK macro statement.

Keyword combination rules include the following:

- FP(E) and SIDL are mutually exclusive, unless SIDL and SIDR are specified as a pair and are equal to the local system ID of this IMS.
- FP(P) and SIDL are mutually exclusive.
- MSNAME and SIDL are mutually exclusive.
- SIDL value must be defined to this IMS.

### **SIDR**

Specifies the system identification (SYSID) of the remote system in a multiple-IMS system (MSC) configuration. The remote system is the system on which the application program executes. The value can be a number from 1 to 2036. The remote SYSID specified must also be defined for an MSNAME.

For a local transaction where SIDR is not specified, SIDR is defined to be the same value as SIDL.

The SIDR parameter is independent of the link type (CTC, MTM, TCP/IP, VTAM) specified on the TYPE= keyword of the MSPLINK macro statement.

Keyword combination rules include the following:

- FP(E) and SIDR are mutually exclusive, unless SIDL and SIDR are specified as a pair and are equal to the local system ID of this IMS.
- FP(P) and SIDR are mutually exclusive.
- MSNAME and SIDR are mutually exclusive.
- SIDR value must be defined to this IMS.

### **SPASZ**

Specifies the scratchpad area (SPA) size, in bytes, for a conversational transaction. The value can be a number from 16 and 32767.

CONV(N) and SPASZ are mutually exclusive.

### **SPATRUNC**

Specifies the scratchpad area (SPA) truncation option of a conversational transaction. This defines whether the SPA data should be truncated or preserved across a program switch to a transaction that is defined with a smaller SPA. If not specified for a conversational transaction, the value is defined to be the TRUNC parameter in DFSDCxxx.

When a conversation initially starts, and when a program is switched, the SPATRUNC option is checked and set or reset as specified. When the option is set, it remains set for the life of the conversation, or until a program switch occurs to a transaction that specifies the option is to be reset.

When a program switch occurs, the truncated data option for the new transaction is first checked, and that specification is set for the conversation and is used for the SPA inserted into the output message. If the option is not specified for the new transaction, the option currently in effect for the conversation is used.

**S** IMS preserves all of the data in the SPA, even when a program switch is made to a transaction that is defined with a smaller SPA. The transaction with the smaller SPA does not see the truncated data, but when the transaction switches to a transaction with a larger SPA, the truncated data is used.

**R** The truncated data is not preserved.

Keyword combination rules include the following:

- CONV(N) and SPATRUNC are mutually exclusive.

### **TRANSTAT**

Specifies whether transaction level statistics should be logged for message driven programs. If Y is specified, transaction level statistics are written to the log in a X'56FA' log record.

**N** Transaction level statistics should not be logged.

**Y** Transaction level statistics should be logged.

The TRANSTAT keyword is optional. If a value is not specified for the TRANSTAT keyword, the system default is used. The system default for the transaction level statistics parameter is set with the TRANSTAT parameter in

the DFSDFxxx PROCLIB member. If a value is not specified for the TRANSTAT parameter, the system default is set to N. The TRANSTAT keyword on the CREATE TRAN or CREATE TRANDESC command gives the user the ability to override the system default when creating a transaction or transaction descriptor.

#### **WFI**

Specifies the wait-for input option. This attribute does not apply to Fast Path transactions, which always behave as wait-for-input transactions.

**N** This is not a wait-for-input transaction. This is the default.

**Y** This is a wait-for-input transaction. A message processing or batch processing application program that processes WFI transactions is scheduled and invoked normally. If the transaction to be processed is defined as WFI, the program is allowed to remain in main storage after it has processed the available input messages. The QC status code (no more messages) is returned to the program if the processing limit count (PLCT) is reached; a command is entered to change the status of the scheduled transaction, database, program, or class; commands relating to the databases used by the transaction are entered, or IMS is terminated with a checkpoint shutdown.

CMTMODE(MULT) and WFI(Y) are mutually exclusive.

### **Usage notes**

Descriptors exist for the life of the IMS unless they are deleted using a DELETE command. Descriptors are recoverable across an IMS warm start or emergency restart. Descriptors are lost if IMS is cold started, unless cold start imports definitions that were exported while IMS was up.

The CREATE TRANDESC command is similar to local online change (using a /MODIFY command) or global online change (using an INITIATE OLC command) for transaction resources in the MODBLKS data set, except that transactions are added dynamically.

The CREATE TRANDESC command can be issued only through the Operations Manager API. These commands apply to DB/DC and DCCTL systems.

These commands are not valid on the XRF alternate, RSR tracker, or FDBR region. The CREATE command is not valid if online change for MODBLKS is enabled (DFSDFxxx or DFSCGxxx defined with MODBLKS=OLC or MODBLKS not defined).

### **Output fields**

The following table shows the CREATE TRANDESC output fields. The columns in the table are as follows:

#### **Short label**

Contains the short label generated in the XML output.

#### **Keyword**

Identifies keyword on the command that caused the field to be generated. N/A appears for output fields that are always returned. The value *error* appears for output fields that are returned only in case of an error.

## Meaning

Provides a brief description of the output field.

Table 173. Output fields for the CREATE TRANDESC command

Short label	Keyword	Meaning
CC	N/A	Completion code.
CCTXT	<i>error</i>	Completion code text that briefly explains the meaning of the non-zero completion code.
DESC	TRANDESC	Transaction descriptor name.
MBR	N/A	IMSplex member that built the output line.
OLDDEF	TRANDESC	Old default descriptor name, if this descriptor is made the default by specifying DEFAULT(Y). The old default descriptor is no longer the default.

## Return, reason, and completion codes

The following table includes the return and reason codes and a brief explanation of the codes. The return or reason code returned for the command might also indicate an error from a CSL request.

Table 174. Return and reason codes for the CREATE TRANDESC command

Return code	Reason code	Meaning
X'00000000'	X'00000000'	Command completed successfully. The command output contains a line for each descriptor, accompanied by its completion code. See the completion code table for details.
X'00000008'	X'0000204C'	The class value is invalid.
X'00000008'	X'00002050'	The current priority CPRI value is invalid.
X'00000008'	X'00002054'	The limit count LCT value is invalid.
X'00000008'	X'00002058'	The limit priority LPRI is invalid.
X'00000008'	X'0000205C'	The maximum region MAXRGN is invalid.
X'00000008'	X'00002060'	The normal priority NPRI value is invalid.
X'00000008'	X'00002064'	The parallel limit count PARLIM value is invalid.
X'00000008'	X'00002068'	The processing limit count PLCT is invalid.
X'00000008'	X'0000206C'	The segment number SEGNO value is invalid.
X'00000008'	X'00002070'	The segment size SEGSZ value is invalid.
X'00000008'	X'00002100'	CMTMODE(MULT) mutually exclusive with WFI(Y). Either both keywords were explicitly specified, or one keyword was explicitly specified and the other attribute was obtained from the model.
X'00000008'	X'00002101'	CONV(Y) mutually exclusive with CMTMODE(MULT). Either both keywords were explicitly specified, or one keyword was explicitly specified and the other attribute was obtained from the model.
X'00000008'	X'00002102'	CONV(Y) mutually exclusive with INQ(Y). Either both keywords were explicitly specified, or one keyword was explicitly specified and the other attribute was obtained from the model.

Table 174. Return and reason codes for the CREATE TRANDESC command (continued)

Return code	Reason code	Meaning
X'00000008'	X'00002103'	CONV(N) mutually exclusive with SPASZ>0 or SPATRUNC. Either both keywords were explicitly specified, or one keyword was explicitly specified and the other attribute was obtained from the model.
X'00000008'	X'00002104'	CONV(Y) mutually exclusive with RECOVER(N). Either both keywords were explicitly specified, or one keyword was explicitly specified and the other attribute was obtained from the model.
X'00000008'	X'00002105'	CONV(Y) requires SPASZ and SPATRUNC. Either SPASZ was not explicitly specified or defined in the model, or SPATRUNC was not explicitly specified or defined in the model. CONV(Y) was either explicitly specified or defined in the model.
X'00000008'	X'00002108'	Invalid EDITRTN name. EDITRTN was specified explicitly or obtained from the model.
X'00000008'	X'00002109'	Maximum 255 edit routines exceeded. EDITRTN was specified explicitly or obtained from the model.
X'00000008'	X'0000210C'	FP(E) mutually exclusive with EDITRTN. Either both keywords were explicitly specified, or one keyword was explicitly specified and the other attribute was obtained from the model.
X'00000008'	X'0000210E'	FP(E) or FP(P) mutually exclusive with MSC keyword, either MSNAME or SIDR and SIDL. Either both keywords were explicitly specified, or one keyword was explicitly specified and the other attribute was obtained from the model.
X'00000008'	X'0000210F'	FP(E) or FP(P) mutually exclusive with MSGTYPE(MULTSEG). Either both keywords were explicitly specified, or one keyword was explicitly specified and the other attribute was obtained from the model.
X'00000008'	X'00002110'	FP(N) mutually exclusive with EMHBSZ > 0. Either both keywords were explicitly specified, or one keyword was explicitly specified and the other attribute was obtained from the model.
X'00000008'	X'00002111'	FP(E) or FP(P) mutually exclusive with RECOVER(N). Either both keywords were explicitly specified, or one keyword was explicitly specified and the other attribute was obtained from the model.
X'00000008'	X'00002112'	FP(E) or FP(P) mutually exclusive with RESP(N). Either both keywords were explicitly specified, or one keyword was explicitly specified and the other attribute was obtained from the model.
X'00000008'	X'00002116'	INQ(N) mutually exclusive with RECOVER(N). Either both keywords were explicitly specified, or one keyword was explicitly specified and the other attribute was obtained from the model.
X'00000008'	X'00002117'	Invalid LIKE descriptor name.
X'00000008'	X'00002118'	Invalid LIKE resource name.

Table 174. Return and reason codes for the CREATE TRANDESC command (continued)

Return code	Reason code	Meaning
X'00000008'	X'00002119'	MSC keyword MSNAME or SIDR and SIDL are mutually exclusive with application program defined as Fast Path exclusive (FP(E)) associated with this transaction.
X'00000008'	X'0000211A'	Invalid MSNAME name.
X'00000008'	X'0000211B'	MSNAME mutually exclusive with SIDR and SIDL.
X'00000008'	X'0000211D'	MAXRGN>0 mutually exclusive with PARLIM(65535). Either both keywords were explicitly specified, or one keyword was explicitly specified and the other attribute was obtained from the model.
X'00000008'	X'0000211E'	MAXRGN>0 mutually exclusive with SERIAL(Y). Either keywords were explicitly specified, or one keyword was explicitly specified and the other attribute was obtained from the model.
X'00000008'	X'00002120'	PGM() definition missing. PGM must be defined for local transactions either explicitly or in the model.
X'00000008'	X'00002121'	PARLIM value mutually exclusive with SERIAL(Y). Either both keywords were explicitly specified, or one keyword was explicitly specified and the other attribute was obtained from the model.
X'00000008'	X'00002123'	Invalid PGM name. PGM was specified explicitly, or obtained from the model. If the IMS-defined descriptor DFSDSTR1 was used, the command failed because DFSDSTR1 defines no program name.
X'00000008'	X'00002125'	REMOTE(Y) requires MSC keyword MSNAME or SIDR and SIDL. Either MSNAME or SIDR and SIDL were not explicitly specified or defined in the model. REMOTE(Y) was either explicitly specified or defined in the model.
X'00000008'	X'00002126'	Invalid SIDL value.
X'00000008'	X'00002127'	SIDL/SIDR must be specified as a pair. Either SIDL was specified alone or SIDR was specified alone.
X'00000008'	X'00002128'	Invalid SIDR value.
X'00000008'	X'00002133'	Multiple name parameters were specified with DEFAULT(Y). Only one descriptor can be the default at one time.
X'0000000C'	X'00003000'	Command was successful for some descriptors but failed for others. The command output contains a line for each descriptor, accompanied by its completion code. See the completion code table for details.
X'0000000C'	X'00003004'	Command was not successful for any of the descriptors. The command output contains a line for each descriptor, accompanied by its completion code. See the completion code table for details.
X'00000010'	X'0000400C'	Command is not valid on the XRF alternate.
X'00000010'	X'00004014'	Command is not valid on the RSR tracker.
X'00000010'	X'00004024'	No Fast Path defined, so FP(E), FP(P), or EMHBSZ > 0 is invalid.

Table 174. Return and reason codes for the CREATE TRANDESC command (continued)

Return code	Reason code	Meaning
X'00000010'	X'00004120'	Online change phase is in progress.
X'00000010'	X'00004204'	Model is quiesced. Cannot quiesce model.
X'00000010'	X'00004300'	Command is not allowed because online change for MODBLKS is enabled (DFSDFxxx or DFSCGxxx defined with MODBLKS=OLC, or MODBLKS not defined).
X'00000010'	X'00004310'	Storage could not be obtained for the Transaction Input edit routine table. A cold start is required to fix this error.
X'00000010'	X'00004314'	The Transaction Input edit routine could not be loaded.
X'00000010'	X'00004318'	A new Transaction Input edit routine could not be added. The maximum of 255 routines has already been reached.
X'00000014'	X'00005004'	DFSOCMD response buffer could not be obtained.
X'00000014'	X'00005008'	DFSPOOL storage could not be obtained.
X'00000014'	X'0000500C'	AWE could not be obtained.
X'00000014'	X'00005010'	Unable to obtain latch.
X'00000014'	X'00005114'	Resource or descriptor that is specified as a model is in the process of being imported from the change list in the IMSRSC repository or was not imported successfully from the change list at the end of IMS restart. This resource or descriptor cannot be referenced as a model on a CREATE command until it is successfully imported from the repository.

Errors unique to the processing of this command are returned as completion codes. The following table includes an explanation of the completion codes.

Table 175. Completion codes for the CREATE TRANDESC command

Completion code	Completion code text	Meaning
0		Command completed successfully for transaction descriptor.
11	RESOURCE ALREADY EXISTS	Transaction descriptor already exists.
17	ANOTHER CMD IN PROGRESS	Another command (such as DELETE PGM or UPDATE PGM) is in progress for the program referenced by this transaction.
36	FP=E/FP=N PGM CONFLICT	Transaction to be created as Fast Path exclusive FP(E) conflicts with program already defined as non-Fast Path FP(N).
3E	FP=N/FP=E PGM CONFLICT	Transaction to be created as non-Fast Path FP(N) conflicts with program already defined as Fast Path exclusive FP(E).

Table 175. Completion codes for the CREATE TRANDESC command (continued)

Completion code	Completion code text	Meaning
3F	FP=P/BMPTYPE=Y CONFLICT	Transaction to be created as Fast Path potential FP(P) conflicts with the program defined as batch BMPTYPE(Y).
5F	INVALID CHARACTERS IN NAME	Name is invalid because it contains an invalid character.
61	DFSBCB STORAGE ERROR	Request to get BCB storage for SMB control block failed.
8A	WILDCARD PARAMETER NOT SUPPORTED	The CREATE command does not support wildcard parameters. You must explicitly specify the names of the descriptors you want to create.
93	RESERVED NAME	Name is reserved. For example, name starts with DFS (except for DFSSAMxx or DFSIVPxx), or is an IMS reserved name such as BASICEDT or WTOR.

## Examples

The following are examples of the CREATE TRANDESC command:

### Example 1 for CREATE TRANDESC command

TSO SPOC input:

```
CRE TRANDESC NAME(CONVDESC) LIKE(RSC(CDEBTRNA))
SET(CONV(Y),SPASZ(128),SPATRUNC(R),PGM(DFSSAM04),DEFAULT(Y))
```

TSO SPOC output:

```
DescName MbrName CC 01dDefault
CONVDESC IMS1 0 DFSDSTR1
```

OM API input:

```
CMD(CRE TRANDESC NAME(CONVDESC) LIKE(RSC(CDEBTRNA))
SET(CONV(Y),SPASZ(128),SPATRUNC(R),PGM(DFSSAM04),DEFAULT(Y)))
```

OM API output:

```
<imsout>
<ctl>
<omname>OM10M </omname>
<omvsn>1.3.0</omvsn>
<xmlvsn>20 </xmlvsn>
<stime>2006.307 18:52:58.805900</stime>
<stotime>2006.307 18:52:58.806677</stotime>
<staseq>BFA6C8765828C44A</staseq>
<stoseq>BFA6C8765859584A</stoseq>
<rqsttkn1>USRT011 10105258</rqsttkn1>
<rc>00000000</rc>
<rsn>00000000</rsn>
</ctl>
<cmd>
<master>IMS1 </master>
<userid>USRT011 </userid>
<verb>CRE </verb>
```

```

<kwd>TRANDESC          </kwd>
<input>CRE TRANDESC NAME(CONVDESC) LIKE(RSC(CDEBTRNA))
  SET(CONV(Y),SPASZ(128),SPATRUNC(R),PGM(DFSSAM04),DEFAULT(Y)) </input>
</cmd>
<cmdrsphdr>
<hdr s1b1="DESC" l1b1="DescName" scope="LCL" sort="a" key="1"
  scroll="no" len="8" dtype="CHAR" align="left" />
<hdr s1b1="MBR" l1b1="MbrName" scope="LCL" sort="a" key="2" scroll="no"
  len="8" dtype="CHAR" align="left" />
<hdr s1b1="CC" l1b1="CC" scope="LCL" sort="n" key="0" scroll="yes"
  len="4" dtype="INT" align="right" skipb="no" />
<hdr s1b1="CCTXT" l1b1="CCText" scope="LCL" sort="n" key="0"
  scroll="yes" len="*" dtype="CHAR" skipb="yes" align="left" />
<hdr s1b1="OLDDEF" l1b1="OldDefault" scope="LCL" sort="a" key="2"
  scroll="no" len="8" dtype="CHAR" align="left" />
</cmdrsphdr>
<cmdrspdata>
<rsp>DESC(CONVDESC) MBR(IMS1) CC( 0) OLDDEF(DFSDSTR1) </rsp>
</cmdrspdata>
</imsout>

```

**Explanation:** A CREATE TRANDESC command is issued to create a transaction descriptor for conversational transactions. Descriptor CONVDESC takes the same attributes as transaction CDEBTRNA except for the conversation attribute, SPA size, SPA truncate option, and program name. It is also made the default descriptor, so that if no LIKE keyword is specified on a CREATE TRAN or CREATE TRANDESC command, the transaction or descriptor is defined with the attributes in descriptor CONVDESC.

**Related concepts:**

 [How to interpret CSL request return and reason codes \(System Programming APIs\)](#)

**Related reference:**

Chapter 2, "Command keywords and their synonyms," on page 63

 [Commit Verify exit routine \(Exit Routines\)](#)

## Chapter 20. /DBDUMP command

Use the /DBDUMP command to prevent transactions or programs from updating DL/I databases. It also can be used to dump all MSDBs to the MSDB dump data set. /DBDUMP does not apply to DEDBs.

Subsections:

- "Environment"
- "Syntax"
- "Keywords"
- "Usage notes" on page 341
- "Equivalent IMS type-2 commands" on page 343
- "Examples" on page 343

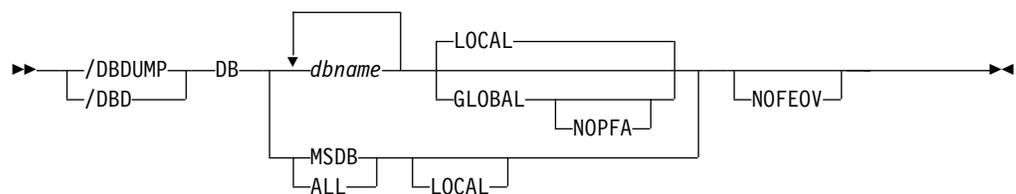
### Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

Table 176. Valid environments for the /DBDUMP command and keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
/DBDUMP	X	X	
DB	X	X	
GLOBAL	X	X	
LOCAL	X	X	
NOFEOV	X	X	
NOPFA	X	X	

### Syntax



### Keywords

The following keywords are valid for the /DBDUMP command:

- DB** Specifies the databases to which the /DBDUMP command applies. When the /DBDUMP command is entered, the message processing regions using the specified databases are terminated at the conclusion of processing their current transactions, in preparation to close the database and enable it to be opened for input only.

If a DL/I database specified in the command is being used by a batch message processing region, an error message is returned to the master terminal. When this message is issued, the command is ignored for the database named in the message; processing continues for the other databases specified in the command. The master terminal operator must wait until the batch message processing concludes processing before reentering the command.

As the message processing regions terminate programs, the data sets of the named databases in the command are closed. The IMS log switches to the next OLDS. This switch to the next OLDS is marked as a recovery point for log archiving purposes. IMS issues a simple checkpoint. The scheduling of transactions is then resumed, although no transactions will be allowed to update the specified databases. Programs with update intent will be scheduled, but update calls to the database will result in a 3303 pseudoabend or a BA status if the INIT call was issued.

/DBDUMP can be used to dump all the MSDBs to the MSDB dump data set by specifying the reserved parameter MSDB with the DB keyword when entering the /DBDUMP DB command or by entering the /DBDUMP DB ALL command. The MSDBs dumped to the MSDB dump data set can be used as input to the MSDB Dump Recovery utility. A specific MSDB cannot be a parameter of the DB keyword.

The /START DB command resets the effect of the /DBDUMP command. The /START command is not required for MSDBs, because the data for these databases resides in processor storage, and the databases are never closed.

For DBCTL, when CCTL schedules a PSB, the DBCTL thread SCHED request defines the thread as LONG or SHORT. If the database is currently scheduled to a LONG thread, the command is rejected; otherwise, the thread is allowed to "complete" before the database is acted upon. This results in either a commit point or transaction termination.

## **GLOBAL**

Applies when an IRLM is active and specifies that the command applies to all online subsystems sharing the database. The /DBDUMP command with the GLOBAL keyword puts the database in read status and prevents transactions from updating the database in all online subsystems that share the database.

The /DBDUMP GLOBAL command is processed by the IMS system where the command was initiated. The systems will process the command locally and then request IRLM NOTIFY to route and process the command on sharing IMS systems.

If global database status is maintained, the global status maintained in RM is also updated. The global status is set to STOUPDS.

If the command is entered from OM API, the global status is updated by the command master IMS. If the command is not entered from OM API, the IMS that initiated the GLOBAL command updates the global status in RM.

If global status in RM is successfully updated, message DFS0988I for RSRCTYPE=DB is issued. If global status is not successfully updated, message DFS3308I is issued, indicating RM failure, and no command response lines are generated. Any RM error is traced to the OCMD trace table. Users can issue a QRY DB STATUS(GLOBAL) command to set the global status of the resources in RM.

The X'4C' log record for databases is updated to include both the global status and global command time stamp.

The GLOBAL keyword is mutually exclusive with the ALL parameter or the MSDB parameter and causes the command to be rejected if both parameters are specified. The GLOBAL keyword requires that IRLM be active and will cause the command to be rejected if IRLM is not active.

If the command with GLOBAL is entered from OM API, the command master IMS becomes the initiating system. The command IMS will process the command locally first and then make DBRC calls to update the RECON with GLOBAL status. It will also request IRLM NOTIFY to route and process the command on sharing IMS systems.

Messages produced on the NOTIFIED systems will appear only on the system console and will not be routed back to the OM API which originally entered the command.

If the command is routed to multiple IMS systems, the non-master IMS systems to which OM routes the command, will reject the command with the return and reason codes shown in the following table.

*Table 177. Return and reason code for the GLOBAL keyword issued from the OM API*

Return code	Reason code	Meaning
X'00000004'	X'00001000'	The command contained the GLOBAL keyword and was routed to more than one IMS system in the IMSPLEX. The non-master IMS systems will reject this command when OM routes the command to them. The master IMS system will process this command and use IRLM NOTIFY to route and process the command on the non-master IMS systems. See the discussion under the GLOBAL keyword.

#### **LOCAL**

Specifies that the command only applies to the subsystem in which the command is entered. This command does not affect any other subsystem sharing the database. The LOCAL keyword can be used to restrict concurrent updates. LOCAL is the default.

#### **NOFEOV**

Specifies that there is no forced end of volume, so that the IMS log does not switch to the next OLDS. If NOFEOV is specified without the MSDB keyword, a simple checkpoint is not taken.

#### **NOPFA**

Specifies that the call to DBRC that sets the Read Only flag in the RECON data set for the database or partition is to be skipped. You can use this keyword when you need to authorize the database for update after the command has been processed. By using this keyword, DBRC does not prevent authorizations for update for the database or partition. NOPFA can be specified only with the GLOBAL keyword.

### **Usage notes**

The /DBDUMP command can be used on HALDB databases.

In an IMSplex, the output of the /DBD command is changed when the command is entered through the OM API. In this case, the DFS058I message is not returned to OM. The command response returned to OM contains one or more of the following messages as appropriate to the database type and the command completion.

This command can be issued to an IMSplex using the Batch SPOC utility.

Full Function Database messages: DFS132, DFS160, DFS216, DFS0488I, DFS1407, DFS2026, DFS3318I, DFS3320I, DFS3321I, DFS3325I, DFS3462I, DFS3463I, DFS3466I

When you enter this command, the database name can be an existing non-HALDB, a HALDB master, or a HALDB partition. A command against a HALDB partition operates exactly like a command against a non-HALDB with the exception of the /START DATABASE and the UPDATE DB START(ACCESS) command. A HALDB partition is not allocated during the command unless it was previously authorized but not allocated, the OPEN keyword was specified, or the partition has EEQEs. The partition is allocated at first reference.

For HALDB databases, IMS tracks partition statuses and master database statuses separately. For example, a partition can be stopped, but its master database can be started. Alternatively, the partition can be started, but its master database can be stopped. Before opening, authorizing, or scheduling a partition, IMS always examines the status of the partition and the master database. If either the partition or the master database has a status that prevents the action, IMS does not perform the action.

Each partition has the access limitations of both itself and its master database. For example, if the master database has an access intent of read (READ) and one of its partitions has an access intent of update (UPD), the partition cannot be updated. Alternatively, if the master database has an access intent of update (UPD) and one of its partitions has an access intent of read (READ), the partition cannot be updated. Similar considerations apply to other statuses that affect access limitations, such as being stopped or locked.

**Exception:** If the HALDB master database has update access (UPD), the partitions can have an access intent of exclusive (EXCL), exceeding the access of the master.

Commands that are issued with a partition name affect only the status of the partition. Commands that are issued against the master database affect only the status of the master database. Therefore, a start of a master database does not update the status of its partitions. If the partitions are stopped, they remain stopped. When a HALDB partition is explicitly stopped, it must be explicitly started again. The type-1 commands with the keyword ALL, type-2 commands with NAME(\*), and commands against a HALDB master do not change the STOPPED (shown as STOACC, STOSCHD, or STOUPDS on QUERY DB) and LOCKED indicators in each HALDB partition.

When the command target is a HALDB master, processing acts on all HALDB partitions. For example, if the IMS command is UPDATE DB STOP(ACCESS) on the HALDB master, all of the HALDB partitions are closed, deallocated, and deauthorized. However, the stopped status is only set in the master database. If a QUERY DB command is issued, only the HALDB master displays a status of STOACC (each HALDB partition does not display STOACC unless it was itself stopped). If a UPDATE DB STOP(ACCESS) command was issued against a HALDB master, the display output of a /DISPLAY DB command shows the HALDB master (as STOPPED), but does not display the status of the partitions.

**Restrictions:**

- The /DBDUMP DB command cannot be processed against a HALDB partition on an IMS system while HALDB Online Reorganization (OLR) is running against that partition on the same IMS system.

- The /DBDUMP DB command cannot be issued against a HALDB master while OLR is reorganizing any of its partitions.
- While the database is being quiesced, this command cannot be processed successfully.

The /DBDUMP DB command is not allowed for a database that is marked bad with the NOTINIT-48-REPOCHGLIST reason code because the IMS change list processing is not complete for the database or the change list processing failed.

## Equivalent IMS type-2 commands

The following table shows variations of the /DBDUMP command and the IMS type-2 commands that perform similar functions.

Table 178. Type-2 equivalents for the /DBDUMP command.

Task	/DBDUMP command	Similar IMS type-2 command
Stops updates to a database.	/DBDUMP DB <i>dbname</i>	UPDATE DB NAME( <i>dbname</i> ) STOP(UPDATES) OPTION(FEOV) <sup>1</sup>

<sup>1</sup> This command does not automatically issue checkpoints unless OPTION(FEOV) is specified.

## Examples

The following are examples of the /DBDUMP command:

### Example 1 for /DBDUMP command

Entry ET:

```
/DBDUMP DATABASE PAYROLL
```

Response ET:

```
DFS058I (time stamp) DBDUMP COMMAND IN PROGRESS
```

Explanation: Currently executing application programs are being terminated. When the termination completes, the databases are stopped for update and the output log is switched to the next OLDS.

Response ET:

```
DFS0488I DBD COMMAND COMPLETED.  
DBN=PAYROLL RC=0  
DFS3257I ONLINE LOG NOW SWITCHED FROM DFSOLP( ) TO DFSOLP( )  
DFS994I *CHKPT 82080/111213**SIMPLE*
```

Explanation: The new OLDS is used to record a simple checkpoint at 111213 (time) on 82080 (Julian date). The checkpoint number is 82080/111213. All /DBDUMP command functions are complete. The /START DATABASE command must be used to start the database after the dump job completes.

### Example 2 for /DBDUMP command

Entry ET:

```
/DBDUMP DATABASE MSDB
```

Response ET:

```
DFS058I (time stamp) DBDUMP COMMAND IN PROGRESS
```

Explanation: All MSDBs are dumped to the MSDB dump data set because MSDB was specified as the parameter of the database keyword.

Response ET:

```
DFS994I  CHKPT 82069/123624**SIMPLE*
```

Explanation: A simple checkpoint is recorded on the new system log at 123624 (time) on 82069 (Julian date). The checkpoint number is 82069/123624. All MSDBs are dumped.

**Related concepts:**

 Maintaining global information for databases, DEDB areas, and transactions (System Administration)

**Related reference:**

 UPDATE DB command (Commands)

## Chapter 21. /DBRECOVERY command

Use the /DBRECOVERY command to prevent transactions or programs from accessing DL/I databases, DEDBs, or DEDB areas.

Subsections:

- "Environment"
- "Syntax"
- "Keywords" on page 346
- "Usage notes" on page 349
- "Equivalent IMS type-2 commands" on page 351
- "Examples" on page 351

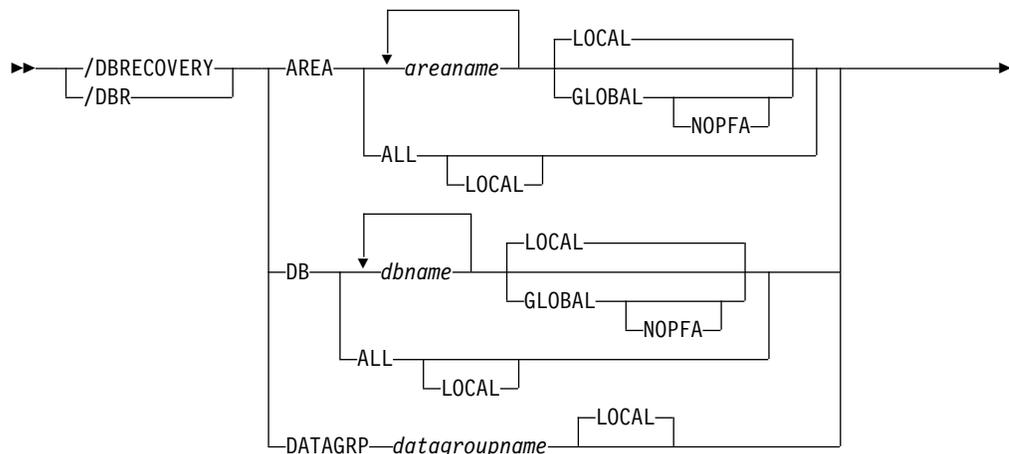
### Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

Table 179. Valid environments for the /DBRECOVERY command and keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
/DBRECOVERY	X	X	
AREA	X	X	
DB	X	X	
DATAGRP	X	X	
GLOBAL	X	X	
LOCAL	X	X	
NOFEOV	X	X	
NOPFA	X	X	

### Syntax



## Keywords

The following keywords are valid for the /DBRECOVERY command:

### AREA

Specifies the DEDB areas (but not DEDBs) to be processed by the /DBRECOVERY command. The /DBRECOVERY AREA command stops the accessing and updating of specified DEDB areas and closes them.

The /START AREA command can be used to reopen and reallocate the areas closed by /DBRECOVERY AREA command.

The /DBRECOVERY command for Fast Path Virtual Storage Option (VSO) areas in a z/OS data space removes the areas from the data space and forces updates to be written back to DASD.

While the database is being quiesced, this command cannot be processed successfully.

### DATAGRP

Specifies a group of DL/I databases and Fast Path DEDBs to be processed by the /DBRECOVERY command. Use the INIT.DBDSGRP command with the GRPNAME and DBGRP keywords to define the data group in the RECON data set.

DATAGRP is valid on active and RSR tracking subsystems.

**Recommendation:** Although you can use DBDS groups as well as database groups for this command, you should use database groups whenever possible to eliminate the overhead of converting the DBDS group to a database group.

- DB** Specifies the DL/I databases and Fast Path DEDBs (and inclusive areas) to be processed by the /DBRECOVERY command. MSDBs cannot be specified with /DBRECOVERY. The /DBRECOVERY DB command allows scheduling of transactions or programs that use the database to continue after successful completion of the command. However, calls to DEDB databases will receive an FH status code. If the application program is prepared to accept status codes regarding data unavailability through the INIT STATUS GROUPA or GROUPB call, calls to DL/I databases will result in either a 3303 pseudoabend, a BA, or BB status code.

When the /DBRECOVERY DB command is entered, the data sets of the databases named in the command are closed. The /DBRECOVERY DB command dynamically deallocates the IMS databases. For DEDBs, the randomizer is unloaded and removed from storage. You can use the UPD DB STOP(Access) OPTION(NORAND) command to prevent the randomizer from being unloaded.

If a DL/I database is in use by a batch message processing region when the /DBRECOVERY command is issued, a DFS0565I message is returned to the master terminal, the command is ignored for the database named in the message, and processing continues for any other databases specified in the command. The master terminal operator must wait until the BMP ends before reissuing the /DBRECOVERY command to close the databases named in any DFS0565I messages. DEDB databases will accept the /DBRECOVERY command

while actively being used by a batch message processing region. However, all further calls to the database will receive an FH status code.

The region identified in the DFS0565I message can also be an MPP region. The error message can be issued in order to break a potential deadlock condition between Online Change, a /DBR, or /STA command against a database being used by the MPP, and an application running in the MPP issuing an ICMD or CMD call.

The /START DB command is used to reallocate the databases closed by the /DBRECOVERY DB command except for HALDB partitions. A HALDB partition can be reallocated if the partition has EEQEs, the partition was previously authorized but not allocated, or the OPEN keyword on the /START DB command was used. HALDB partitions are allocated at first reference. For DEDBs, a /START DB command also causes a reload of the randomizer.

For DBCTL, when CCTL schedules a PSB, the DBCTL thread SCHED request defines the thread as LONG or SHORT. If the database is currently scheduled to a LONG thread, the command is rejected; otherwise, the thread is allowed to “complete” before the database is acted upon. This results in either a commit point or transaction termination.

On an RSR tracking subsystem, /DBRECOVERY DB for a DEDB is not permitted.

While the database is being quiesced, this command cannot be processed successfully.

The /DBRECOVERY DB command is not allowed for a database that is marked bad with the NOTINIT-48-REPOCHGLIST reason code because the IMS change list processing is not complete for the database or the change list processing failed.

## **GLOBAL**

Applies when the database is registered to DBRC and specifies that the command applies to all sharing online subsystems. You can use the /DBRECOVERY GLOBAL command to close the database so that it can be taken offline when this command successfully completes in all online subsystems. If the database is registered in the RECON data set, the prohibit authorization flag will be set (PROHIBIT AUTH = ON), and DBRC will prevent any further authorization of the database. This is equivalent to the CHANGE.DB DBD(*xxx*) NOAUTH DBRC command.

The /DBRECOVERY GLOBAL command is processed by the IMS system where the command was initiated. This system will process the command locally and then request IRLM NOTIFY to route and process the command on sharing IMS systems.

If global DB or AREA status is maintained, the global status maintained in RM is also updated. The global status is set to STOACC.

If the command is entered from OM API, the global status is updated by the command master IMS. If the command is not entered from OM API, the IMS that initiated the GLOBAL command updates the global status in RM.

If global status in RM is successfully updated, message DFS0988I for RSRCTYPE=DB is issued. If global status is not successfully updated, message DFS3308I is issued, indicating RM failure, and no command response lines are generated. Any RM error is traced to the OCMD trace table. Users can issue a QRY DB STATUS(GLOBAL) or QUERY AREA STATUS(GLOBAL) to obtain the global status of the resources in RM.

The X'4C' log record for databases is updated to include both global status and global command time stamp. A new log record for DEDB areas, X'594C', includes both global status and global command time stamp.

The GLOBAL keyword and the ALL parameter are mutually exclusive. The /DBRECOVERY command will be rejected if both GLOBAL and ALL are specified. The IRLM must be active when the GLOBAL keyword is used. If IRLM is not active, the command is rejected.

If the command with GLOBAL is entered from OM API, the command master IMS becomes the initiating system. The command IMS will process the command locally first and then make DBRC calls to update the RECON with GLOBAL status. It will also request IRLM NOTIFY to route and process the command on sharing IMS systems.

Messages produced on the NOTIFIED systems will appear only on the system console and will not be routed back to the OM API that originally entered the command.

If the command is routed to multiple IMS systems, the non-master IMS systems to which OM routes the command, will reject the command with the return and reason codes shown in the following table.

*Table 180. Return and reason code for GLOBAL keyword issued from the OM API*

Return code	Reason code	Meaning
X'00000004'	X'00001000'	The command contained the GLOBAL keyword and was routed to more than one IMS system in the IMSplex. The non-master IMS systems will reject this command when OM routes the command to them. The master IMS system will process this command and use IRLM NOTIFY to route and process the command on the non-master IMS systems. See the discussion under the GLOBAL keyword.

**Restriction:** When the GLOBAL keyword is specified, the command is performed for the named database only. If the command is to be performed for a HIDAM database and its index, both must be named.

The GLOBAL keyword is not supported for an RSR tracking subsystem.

The GLOBAL keyword cannot be combined with the DATAGRP keyword.

#### **LOCAL**

Specifies that the command applies only to the subsystem in which the command is entered. This command does not affect any other subsystem sharing the database or area. Use the LOCAL keyword to increase usability for the other system sharing the database or area. LOCAL is the default.

#### **NOFEOV**

Specifies that there is no forced end of volume, so that the log does not switch to the next OLDS and simple checkpoint is not taken.

The NOFEOV keyword is the default for an RSR tracking subsystem so that the OLDS is not switched, nor is a simple checkpoint taken.

#### **NOPFA**

Specifies to skip the call to DBRC that sets the Prohibit Authorization flag in the RECON data set for the database or area. You can use this keyword when

you need to authorize the database for use after it is offline, for example, for offline utilities. By using this keyword, DBRC does not prevent further authorizations for the database or area.

NOPFA can be specified only with the GLOBAL keyword.

## Usage notes

The /DBRECOVERY command can be used on HALDB databases.

Use of the ALL parameter indicates that the command applies to all databases, even those not opened.

The /DBRECOVERY command closes and deallocates the databases, and deauthorizes them with DBRC. Once the database or area referenced by /DBRECOVERY is closed, the IMS log switches to the next OLDS (unless you specify the NOFEOV keyword). This switch to the next OLDS is marked as a recovery point for log archiving purposes. IMS issues a simple checkpoint.

The /DBRECOVERY command deletes the randomizer routine from memory.

/DBRECOVERY is supported on an RSR tracking subsystem, but only for a database readiness level (DLT) subsystem. The /DBRECOVERY command is used on an RSR tracking subsystem to take shadow areas and databases offline for image copy and recovery. The /DBRECOVERY command also stops online forward recovery (OFR) if it is in progress for the specified database or area.

IMS databases, DEDBs, and DEDB areas closed by the master terminal operator with the /DBRECOVERY command can be used as input to the offline IMS Database Recovery utility. The Database Recovery utility runs in a batch region.

If the /DBRECOVERY command does not deallocate a data set because a VSAM data set hardware error occurred, use the VARY *nnn*,OFFLINE,FORCE command to deallocate the data set (where *nnn* is the number of the device).

In an IMSplex, the output of the /DBR command is different when the command is entered through the OM API. In this case, the DFS058I message is not returned to OM. The command response returned to OM contains one or more of the following messages as appropriate to the database type and the command completion.

- For /DBR AREA commands that specify GLOBAL, only the command master returns the asynchronous messages to OM. When a command is processed with the LOCAL keyword, all IMS systems are able to return the asynchronous messages to OM.
- Full-function database messages for /DBR DB: DFS132, DFS160, DFS216, DFS0488I, DFS0565I, DFS1407, DFS2026, DFS3318I, DFS3320I, DFS3321I, DFS3464I, DFS3466I.
- Fast Path database messages for /DBR DB: DFS140I, DFS0666I, DFS3062I, DFS3320I.
- Fast Path database messages for /DBR AREA: DFS140I, DFS0488I, DFS0666I, DFS1407I, DFS3060I, DFS3062I, DFS3320I, DFS3325I, DFS3342I, DFS3720I.

This command can be issued to an IMSplex using the Batch SPOC utility.

When you enter this command, the database name can be an existing non-HALDB, a HALDB master, or a HALDB partition. A command against a HALDB partition operates exactly like a command against a non-HALDB except for the /START DB and the UPDATE DB START(ACCESS) commands. A HALDB partition is not allocated during the command unless it was previously authorized but not allocated, the OPEN keyword was specified, or the partition has EEQEs. The partition is allocated at first reference.

For HALDB databases, IMS tracks partition statuses and master database statuses separately. For example, a partition can be stopped, but its master database can be started. Alternatively, the partition can be started, but its master database can be stopped. Before opening, authorizing, or scheduling a partition, IMS always examines the status of the partition and the master database. If either the partition or the master database has a status that prevents the action, IMS does not perform the action.

Each partition has the access limitations of both itself and its master database. For example, if the master database has an access intent of read (READ) and one of its partitions has an access intent of update (UPD), the partition cannot be updated. Alternatively, if the master database has an access intent of update (UPD) and one of its partitions has an access intent of read (READ), the partition cannot be updated. Similar considerations apply to other statuses that affect access limitations, such as being stopped or locked.

**Exception:** If the HALDB master database has update access (UPD), the partitions can have an access intent of exclusive (EXCL), exceeding the access of the master.

Commands that are issued with a partition name affect only the status of the partition. Commands that are issued against the master database affect only the status of the master database. Therefore, a start of a master database does not update the status of its partitions. If the partitions are stopped, they remain stopped. When a HALDB partition is explicitly stopped, it must be explicitly started again. The type-1 commands with the keyword ALL, type-2 commands with NAME(\*), and commands against a HALDB master do not change the STOPPED (shown as STOACC, STOSCHD, or STOUPDS on QUERY DB) and LOCKED indicators in each HALDB partition.

When the command target is a HALDB master, processing acts on all HALDB partitions. For example, if the IMS command is UPDATE DB STOP(ACCESS) on the HALDB master, all of the HALDB partitions are closed, deallocated, and deauthorized. However, the stopped status is only set in the master database. If a QUERY DB command is issued, only the HALDB master displays a status of STOACC (each HALDB partition does not display STOACC unless it was itself stopped). If a UPDATE DB STOP(ACCESS) command was issued against a HALDB master, the display output of a /DISPLAY DB command shows the HALDB master (as STOPPED), but does not display the status of the partitions.

**Restrictions:**

- The /DBRECOVERY DB command cannot be processed against a HALDB partition on an IMS system while HALDB Online Reorganization (OLR) is running against that partition on the same IMS system.
- The /DBRECOVERY DB command cannot be issued against a HALDB master while OLR is reorganizing any of its partitions.
- If the /DBRECOVERY DB or UPDATE DB STOP(ACCESS) commands are issued for a DEDB database that has newly added areas using the DEDB Alter utility,

you might need to start the DEDB database using the ACCESS option if your PSB has a PCB that is not accessing the newly added areas by their DEDB name. In this situation, issue one of the following commands:

- /STA DB ACCESS=UP
- UPDATE DB START(ACCESS) SET(ACCTYPE(UPD))

The NODBR keyword can be specified in the DFSVSMxx member of the IMS PROCLIB data set to prevent a /DBRECOVERY command from processing against a database that has in-doubt EEQEs. NODBR does not apply to Fast Path databases.

## Equivalent IMS type-2 commands

The following table shows variations of the /DBRECOVERY command and the IMS type-2 commands that perform similar functions.

Table 181. Type-2 equivalents for the /DBRECOVERY command.

Task	/DBRECOVERY command	Similar IMS type-2 command
Stops the accessing and updating of an area.	/DBRECOVERY AREA <i>areaname</i>	UPDATE AREA NAME( <i>areaname</i> ) STOP(ACCESS)
Stops the accessing and updating of all areas and databases of the data group.	/DBRECOVERY DATAGRP <i>datagrpname</i>	UPDATE DATAGRP NAME( <i>datagrpname</i> ) STOP(ACCESS)
Stops access to the database and take it offline.	/DBRECOVERY DB <i>dbname</i>	UPDATE DB NAME( <i>dbname</i> ) STOP(ACCESS) OPTION(FEOV) <sup>1</sup>

<sup>1</sup> This command does not automatically issue checkpoints unless OPTION(FEOV) is specified.

## Examples

The following are examples of the /DBRECOVERY command:

### Example 1 for /DBRECOVERY command

Entry ET:

```
/DBRECOVERY AREA ALL
```

Response ET:

```
DFS058I DBRECOVERY COMMAND IN PROGRESS
DFS0488I DBRECOVERY COMMAND COMPLETED. AREA=DBAREA0
DFS0488I DBRECOVERY COMMAND COMPLETED. AREA=DBAREA1
```

Explanation: DEDB areas are being stopped for inquiry and update, closed, and deallocated; the output log is switched to the next OLDS.

### Example 2 for /DBRECOVERY command

Entry ET:

```
/DBRECOVERY DATABASE SKILL
```

Response ET:

```

DFS058I DBRECOVERY COMMAND IN PROGRESS
DFS2500I DATABASE SKILL SUCCESSFULLY DEALLOCATED
DFS0488I DBRECOVERY COMMAND COMPLETED. DBN=SKILL RC=0
DFS3257I ONLINE LOG NOW SWITCHED FROM DFSOLP( ) TO DFSOLP( )
DFS944I *CHKPT 82123/121314**SIMPLE**

```

Explanation: The new OLDS is used to record a simple checkpoint at 121314 (time) on 82123 (Julian date). The checkpoint number is 82123/121314. All /DBRECOVERY command functions are complete. The Database Recovery utility can be executed after archiving. Currently executing application programs are being terminated. When the termination completes, the SKILL database is stopped for inquiry and update, closed, and deallocated; the output log is switched to the next OLDS. The /START DATABASE command must be used to start the database after the recovery job completes.

### *Example 3 for /DBRECOVERY command*

This example shows how the /DBRECOVERY command might look when entered from the SPOC.

TSO SPOC input:

```
DBR DB BANKATMS BANKTERM BANKLDGR BE3ORDER
```

TSO SPOC output:

```

SYS3 DFS0488I DBR COMMAND COMPLETED. DBN= BANKATMS RC=04
SYS3 DFS0488I DBR COMMAND COMPLETED. DBN= BANKTERM RC=04
SYS3 DFS0488I DBR COMMAND COMPLETED. DBN= BANKLDGR RC=04
SYS3 DFS0488I DBR COMMAND COMPLETED. DBN= BE3ORDER RC= 0
IMS3 DFS0488I DBR COMMAND COMPLETED. DBN= BANKATMS RC=04
IMS3 DFS0488I DBR COMMAND COMPLETED. DBN= BANKTERM RC=04
IMS3 DFS0488I DBR COMMAND COMPLETED. DBN= BANKLDGR RC=04
IMS3 DFS0488I DBR COMMAND COMPLETED. DBN= BE3ORDER RC= 0

```

OM API input:

```
CMD (DBR DB BANKATMS BANKTERM BANKLDGR BE3ORDER)
```

OM API output:

```

<?xml version="1.0"?>
<!DOCTYPE imsout SYSTEM "imsout.dtd">
<imsout>
<ctl>
<omname>OM10M </omname>
<omvsn>1.1.0</omvsn>
<xmlvsn>1 </xmlvsn>
<statime>2002.197 21:52:02.730359</statime>
<stotime>2002.197 21:52:03.383199</stotime>
<staseq>B7EFBE716A9770A4</staseq>
<stoseq>B7EFBE7209F9FD2F</stoseq>
<rqsttkn1>USRT005 10145202</rqsttkn1>
<rc>0200000C</rc>
<rsn>00003008</rsn>
</ctl>
<cmderr>
<mbr name="SYS3 ">
<typ>IMS </typ>
<styp>DBDC </styp>
<rc>00000014</rc>
<rsn>00005050</rsn>
<rsntext>Command processing error</rsntext>
</mbr>
<mbr name="IMS3 ">

```

```

<typ>IMS      </typ>
<styp>DBDC    </styp>
<rc>00000014</rc>
<rsn>00005050</rsn>
<rsntext>Command processing error</rsntext>
</mbr>
</cmderr>
<cmd>
<master>SYS3  </master>
<userid>USRT005 </userid>
<verb>DBR </verb>
<kwd>DB      </kwd>
<input>/DBR DB BANKATMS BANKTERM BANKLDGR BE3ORDER </input>
</cmd>
<msgdata>
<mbr name="SYS3  ">
<msg>DFS0488I  DBR COMMAND COMPLETED.  DBN= BANKATMS RC=04</msg>
<msg>DFS0488I  DBR COMMAND COMPLETED.  DBN= BANKTERM RC=04</msg>
<msg>DFS0488I  DBR COMMAND COMPLETED.  DBN= BANKLDGR RC=04</msg>
<msg>DFS0488I  DBR COMMAND COMPLETED.  DBN= BE3ORDER RC= 0</msg>
</mbr>
<mbr name="IMS3  ">
<msg>DFS0488I  DBR COMMAND COMPLETED.  DBN= BANKATMS RC=04</msg>
<msg>DFS0488I  DBR COMMAND COMPLETED.  DBN= BANKTERM RC=04</msg>
<msg>DFS0488I  DBR COMMAND COMPLETED.  DBN= BANKLDGR RC=04</msg>
<msg>DFS0488I  DBR COMMAND COMPLETED.  DBN= BE3ORDER RC= 0</msg>
</mbr>
</msgdata>
</imsout>

```

Explanation: The DBR command is routed from OM to the two active IMS systems -- SYS3 and IMS3. The response from both IMS systems is returned to OM. The databases BANKATMS, BANKTERM, BANKLDGR, and BE3ORDER are taken offline at both IMS systems.

#### *Example 4 for /DBRECOVERY command*

This example shows how to stop an online forward recovery (OFR) at the RSR tracking site with a /DBRECOVERY command.

Entry ET (on tracking subsystem):

```
/DISPLAY DATABASE OFR
```

Response ET (to tracking subsystem):

DATABASE	TYPE	TOTAL UNUSED	TOTAL UNUSED	ACC	CONDITIONS
BE2PCUST	DL/I			EX	ALLOCS OFR
BE3ORDER	DL/I			EX	ALLOCS OFR
BE3ORDRX	DL/I			EX	ALLOCS OFR
BE3PARTS	DL/I			EX	ALLOCS OFR
BE3PSID1	DL/I			EX	ALLOCS OFR
*91240/132406*		SYS3			

Entry ET (on tracking subsystem):

```
/DBRECOVERY DATABASE BE3ORDER BE3PSID1
```

Response ET (to tracking subsystem):

```

DFS058I DBRECOVERY COMMAND IN PROGRESS
DFS0488I DBR COMMAND COMPLETED.  DBN=BE3ORDER.  RC=0
DFS0488I DBR COMMAND COMPLETED.  DBN=BE3PSID1.  RC=0

```

Explanation: Databases BE3ORDER and BE3PSID1 are closed, unallocated, and deauthorized with DBRC. Online forward recovery for the databases is also stopped.

**Related concepts:**

[↗](#) Maintaining global information for databases, DEDB areas, and transactions (System Administration)

**Related reference:**

“IMS type-1 command format” on page 16

[↗](#) z/OS: Recovering VSAM data sets

[↗](#) DFSVSMxx member of the IMS PROCLIB data set (System Definition)

[↗](#) UPDATE AREA command (Commands)

[↗](#) UPDATE DATAGRPF command (Commands)

[↗](#) UPDATE DB command (Commands)

**Related information:**

[↗](#) DFS2406I (Messages and Codes)

[↗](#) DFS2838I (Messages and Codes)

---

## Chapter 22. /DELETE command

Use the /DELETE command to delete LU 6.2 descriptors, password security authorizations for a given transaction code, logical terminal, relative physical terminal, database, node, or program, and terminal and logical link security for a given transaction code.

This command can be issued to an IMSplex using the Batch SPOC utility.

Subsections:

- "Environment"
- "Syntax"
- "Keywords"

### Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

Table 182. Valid environments for the /DELETE command and keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
/DELETE	X	X	X
DESC	X		X

### Syntax

►► `[/DELETE | /DEL] DESC descriptorname` ►►

### Keywords

The following keyword is valid for the /DELETE command:

#### **DESC *descriptorname***

Specifies the LU62 descriptor to be deleted. Note that this will not delete the descriptor in the PROCLIB member. It is necessary to remove the descriptor from the PROCLIB member; otherwise, the descriptor will be defined at the next IMS restart.



---

## Chapter 23. DELETE commands

Use the IMS DELETE commands to delete Language Environment (LE) runtime options, resources, and resource descriptors.

These commands can be issued through TSO SPOC, IBM Management Console for IMS and DB2 for z/OS, the Manage Resources options in the IMS Applications menu, or other user-written programs that issue commands through OM API. These commands can also be issued to an IMSplex by using the Batch SPOC utility.

Subsections:

- “DELETE DB command”
- “DELETE DBDESC command” on page 365
- “DELETE DEFN command” on page 371
- “DELETE IMSCON commands” on page 390
- “DELETE LE command” on page 405
- “DELETE OTMADESC command” on page 432
- “DELETE PGM command” on page 436
- “DELETE PGMDESC command” on page 441
- “DELETE RTC command” on page 447
- “DELETE RTCDESC command” on page 452
- “DELETE TRAN command” on page 457
- “DELETE TRANDESC command” on page 463

---

### DELETE DB command

Use the DELETE DB command to delete database resources.

Subsections:

- “Environment”
- “Syntax” on page 358
- “Keywords” on page 358
- “Usage notes” on page 358
- “Output fields” on page 359
- “Return, reason, and completion codes” on page 359
- “Examples” on page 363

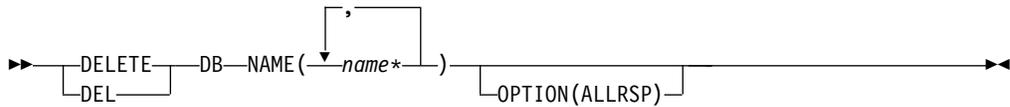
#### Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

*Table 183. Valid environments for the DELETE DB command and keywords*

Command / Keywords	DB/DC	DBCTL	DCCTL
DELETE DB	X	X	
NAME	X	X	
OPTION	X	X	

## Syntax



## Keywords

The following keywords are valid for the DELETE DB command:

### NAME

Specifies the 1-8 character name of the database. Wildcards are supported in the name. Examples of valid names are NAME(\*) and NAME(abc\*). If the NAME parameter specified is a specific or wildcard name, command responses are returned for all the resource names that are processed. For NAME(\*), command responses are returned only for the resource names that resulted in an error. For a DEL DB command when the database contains Fast Path areas, if all areas are successfully deleted, no area names are displayed. If the delete fails for any area, then all area names are displayed to show which deletes were successful and which were not.

### OPTION

Specifies additional functions to be performed along with the command.

### ALLRSP

Indicates that the response lines are to be returned for all resources that are processed on the command. The default action is to return response lines only for the resources that resulted in an error. The ALLRSP keyword is valid only if NAME(\*) is specified.

## Usage notes

This command can only be issued through the Operations Manager API. This command applies to DB/DC systems, DBCTL systems, and DCCTL systems.

The DELETE DB command is not valid on the XRF alternate, RSR tracker, or FDBR region. This command is not valid if online change for MODBLKS is enabled (DFSDFxxx or DFSCGxxx defined with MODBLKS=OLC or MODBLKS not defined).

This command is recoverable.

Each database is deleted individually, unlike the online change process where either all databases are deleted or no databases are deleted. The runtime resource definition for a database can only be deleted if the database is not in use. If the database is in use, the delete fails. In a sysplex environment with multiple IMS subsystems, the delete might succeed on some IMS subsystems and fail on others.

The DELETE DB command removes ACB members from the 31-bit nonresident pool as well as the 64-bit storage pool.

The resources remain deleted across an IMS warm start or emergency restart. To ensure that the resources remain deleted across a cold start, the existing resource definitions must be exported to a system RDDS and then imported from the RDDS

at cold start. Or, if the IMSRSC repository is enabled, the DELETE DEFN command must be issued to delete the definition from the repository. If the resource definitions are to be imported from the MODBLKS data set at cold start instead of from an RDDS or the repository, the resource definitions must be deleted from system definition and a new MODBLKS data set must be generated.

The database can be HSAM, HISAM, HDAM, HIDAM, DEDB, or a HALDB master database. The database cannot be a HALDB partition. If a database is logically related to another database, you need to consider whether the logically related databases should also be deleted. For example, if the database is a HIDAM database, both the primary database and the index database must be deleted.

To delete a HALDB master database, a /DBRECOVERY command must be issued. The /DBRECOVERY command removes this IMS subsystem's knowledge of the HALDB partitions and stops the HALDB master database so that it can be deleted.

## Output fields

The following table shows the DELETE DB output fields. The columns in the table are as follows:

### Short label

Contains the short label generated in the XML output.

### Long label

Contains the long label generated in the XML output.

### Keyword

Identifies keyword on the command that caused the field to be generated. N/A appears for output fields that are always returned. *error* appears for output fields that are returned only in case of an error.

### Meaning

Provides a brief description of the output field.

Table 184. Output fields for the DELETE DB command

Short label	Long label	Keyword	Meaning
CC	CC	N/A	Completion code.
CCTXT	CCText	<i>error</i>	Completion code text that briefly explains the meaning of the nonzero completion code.
DB	DBName	N/A	Database name being deleted.
ERRT	ErrorText	<i>error</i>	Error text that provides diagnostic information. Error text can be returned for a nonzero completion code and further explains the completion code.
MBR	MbrName	N/A	IMSplex member that built the output line.
TYPE	Type	DB	Resource type, which can be AREA, DEDB, DL/I, or blank (unknown). Resource type is returned only if areas are returned.

## Return, reason, and completion codes

The following table includes the return and reason codes and a brief explanation of the codes. The return or reason code returned for the command might also indicate an error from a CSL request.

*Table 185. Return and reason codes for the DELETE DB command*

<b>Return code</b>	<b>Reason code</b>	<b>Meaning</b>
X'00000000'	X'00000000'	Command completed successfully. The command output contains a line for each resource, accompanied by its completion code. If NAME(*) is specified without OPTION(ALLRSP), no output lines are returned. See the completion code table for details.
X'0000000C'	X'00003000'	Command was successful for some resources but failed for others. The command output contains a line for each resource, accompanied by its completion code. If NAME(*) is specified without OPTION(ALLRSP), only resources with nonzero completion codes are returned. See the completion code table for details.
X'0000000C'	X'00003004'	Command was not successful for any of the resources. The command output contains a line for each resource, accompanied by its completion code. See the completion code table for details.
X'00000010'	X'0000400C'	Command is not valid on the XRF alternate.
X'00000010'	X'00004014'	Command is not valid on the RSR tracker.
X'00000010'	X'00004120'	Online change phase is in progress.
X'00000010'	X'00004200'	The DELETE DB command is not processed because IMS shutdown is in progress.
X'00000010'	X'00004300'	Command is not allowed because online change for MODBLKS is enabled (DFSDFxxx or DFSCGxxx defined with MODBLKS=OLC, or MODBLKS not defined).
X'00000010'	X'00004301'	Command is not allowed because the database has a logical relationship with another database.
X'00000014'	X'00005004'	DFSOCMD response buffer could not be obtained.
X'00000014'	X'00005008'	DFSPPOOL storage could not be obtained.
X'00000014'	X'0000500C'	AWE could not be obtained.

Errors unique to the processing of this command are returned as completion codes. The following table includes an explanation of the completion codes.

*Table 186. Completion codes for the DELETE DB command*

<b>Completion code</b>	<b>Completion code text</b>	<b>Meaning</b>	<b>Error text</b>
0		Command completed successfully for database.	
8	COMMAND COMPLETE FOR SOME	Command completed for some of the areas. Area names are returned with a completion code.	
C	COMMAND COMPLETE FOR NONE	Command failed for all of the areas. Area names are returned with completion codes.	

Table 186. Completion codes for the DELETE DB command (continued)

Completion code	Completion code text	Meaning	Error text
10	NO RESOURCES FOUND	Database name is invalid, or the wildcard parameter specified does not match any database names.	
17	ANOTHER CMD IN PROGRESS	Another command (such as DELETE or UPDATE) is in progress for this database. This could also mean this command, if the resource is specified by more than one specific or wildcard parameter.	
23	DB STOP ACCESS IN PROGRESS	A /DBRECOVERY or UPDATE DB STOP(ACCESS) command to stop database access is in progress. This takes the database offline.	
25	DB STOP UPDATES IN PROGRESS	A /DBDUMP or UPDATE DB STOP(UPDATES) command to stop database updates is in progress for the database.	
26	DEDB STOP IN PROGRESS	A /DBRECOVERY, /STOP, or UPDATE DB STOP(SCHD) command to stop database scheduling is in progress for a DEDB.	
27	REJECTED FOR DEFAULT DESCRIPTOR	DELETE command not allowed for user-defined default descriptor. You must issue an UPDATE command to set another descriptor as the default before deleting this descriptor.	
60	GETMAIN STORAGE ERROR	A GETMAIN request for storage was unsuccessful.	
61	DFSBCB STORAGE ERROR	A storage request for a DDIR block failed.	
65	DMB POOL STORAGE ERROR	During a DELETE command for a DB resource, there was an error in the DMB pool processing of a resident DMB.	

Table 186. Completion codes for the DELETE DB command (continued)

Completion code	Completion code text	Meaning	Error text
6C	NOT ALLOWED FOR HALDB PARTITION	DELETE is not allowed for a HALDB partition.	
6F	REFERENCED BY PROGRAM	Program references the database to be deleted. The program name that references this database to be deleted is returned as error text. Suggested actions: Use online change for ACBLIB to change the PSB to no longer reference the database to be deleted or delete the PSB or issue DELETE PGM.	pgmname (8 characters)
76	RECOVER CMD ACTIVE	RECOVER START command is in progress to recover one or more databases with the database recovery services.	
77	DEPENDENT REGIONS ACTIVE	One or more active dependent regions reference the database.	
90	INTERNAL ERROR	An IMS system service failure.	
DE	ACBLIB READ FAILURE	A read request for a member in ACBLIB failed.	
E1	DB MUST BE STOPPED AND OFFLINE	The database must be stopped and taken offline in order for the database to be deleted. You might need to issue a /DBR DB command or an UPDATE DB STOP(ACCESS) command to stop the database and take it offline before issuing the DELETE DB command.	
E2	PARALLEL DB OPEN NOT COMPLETE	The DELETE DB command was rejected because DB parallel open is in progress.	

Table 186. Completion codes for the DELETE DB command (continued)

Completion code	Completion code text	Meaning	Error text
1D7	IMPORT CHANGE LIST ERROR	Resource or descriptor is in the process of being imported from the change list in the IMSRSC repository, or the import from the change list failed. The resource cannot be deleted until it is successfully imported.	

## Examples

The following are examples of the DELETE DB command:

### Example 1 for DELETE DB command

TSO SPOC input:

```
DEL DB NAME(DBHID0*,BADNAME,DEDBJ%1,IPODB)
```

TSO SPOC output:

```
Response for: DEL DB NAME(DBHID0*,BADNAME,DEDBJ%1,IPODB)
DBName      MbrName    CC CText                                     ErrorText
BADNAME     IMS1       10 NO RESOURCES FOUND
DBHID0K2    IMS1       E1 DATABASE HAS NOT BEEN DBR"D
DBHID0K8    IMS1       6F REFERENCED BY PROGRAM                 PBODSAGM
DBHID0K8    IMS1       6F REFERENCED BY PROGRAM                 PBODSAGR
DBHID0K8    IMS1       6F REFERENCED BY PROGRAM                 PBODSALU
DEDBJN01    IMS1       6F REFERENCED BY PROGRAM                 DDLTLM02
DEDBJN21    IMS1       E1 DATABASE HAS NOT BEEN DBR"D
DEDBJ001    IMS1       E1 DATABASE HAS NOT BEEN DBR"D
DEDBJ011    IMS1       E1 DATABASE HAS NOT BEEN DBR"D
DEDBJ021    IMS1       0
DEDBJ031    IMS1       E1 DATABASE HAS NOT BEEN DBR"D
DEDBJ041    IMS1       E1 DATABASE HAS NOT BEEN DBR"D
DEDBJ051    IMS1       E1 DATABASE HAS NOT BEEN DBR"D
DEDBJ061    IMS1       E1 DATABASE HAS NOT BEEN DBR"D
DEDBJ071    IMS1       E1 DATABASE HAS NOT BEEN DBR"D
DEDBJ081    IMS1       E1 DATABASE HAS NOT BEEN DBR"D
DEDBJ091    IMS1       E1 DATABASE HAS NOT BEEN DBR"D
IPODB       IMS1       E1 DATABASE HAS NOT BEEN DBR"D
```

OM API input:

```
CMD(DEL DB NAME(DBHID0*,BADNAME,DEDBJ%1,IPODB))
```

OM API output:

```
<imsout>
<ctl>
<omname>OM10M </omname>
<omvsn>1.3.0</omvsn>
<xmlvsn>20 </xmlvsn>
<statime>2006.310 23:26:32.420713</statime>
<stotime>2006.310 23:26:32.422399</stotime>
<stoseq>BFAACB33F8569D5C</stoseq>
<stoseq>BFAACB33F8BFFD5C</stoseq>
<rqsttkn1>USRT011 10152632</rqsttkn1>
```

```

<rc>0200000C</rc>
<rsn>00003008</rsn>
<rsnmsg>CSLN054I</rsnmsg>
<rsntxt>None of the clients were successful.</rsntxt>
</ctl>
<cmderr>
<mbr name="IMS1  ">
<typ>IMS      </typ>
<styp>DBDC    </styp>
<rc>0000000C</rc>
<rsn>00003000</rsn>
<rsntxt>At least one request successful</rsntxt>
</mbr>
</cmderr>
<cmd>
<master>IMS1    </master>
<userid>USRT011 </userid>
<verb>DEL </verb>
<kwd>DB          </kwd>
<input>DEL DB NAME(DBHID0*,BADNAME,DEDBJ%1,IPODB) </input>
</cmd>
<cmdrsphdr>
<hdr s1bl="DB" l1bl="DBName" scope="LCL" sort="a"
  key="1" scroll="no" len="8" dtype="CHAR" align="left" />
<hdr s1bl="MBR" l1bl="MbrName" scope="LCL" sort="a" key="2" scroll="no"
  len="8" dtype="CHAR" align="left" />
<hdr s1bl="CC" l1bl="CC" scope="LCL" sort="n" key="0" scroll="yes"
  len="4" dtype="INT" align="right" skipb="no" />
<hdr s1bl="CCTXT" l1bl="CCText" scope="LCL" sort="n" key="0"
  scroll="yes" len="*" dtype="CHAR" skipb="yes" align="left" />
<hdr s1bl="ERRT" l1bl="ErrorText" scope="LCL" sort="n" key="0"
  scroll="yes" len="*" dtype="CHAR" skipb="yes" align="left" />
</cmdrsphdr>
<cmdrspdata>
<rsp>DB( DBHIDOK2) MBR(IMS1) CC(  E1) CCTXT(DATABASE HAS NOT BEEN
  DBR"D) </rsp>
<rsp>DB( BADNAME ) MBR(IMS1) CC( 10) CCTXT(NO RESOURCES FOUND) </rsp>
<rsp>DB( DEDBJN01) MBR(IMS1) CC(  6F) CCTXT(REFERENCED BY PROGRAM)
  ERRT(DDLTLMO2) </rsp>
<rsp>DB( IPODB   ) MBR(IMS1) CC(  E1) CCTXT(DATABASE HAS NOT BEEN
  DBR"D) </rsp>
<rsp>DB( DBHIDOK8) MBR(IMS1) CC(  6F) CCTXT(REFERENCED BY PROGRAM)
  ERRT(PBODSAGM) </rsp>
<rsp>DB( DEDBJN21) MBR(IMS1) CC(  E1) CCTXT(DATABASE HAS NOT BEEN
  DBR"D) </rsp>
<rsp>DB( DEDBJ001) MBR(IMS1) CC(  E1) CCTXT(DATABASE HAS NOT BEEN
  DBR"D) </rsp>
<rsp>DB( DEDBJ011) MBR(IMS1) CC(  E1) CCTXT(DATABASE HAS NOT BEEN
  DBR"D) </rsp>
<rsp>DB( DEDBJ021) MBR(IMS1) CC(   0) </rsp>
<rsp>DB( DEDBJ031) MBR(IMS1) CC(  E1) CCTXT(DATABASE HAS NOT BEEN
  DBR"D) </rsp>
<rsp>DB( DEDBJ041) MBR(IMS1) CC(  E1) CCTXT(DATABASE HAS NOT BEEN
  DBR"D) </rsp>
<rsp>DB( DEDBJ051) MBR(IMS1) CC(  E1) CCTXT(DATABASE HAS NOT BEEN
  DBR"D) </rsp>
<rsp>DB( DEDBJ061) MBR(IMS1) CC(  E1) CCTXT(DATABASE HAS NOT BEEN
  DBR"D) </rsp>
<rsp>DB( DEDBJ071) MBR(IMS1) CC(  E1) CCTXT(DATABASE HAS NOT BEEN
  DBR"D) </rsp>
<rsp>DB( DEDBJ081) MBR(IMS1) CC(  E1) CCTXT(DATABASE HAS NOT BEEN
  DBR"D) </rsp>
<rsp>DB( DEDBJ091) MBR(IMS1) CC(  E1) CCTXT(DATABASE HAS NOT BEEN
  DBR"D) </rsp>
<rsp>DB( DBHIDOK8) MBR(IMS1) CC(  6F) CCTXT(REFERENCED BY PROGRAM)
  ERRT(PBODSAGR) </rsp>

```

```

<rsp>DB( DBHIDOK8) MBR(IMS1) CC( 6F) CCTXT(REFERENCED BY PROGRAM)
ERRT(PBODSALU) </rsp>
</cmdrspdata>
</imsout>

```

**Explanation:** A DELETE DB command was issued for several databases. The DELETE DB worked for one database, DEDBJ021, as shown by completion code 0. The DELETE DB command failed for several databases with completion code E1, because the databases need to be taken offline by use of the /DBRECOVERY command first. The DELETE DB command failed for databases DBHIDOK8 and DEDBJN01 with completion code 6F, because the databases are referenced by one or more programs. Each program name is displayed as error text.

**Related concepts:**

 [How to interpret CSL request return and reason codes \(System Programming APIs\)](#)

**Related tasks:**

 [Deleting runtime database resource and descriptor definitions with the DELETE command \(System Definition\)](#)

**Related reference:**

Chapter 2, “Command keywords and their synonyms,” on page 63

## DELETE DBDESC command

Use the DELETE DBDESC command to delete database descriptors.

Subsections:

- “Environment”
- “Syntax” on page 366
- “Keywords” on page 366
- “Usage notes” on page 366
- “Output fields” on page 367
- “Return, reason, and completion codes” on page 367
- “Examples” on page 369

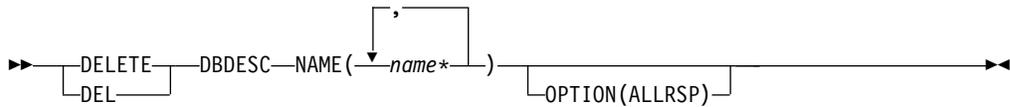
### Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) from which the command keywords can be issued.

*Table 187. Valid environments for the DELETE DBDESC command and keywords*

Command / Keywords	DB/DC	DBCTL	DCCTL
DELETE DBDESC	X	X	
NAME	X	X	
OPTION	X	X	

## Syntax



## Keywords

The following keywords are valid for the DELETE DBDESC command:

### NAME

Specifies the 1-8 character name of the database descriptor. Wildcards are supported in the name. Examples of valid names are NAME(\*) and NAME(abc\*). If the NAME parameter specified is a specific or wildcard name, command responses are returned for all the descriptor names that are processed. For NAME(\*), command responses are returned only for the descriptor names that resulted in an error. OPTION(ALLRSP) can be specified with NAME(\*) to obtain the command responses for all the descriptor names that are processed.

### OPTION

Specifies additional functions to be performed along with the command.

### ALLRSP

Indicates that the response lines are to be returned for all descriptors that are processed on the command. The default action is to return response lines only for the descriptors that resulted in an error. It is valid only with NAME(\*). ALLRSP is ignored for other NAME values.

## Usage notes

This command can only be issued through the Operations Manager API. This command applies to DB/DC systems, and DBCTL system.

The DELETE DBDESC command is not valid on the XRF alternate, RSR tracker, or FDBR region. The DELETE DBDESC command is not valid if online change for MODBLKS is enabled (DFSDFxxx or DFSCGxxx defined with MODBLKS=OLC or MODBLKS not defined).

This command is recoverable.

Each descriptor is deleted individually. Individual deletion does not work like online change where either all descriptors are deleted or no descriptors are deleted. Descriptors can be successfully deleted if they are not currently in use. If a descriptor is in use, the delete fails. In a sysplex environment, the delete might succeed on some IMS systems and fail on others. A descriptor is in use if another command is in progress that references the descriptor.

The descriptors remain deleted across an IMS warm start or emergency restart. To ensure they remain deleted across a cold start, the existing descriptor definitions must be exported to an RDDS and imported from the RDDS during cold start.

The DELETE DBDESC command is not allowed for IMS descriptors that are identified with a definition type of IMS. For DELETE DBDESC commands, this means IMS database descriptor DFSDSDB1.

## Output fields

The following table shows the DELETE DBDESC output fields. The columns in the table are as follows:

### Short label

Contains the short label generated in the XML output.

### Long label

Contains the long label generated in the XML output.

### Keyword

Identifies keyword on the command that caused the field to be generated. N/A appears for output fields that are always returned. *error* appears for output fields that are returned only in case of an error.

### Meaning

Provides a brief description of the output field.

Table 188. Output fields for the DELETE DBDESC command

Short label	Long label	Keyword	Meaning
CC	CC	N/A	Completion code.
CCTXT	CCText	<i>error</i>	Completion code text that briefly explains the meaning of the non-zero completion code.
DESC	DescName	DBDESC	Database descriptor name.
ERRT	ErrorText	<i>error</i>	Error text that provides diagnostic information. Error text can be returned for a non-zero completion code and further explains the completion code.
MBR	MbrName	N/A	IMSplex member that built the output line.

## Return, reason, and completion codes

The following table includes the return and reason codes and a brief explanation of the codes. The return or reason code returned for the command might also indicate an error from a CSL request.

Table 189. Return and reason codes for the DELETE DBDESC command

Return code	Reason code	Meaning
X'00000000'	X'00000000'	Command completed successfully. The command output contains a line for each descriptor, accompanied by its completion code. If NAME(*) is specified without OPTION(ALLRSP), no output lines are returned. See the completion code table for details.
X'0000000C'	X'00003000'	Command was successful for some descriptors but failed for others. The command output contains a line for each descriptor, accompanied by its completion code. If NAME(*) is specified without OPTION(ALLRSP), only descriptors with non-zero completion codes are returned. See the completion code table for details.

Table 189. Return and reason codes for the DELETE DBDESC command (continued)

Return code	Reason code	Meaning
X'0000000C'	X'00003004'	Command was not successful for any of the descriptors. The command output contains a line for each descriptor, accompanied by its completion code. See the completion code table for details.
X'00000010'	X'0000400C'	Command is not valid on the XRF alternate.
X'00000010'	X'00004014'	Command is not valid on the RSR tracker.
X'00000010'	X'00004120'	Online change phase is in progress.
X'00000010'	X'00004200'	The DELETE DBDESC command is not processed because IMS shutdown is in progress.
X'00000010'	X'00004300'	Command is not allowed because online change for MODBLKS is enabled (DFSDFxxx or DFSCGxxx defined with MODBLKS=OLC, or MODBLKS not defined).
X'00000014'	X'00005004'	DFSOCMD response buffer could not be obtained.
X'00000014'	X'00005008'	DFSPOOL storage could not be obtained.
X'00000014'	X'0000500C'	AWE could not be obtained.

Errors unique to the processing of this command are returned as completion codes. The following table includes an explanation of the completion codes.

Table 190. Completion codes for the DELETE DBDESC command

Completion code	Completion code text	Meaning	Error text
0		Command completed successfully for database descriptor.	
17	ANOTHER CMD IN PROGRESS	Another command (such as DELETE or UPDATE) is in progress for this database descriptor. This could also mean this command, if the descriptor is specified by more than one specific or wildcard parameter.	
27	REJECTED FOR DEFAULT DESCRIPTOR	DELETE command not allowed for user-defined default descriptor. You must issue an UPDATE command to set another descriptor as the default before deleting this descriptor.	
48	NOT ALLOWED FOR IMS RESOURCE	DELETE command not allowed for IMS defined-descriptor or resource. An IMS descriptor example is DFSDSDB1.	

Table 190. Completion codes for the DELETE DBDESC command (continued)

Completion code	Completion code text	Meaning	Error text
60	GETMAIN STORAGE ERROR	A GETMAIN request for storage was unsuccessful.	
61	DFSBCB STORAGE ERROR	A storage request for a DDIR block failed.	
90	INTERNAL ERROR	AN IMS system service failure.	
1D7	IMPORT CHANGE LIST ERROR	Descriptor is in the process of being imported from the change list in the IMSRSC repository, or the import from the change list failed. The descriptor cannot be deleted until it is successfully imported.	

## Examples

The following are examples of the DELETE DBDESC command:

### Example 1 for DELETE DBDESC command

TSO SPOC input:

```
DEL DBDESC NAME(*) OPTION(ALLRSP)
```

TSO SPOC output:

```
Response for: DEL DBDESC NAME(*) OPTION(ALLRSP)
DescName MbrName  CC CCText
BRWSDDESC IMS1    27 REJECTED FOR DEFAULT DESCRIPTOR
DESC001  IMS1      0
DESC002  IMS1      0
DESC003  IMS1      0
DESC004  IMS1      0
DESC005  IMS1      0
DFSDSDB1 IMS1    48 NOT ALLOWED FOR IMS RESOURCE
EXCLDESC IMS1      0
RESDESC  IMS1      0
```

OM API input:

```
CMD(DEL DBDESC NAME(*) OPTION(ALLRSP))
```

OM API output:

```
<imsout>
<ctl>
<omname>OM10M  </omname>
<omvsn>1.3.0</omvsn>
<xmlvsn>20  </xmlvsn>
<statime>2006.310 23:15:13.678125</statime>
<stotime>2006.310 23:15:13.679097</stotime>
<staseq>BFAAC8ACABB2D8DC</staseq>
<stoseq>BFAAC8ACABEF9C5C</stoseq>
<rqsttkn1>USRT011 10151513</rqsttkn1>
<rc>0200000C</rc>
<rsn>00003008</rsn>
<rsnmsg>CSLN054I</rsnmsg>
```

```

<rsntxt>None of the clients were successful.</rsntxt>
</ctl>
<cmderr>
<mbr name="IMS1  ">
<typ>IMS  </typ>
<styp>DBDC  </styp>
<rc>0000000C</rc>
<rsn>00003000</rsn>
<rsntxt>At least one request successful</rsntxt>
</mbr>
</cmderr>
<cmd>
<master>IMS1  </master>
<userid>USRT011 </userid>
<verb>DEL </verb>
<kwd>DBDESC  </kwd>
<input>DEL DBDESC NAME(*) OPTION(ALLRSP) </input>
</cmd>
<cmdrsphdr>
<hdr s1b1="DESC" l1b1="DescName" scope="LCL" sort="a" key="1"
  scroll="no" len="8" dtype="CHAR" align="left" />
<hdr s1b1="MBR" l1b1="MbrName" scope="LCL" sort="a" key="2" scroll="no"
  len="8" dtype="CHAR" align="left" />
<hdr s1b1="CC" l1b1="CC" scope="LCL" sort="n" key="0" scroll="yes"
  len="4" dtype="INT" align="right" skipb="no" />
<hdr s1b1="CCTXT" l1b1="CCText" scope="LCL" sort="n" key="0"
  scroll="yes" len="*" dtype="CHAR" skipb="yes" align="left" />
<hdr s1b1="ERRT" l1b1="ErrorText" scope="LCL" sort="n" key="0"
  scroll="yes" len="*" dtype="CHAR" skipb="yes" align="left" />
</cmdrsphdr>
<cmdrspdata>
<rsp>DESC(EXCLDESC) MBR(IMS1) CC( 0) </rsp>
<rsp>DESC(DESC004 ) MBR(IMS1) CC( 0) </rsp>
<rsp>DESC(DESC005 ) MBR(IMS1) CC( 0) </rsp>
<rsp>DESC(DESC001 ) MBR(IMS1) CC( 0) </rsp>
<rsp>DESC(RESDESC ) MBR(IMS1) CC( 0) </rsp>
<rsp>DESC(BRWSDESC) MBR(IMS1) CC( 27) CCTXT(REJECTED FOR DEFAULT
  DESCRIPTOR) </rsp>
<rsp>DESC(DESC002 ) MBR(IMS1) CC( 0) </rsp>
<rsp>DESC(DESC003 ) MBR(IMS1) CC( 0) </rsp>
<rsp>DESC(DFSDSDB1) MBR(IMS1) CC( 48) CCTXT(NOT ALLOWED FOR IMS
  RESOURCE) </rsp>
</cmdrspdata>
</imsout>

```

**Explanation:** A DELETE DBDESC command is issued to delete all database descriptors from an active IMS system. Many database descriptors are successfully deleted, as indicated by the completion code 0. Because the DELETE command attempted to delete all of the database descriptors, OPTION(ALLRSP) is specified to display the successfully deleted database descriptors. Database descriptor BRWSDESC fails, because it is the default database descriptor. Database descriptor DFSDSDB1 fails, because it is the IMS-defined database descriptor.

**Related concepts:**

 [How to interpret CSL request return and reason codes \(System Programming APIs\)](#)

**Related reference:**

Chapter 2, "Command keywords and their synonyms," on page 63

## DELETE DEFN command

Use the DELETE DEFN command to delete one or more stored resource definitions from the IMSRSC repository.

Subsections:

- “Environment”
- “Syntax”
- “Keywords” on page 372
- “Usage notes” on page 376
- “Output fields” on page 378
- “Return, reason, and completion codes” on page 379
- “Examples” on page 382

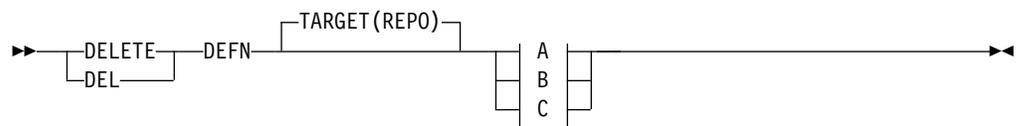
### Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

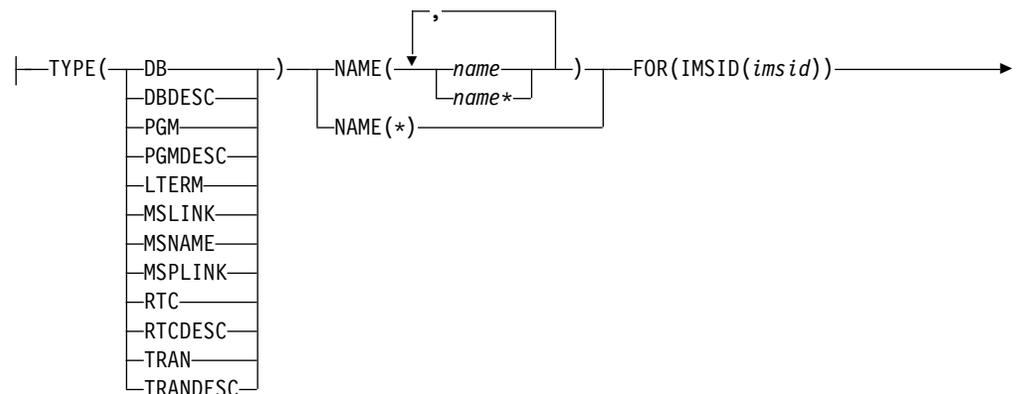
Table 191. Valid environments for the DELETE DEFN command and keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
DELETE DEFN	X	X	X
FOR	X	X	X
NAME	X	X	X
OPTION	X	X	X
TARGET	X	X	X
TYPE	X	X	X

### Syntax



**A:**



OPTION(ALLRSP)

**B:**

TYPE(CHGLIST) FOR(IMSID(*imsid*))

**C:**

TYPE(IMSIDMBR) FOR(IMSID(*imsid*))

## Keywords

The following keywords are valid for the DELETE DEFN command:

**DEFN**

Specifies that resource and descriptor definitions are to be deleted.

**TARGET**

Indicates where the definitions must be deleted from. TARGET(REPO) is the default.

**REPO**

TARGET(REPO) is valid only when IMS is enabled to use the repository. The DELETE DEFN command is defined to OM with ROUTE=ANY. OM routes the command to one command master IMS for processing.

All resource definitions are deleted from the repository as a single unit of work (UOW). If the deletion of any of the resource definitions results in an error, none of the resource definitions are deleted.

A resource definition not being found in the repository is not treated as an error in the command processing. If one or more resource definitions for the names that are specified on the DELETE DEFN command are not found in the repository, the command is processed for the names that have resource definitions in the repository.

If any CSLRPDEL errors are returned for the DELETE command, you must check to see whether there are any in-doubts. If the error is before the UOW is committed, all deletes from the repository are backed out. If the failure is during UOW commit and if the primary data set is updated, the UOW is committed. You must issue the QUERY *xxx* SHOW(DEFN, IMSID) commands to verify whether the resource definitions exist in the repository. If they still exist, you must reissue the DELETE DEFN command.

**Recommendation:** Avoid issuing the DELETE DEFN TARGET(REPO) NAME(\*) FOR(IMSID(\*)) command during peak hours because deleting all the resource definitions from the repository can incur a larger than normal command response time. However, while the resource definitions are being deleted from the repository, there is no impact to other work in IMS.

**TYPE**

Specifies the type of resource or descriptor definitions to delete.

## CHGLIST

Specifies that any residual IMS change list that could not be deleted by IMS during IMS change list processing is to be deleted.

If the IMS system goes down without deleting the change list, another change list cannot be created at the next IMPORT DEFN SOURCE(REPO) SCOPE(ALL) command. The IMPORT command will fail because the change list is locked and not accessible.

Use the CHGLIST keyword under the following situations:

- After a DFS4409A message (REPOSITORY CHANGE LIST IS NOT ACCESSIBLE) is issued and you issue an IMPORT DEFN SOURCE(REPO) SCOPE(ALL) command manually to successfully synchronize IMS with the repository.

Because the IMPORT command is issued manually, the change list is not needed and should be deleted. If the change list is not deleted, it is processed at the next IMS restart.

- After a DFS4401E message is issued (RM CSLRPDEL REQUEST FAILED) when IMS could not delete the change list.

Because IMS could not delete the change list automatically, you can use the DELETE DEFN command to delete the change list. If this residual change list is not deleted, it is deleted at the next IMS restart.

TYPE(CHGLIST) does not apply to TYPE LTERM, MSLINK, MSNAME, or MSPLINK.

**DB** Specifies that one or more database definitions are to be deleted from the repository. If TYPE(DB) is specified on a DCCTL system, completion code 1D3 ("REPOSITORY MEMBER NOT FOUND") is returned for each name specified.

## DBDESC

Specifies that one or more database descriptor definitions are to be deleted from the repository. If TYPE(DBDESC) is specified on a DCCTL system, completion code 1D3 ("REPOSITORY MEMBER NOT FOUND") is returned for each name specified.

## IMSIDMBR

TYPE(IMSIDMBR) can be used on the DELETE DEFN command to delete the IMS ID specified on the FOR(IMSID()) from the repository IMSID member table.

The DELETE DEFN TARGET(REPO) TYPE(IMSIDMBR) FOR(IMSID()) command is processed by the command master IMS. The command will delete the IMS ID from the IMSID member table in the IMSRSC repository.

The DELETE DEFN TYPE(IMSIDMBR) command should be issued:

- before or after the /CHE FREEZE | DUMPQ | PURGE LEAVEPLEX command is issued to cleanup information of the IMS leaving the IMSplex from the IMSRSC repository.
- To remove an IMS ID of an inactive IMS from the repository that has been added in error. For example, if the EXPORT DEFN command or CSLURP10 utility specified an incorrect IMS ID on the IMS ID list.

Only a single specific IMS ID can be specified with TYPE(IMSIDMBR). Wildcard names or multiple IMS IDs are not supported.

The command will first check if any resource definitions exist in the IMSRSC repository for the IMS ID specified in the FOR(IMSID()). When

| any resource definitions exist in the repository for the IMS ID, the  
| command will fail. A response line is returned for each resource type that  
| has one or more resource definitions existing in the repository for the IMS  
| ID that is removed from the IMSID member table. A DELETE DEFN  
| TYPE(rsctype) command must be issued to delete the resource definitions  
| of the resource type that exist in the repository for the IMS and then  
| reissue the DELETE DEFN TYPE(IMSIDMBR) command to remove the  
| IMS ID from the repository IMSID member table.

| If the command is successful, the IMS ID is removed from the IMSRSC  
| repository. A response line is returned with the IMS ID that is removed  
| from the IMSID member table. After the IMS ID is removed from the  
| IMSID member table, a subsequent EXPORT DEFN TARGET(REPO)  
| SET(IMSID(\*)) command routed to any IMS or an autoexport when  
| AUTOEXPORT\_IMSID=ALL is specified and is triggered by another IMS  
| other than the IMS whose IMS ID is removed, will not write the resource  
| definitions for the IMS that has been removed from the IMSID member  
| table.

| **Note:**

| If the IMS that is removed from the repository is still active, an EXPORT  
| DEFN TARGET(REPO) command directed to the IMS, or the next  
| autoexport at the IMS if there are any changes, will write the resource  
| definitions to the repository and re-add the IMS to the repository.

| On a XRF system, if the active and alternate are leaving the IMSplex, then  
| the DELETE DEFN TYPE(IMSIDMBR) command must be issued with the  
| IMS ID of both the active and the alternate.

| **LTERM**

| Specifies that one or more Multiple Systems Coupling (MSC) remote  
| logical terminal (LTERM) definitions are to be deleted from the repository.  
| If TYPE(LTERM) is specified on a DBCTL system, completion code 1D3  
| ("REPOSITORY MEMBER NOT FOUND") is returned.

| **MSLINK**

| Specifies that one or more MSC logical link definitions are to be deleted  
| from the repository. If TYPE(MSLINK) is specified on a DBCTL system,  
| completion code 1D3 ("REPOSITORY MEMBER NOT FOUND") is  
| returned.

| **MSNAME**

| Specifies that one or more MSC logical link path definitions are to be  
| deleted from the repository. If TYPE(MSNAME) is specified on a DBCTL  
| system, completion code 1D3 ("REPOSITORY MEMBER NOT FOUND") is  
| returned.

| **MSPLINK**

| Specifies that one or more MSC physical link definitions are to be deleted  
| from the repository. If TYPE(MSPLINK) is specified on a DBCTL system,  
| completion code 1D3 ("REPOSITORY MEMBER NOT FOUND") is  
| returned.

| **PGM**

| Specifies that one or more program definitions are to be deleted from the  
| repository.

**PGMDESC**

Specifies that one or more program descriptor definitions are to be deleted from the repository.

**RTC**

Specifies that one or more routing code definitions are to be deleted from the repository. If TYPE(RTC) is specified on a DBCTL system, completion code 1D3 ("REPOSITORY MEMBER NOT FOUND") is returned for each name specified.

**RTCDESC**

Specifies that one or more routing code descriptor definitions are to be deleted from the repository. If TYPE(RTCDESC) is specified on a DBCTL system, completion code 1D3 ("REPOSITORY MEMBER NOT FOUND") is returned for each name specified.

**TRAN**

Specifies that one or more transaction definitions are to be deleted from the repository. If TYPE(TRAN) is specified on a DBCTL system, completion code 1D3 ("REPOSITORY MEMBER NOT FOUND") is returned for each name specified.

When an FP(E) transaction is deleted from the repository by using the DELETE DEFN command, the routing code that is associated with the FP(E) transaction will also be deleted from the repository by RM.

**TRANDESC**

Specifies that one or more transaction descriptor definitions are to be deleted from the repository. If TYPE(TRANDESC) is specified on a DBCTL system, completion code 1D3 ("REPOSITORY MEMBER NOT FOUND") is returned for each name specified.

**NAME**

Specifies the 1- to 8-character name of a resource or descriptor. Wildcards can be specified in the name. The name is a repeatable parameter. If the NAME parameter specified is a specific name or a wildcard, command responses are returned for all the resource names that are processed. NAME is a required parameter.

For NAME(\*), command responses are returned only for the resource and descriptor names that resulted in an error. OPTION(ALLRSP) can be specified with NAME(\*) to obtain the command responses for all the resource and descriptor names that are deleted.

NAME is not supported with TYPE(CHGLIST) or TYPE(IMSIDMBR).

**FOR**

Specifies what actions to be set.

**IMSID**

Specifies the list of IMS IDs for which the resource and descriptor definitions are to be deleted.

An IMS resource list is a list of resource and descriptor names and types that defines the resources for an IMS system. IMS resource lists exist in the repository and are used during IMS cold start to identify all the resource and descriptor definitions that are to be read during the cold start. The IMSID values specified on the DELETE DEFN command identify the IMS resource lists from which the resources are to be deleted when a stored resource definition is deleted from the repository.

The IMSID value can be a specific IMS ID, a list of specific IMS IDs, a wildcard name of \*, or a wildcard name with \* or %.

If the IMSID value specified is a wildcard name other than \*, the list of IMS IDs in RM that match the IMS ID wildcard name will be used on the command. The list of IMS IDs that are processed is returned on the command output.

IMSID(\*) indicates that all IMS resource lists must be updated to remove the resource name being deleted. As a result of the DELETE DEFN command, the resource definition does not belong to any IMS systems that are defined to use the repository.

You can specify one or more IMS IDs. When one or more IMS IDs are specified, only the IMS resource lists for the specified IMS IDs are modified to delete the resource names being deleted from the repository.

If the resource name belongs to one or more IMS resource lists, and the DELETE DEFN command does not specify deletion from all IMS IDs, the stored resource definition from the repository is not deleted. Only the IMS resource lists for the specified IMS subsystems are modified to delete the resource names that are specified on the command.

The IMS ID specified on the FOR(IMSID()) keyword must be an IMS that is defined to RM to use the repository. The IMS specified might be active or down when the command is issued.

Only a single specific IMS ID can be specified with the TYPE(CHGLIST) or TYPE(IMSIDMBR). Wildcard names or multiple IMS IDs are not supported.

#### **OPTION**

Specifies additional functions to be performed along with the command.

#### **ALLRSP**

Specifies that all response lines must be returned. You can specify OPTION(ALLRSP) only if NAME(\*) is also specified. If OPTION(ALLRSP) is not specified, only the response lines for the names in error are returned.

OPTION(ALLRSP) is not valid when NAME parameters are specified. A response line is returned for each name that is specified on the NAME keyword.

### **Usage notes**

The DELETE DEFN command is valid from the Operations Manager (OM) API. This command is defined as ROUTE=ANY to OM and is processed only by the command master IMS.

You cannot delete an MSC physical link definition from the IMSRSC repository if the physical link is referenced by one or more logical links in the repository.

You cannot delete an MSC logical link definition from the IMSRSC repository if the logical link is referenced by one or more logical link paths in the repository.

You cannot delete an MSC logical link path definition from the IMSRSC repository if the logical link path is referenced by one or more remote LTERMs in the repository.

When you delete MSC resource definitions from the repository for multiple resource types, delete the definitions in the following order:

- Remote transaction definitions
- Remote LTERM definitions
- MSC logical link paths (MSNAMEs)
- MSC logical links (MSLINKs)
- MSC physical links (MSPLINKs)

The DELETE DEFN TARGET(REPO) command is enhanced to add the TYPE(IMSIDMBR) keyword.

The DELETE DEFN TARGET(REPO) TYPE(IMSIDMBR) FOR(IMSID()) command is processed by the command master IMS. The command deletes the IMS ID from the IMSID member table in the IMSRSC repository.

You cannot delete a program definition from the repository if the program is referenced by one or more transaction or routing code definitions in the IMSRSC repository.

When you delete definitions from the repository for multiple resource types, delete the definitions in the following order:

1. Transaction and transaction descriptor definitions
2. Routing code and routing code descriptor definitions
3. Program and program descriptor definitions

Database and database descriptor definitions can be deleted at any time.

Before a program definition is deleted from the IMSRSC repository, each transaction and routing code definition that is in the repository is examined to determine whether it references the program. However, for performance reasons, if you specify NAME(\*) on the DELETE DEFN TARGET(REPO) TYPE(PGM) command to delete all program definitions for the specified IMS system, the check is not made for the transaction and routing code definitions to determine whether they reference the program definitions. Instead, a check is made to determine whether any transactions or routing codes are defined in the repository for the specified IMS system. If one or more transactions or routing codes are defined in the repository for the IMS system, the command fails. The DEL DEFN TARGET(REPO) TYPE(PGM) NAME(\*) command also fails if a remote transaction definition is in the repository for the specified IMS system, even if the remote transaction definition does not reference a program. Therefore, if you issue the DEL DEFN TARGET(REPO) TYPE(PGM) command with NAME(\*) specified, you must delete all transaction and routing code definitions from the repository before you issue the command.

As a result of the DELETE DEFN command with keyword DB, DBDESC, RTC, RTCDESC, PGM, PGMDESC, TRAN, or TRANDESC, definitions of the specified resource names are deleted from the IMSRSC repository for the specified IMS systems. Additionally, the DELETE DEFN command deletes the resource names from an IMS change list in the repository if a change list exists for one or more IMS systems that are specified on the FOR() keyword. Because the resource definition is being deleted and will not be available to be imported when the IMS system restarts, it is also deleted from the IMS system's change list.

The IMS change list in the repository is not created when a DELETE DEFN command is issued and one or more IMS systems that are specified on the FOR() keyword are down. When you delete a resource definition, delete the IMS runtime definition from IMS by using the DELETE command and then delete the stored

resource definition from the repository by using the DELETE DEFN command. If the stored resource definition is deleted from the repository for the IMS before the runtime resource definition is deleted from the IMS system, the resource definition still exists in the IMS after restart. In this case, issue the DELETE command to delete the runtime resource definition after IMS restarts.

**Considerations for Fast Path exclusive transactions and their associated routing codes:** When you delete a definition for a Fast Path exclusive transaction from the repository, the definition for its associated routing code is also deleted. If you issue the DELETE DEFN command to delete a routing code definition from the repository and the routing code is associated with a Fast Path exclusive transaction, the deletion fails. To delete the routing code definition, you must first delete the transaction definition.

**Consideration for an XRF-capable system:** When the DELETE DEFN command is used in an XRF environment, and the resource definition exists in the IMSRSC repository for both the active system and the alternate system, you must issue the DELETE DEFN command with the IMS IDs of both the active system and the alternate system on the FOR(IMSID(*imsid*)) keyword.

## Output fields

The DELETE DEFN TARGET(REPO) command results in the DFS3406I message, which indicates that the deletion from the repository was successful, or the DFS3407E message, which indicates that the deletion was not successful. These messages include the repository name and repository type. If the deletion from the repository is not successful, the DFS3407E message consists of the CSLRPDEL request return code and reason code. The CSLRPDEL return and reason codes are in the CSLRRR macro.

The following table shows the DELETE DEFN output fields. The columns in the table are as follows:

### Short label

Contains the short label that is generated in the XML output.

### Long label

Contains the long label that is generated in the XML output.

### Keyword

Identifies keyword on the command that caused the field to be generated. N/A indicates output fields that are always returned. *error* indicates output fields that are returned only in case of an error.

### Meaning

Provides a brief description of the output field.

Table 192. Output fields for the DELETE DEFN command

Short label	Long label	Keyword	Meaning
CC	CC	N/A	Completion code.
CCTXT	CCText	<i>error</i>	Completion code text that briefly explains the meaning of the nonzero completion code.
DTYP	DelType	N/A	The value CHGLIST is returned under DelType if the resource name on the output line was deleted from the IMS change list and the IMS resource list.

Table 192. Output fields for the DELETE DEFN command (continued)

Short label	Long label	Keyword	Meaning
IMSID	IMSid	N/A	The names of IMS resource lists that are updated or deleted by the DELETE DEFN command.
MBR	MbrName	N/A	The IMS identifier of the command master IMS that processed the DELETE DEFN command.
NAME	Name	N/A	The name of the resource or descriptor definition that was updated or deleted from the repository, or the name of the IMS resource list that was updated in the repository.
TYPE	Type	N/A	The resource or descriptor type, which can be DB, DBDESC, LTERM, MSLINK, MSNAME, MSPLINK, PGM, PGMDESC, RTC, RTCDESC, TRAN, or TRANDESC; or LIST for IMS resource lists; or IMSID for IMS ID.

## Return, reason, and completion codes

The following table includes the return and reason codes and a brief explanation of the codes. The return or reason code returned for the command might also indicate an error from a CSL request.

Table 193. Return and reason codes for the DELETE DEFN command

Return code	Reason code	Meaning
X'00000000'	X'00000000'	Command completed successfully.
X'0000000C'	X'00003004'	No requests were successful.
X'00000008'	X'00002004'	Invalid command keyword or invalid command keyword combination.
X'00000008'	X'00002009'	Multiple values for same attribute.
X'00000008'	X'00002010'	Generic value not supported for parameter.
X'00000008'	X'00002018'	Invalid characters in IMS ID.
X'00000008'	X'00002141'	TYPE parameter conflict.
X'00000010'	X'00004004'	No CQS address space.
X'00000010'	X'00004018'	No resource structure, or resource structure not available.
X'00000010'	X'0000402C'	Command is not valid because MSC is not enabled in the IMS system.
X'00000010'	X'0000402D'	Command is not allowed because dynamic definition is not enabled for MSC. MSCRSCS=DYN is not specified in the MSC section of the DFSDFxxx member of the IMS PROCLIB data set.
X'00000010'	X'00004100'	Resource structure is full.
X'00000010'	X'00004104'	No RM address space.
X'00000010'	X'00004108'	No SCI address space.
X'00000010'	X'00004500'	IMS is not enabled to use the repository.

Table 193. Return and reason codes for the DELETE DEFN command (continued)

Return code	Reason code	Meaning
X'00000010'	X'00004501'	RM is not enabled with the repository.
X'00000010'	X'00004502'	Repository is not available.
X'00000010'	X'00004503'	Repository is stopped.
X'00000010'	X'00004504'	Repository spare recovery is in progress.
X'00000010'	X'00004505'	No IMS resource list exists, or no resources for the resource type exist in the IMS resource list.
X'00000010'	X'00004507'	Repository access is denied.
X'00000010'	X'00004508'	Repository maximum put length exceeded.
X'00000010'	X'00004509'	RM data version is lower than the IMS data version.
X'00000010'	X'0000450A'	Repository Server is being shut down.
X'00000010'	X'0000450B'	Repository Server is not available.
X'00000010'	X'0000450C'	Repository Server is busy.
X'00000010'	X'0000450D'	RM failed to define some of the internal fields that are related to the IMSRSC repository.
X'00000010'	X'00004511'	Programs cannot be deleted because transactions exist for one or more IMS systems.
X'00000010'	X'00004512'	Programs cannot be deleted because routing codes exist for one or more IMS systems.
X'00000014'	X'0000501C'	IMODULE GETMAIN error.
X'00000014'	X'00005100'	RM request error.
X'00000014'	X'00005104'	CQS error.
X'00000014'	X'00005108'	SCI request error.
X'00000014'	X'00005110'	Repository error.
X'00000014'	X'0000551C'	A request to update the repository was rejected. Either a corrupted resource list was detected, or the update request was rejected to prevent a corruption of the resource list. Contact IBM Software Support. Send in the output from the command in error, a dump of the local RM address space where the command executed or the dump of all RM address spaces in the IMSplex. Also, send a copy of the IMSRSC repository RID and RMD data sets.

Errors unique to the processing of this command are returned as completion codes. The following table includes an explanation of the completion codes.

Table 194. Completion codes for the DELETE DEFN command

Completion code	Completion code text	Meaning
0		Command completed successfully for database.
10	NO RESOURCES FOUND	The specified resource was not found.
39	FP=E TRAN FOR RTC EXISTS	A routing code definition is being deleted, but the routing code is associated with a Fast Path exclusive FP(E) transaction.

Table 194. Completion codes for the DELETE DEFN command (continued)

Completion code	Completion code text	Meaning
48	NOT ALLOWED FOR IMS RESOURCE	The specified resource is not allowed.
5F	INVALID CHARACTERS IN NAME	The specified name contains invalid characters.
7B	REFERENCED BY RTCODE	A program definition is being deleted, but the program is referenced by one or more routing code definitions.
88	REFERENCED BY TRAN	A program definition is being deleted, but the program is referenced by one or more transaction definitions.
94	RM REQUEST ERROR	An error resulted in the command not being processed for the RM.
14D	INVALID CHARACTER IN IMSID	There is an invalid character in the IMS ID.
1D0	NOT DONE DUE TO ERROR	An error resulted in the command not being processed for the resource name in the response line. The resource name that had the error will show the error that was encountered. All other resource names will have 1D0 completion code because the deletion was aborted due to one error. All members are deleted as a single unit of work.
1D1	NO RESPOSITORY DEFINED	An error resulted due to repository not being defined.
1D3	MEMBER NOT FOUND	The specified resource name is not found.
1D4	REPOSITORY MEMBER IN USE	The specified repository member is already in use.
1D5	NO IMS RESOURCE LIST	The specified IMS resource list is not found.
1D6	REPOSITORY ERROR	An error resulted due to a repository error.
1D8	CHANGE LIST IN USE	The DELETE DEFN TARGET(REPO) command could not be completed because the change list for an IMS was in use and not accessible to be deleted.
1D9	DEFINITIONS EXIST FOR RSCTYPE	The IMS ID cannot be deleted as resource definitions exist for the resource type specified under TYPE column. Resource definitions must be deleted before the IMS ID can be deleted.
209	REFERENCED BY MSLINK	The physical link cannot be deleted because a logical link is assigned to the physical link.

Table 194. Completion codes for the DELETE DEFN command (continued)

Completion code	Completion code text	Meaning
20A	REFERENCED BY MSNAME	The logical link cannot be deleted because it is referenced by one or more logical link paths (MSNAMEs).
20B	REFERENCED BY LTERM	The MSNAME cannot be deleted because it is referenced by one of more remote LTERMs.

## Examples

The following are examples of the DELETE DEFN command:

### Example 1 for DELETE DEFN command

TSO SPOC input:

```
DELETE DEFN TARGET(REPO) NAME(*) TYPE(DB) FOR(IMSID(*))
```

TSO SPOC output:

```
IMSpIex . . . . . : PLEX1
Routing . . . . . :
Start time. . . . : 2011.133 12:45:21.34
Stop time . . . . : 2011.133 12:45:21.82
Return code . . . : 00000000
Reason code . . . : 00000000
Reason text . . . :
Command master. . : IMS1
```

MbrName Messages

```
-----
IMS1 DFS3406I DELETE COMMAND COMPLETE FOR ALL, REPOTYPE=IMSRSC,
      REPONAME=IMSRSC_REPOSITORY
```

OM API input:

```
CMD(DELETE DEFN TARGET(REPO) NAME(*) TYPE(DB) FOR(IMSID(*)))
```

OM API output:

```
<imsout>
<ctl>
<omname>OM10M </omname>
<omvsn>1.5.0</omvsn>
<xmlvsn>20 </xmlvsn>
<statime>2011.133 19:45:21.399790</statime>
<stotime>2011.133 19:45:21.823791</stotime>
<staseq>C7C3DD48D2FEE475</staseq>
<stoseq>C7C3DD493A82F2A5</stoseq>
<rqsttkn1>USRT011 10124521</rqsttkn1>
<rc>00000000</rc>
<rsn>00000000</rsn>
</ctl>
<cmd>
<master>IMS1 </master>
<userid>USRT011 </userid>
<verb>DEL </verb>
<kwd>DEFN </kwd>
<input>DEL DEFN TARGET(REPO) NAME(*) TYPE(DB) FOR(IMSID(*)) </input>
</cmd>
<msgdata>
```

```

<mbr name="IMS1  ">
<msg>DFS3406I DELETE COMMAND COMPLETE FOR ALL, REPOTYPE=IMSRSC,
  REPONAME=IMSRSC_REPOSITORY                                </msg>
</mbr>
</msgdata>
</imsout>

```

**Explanation:** All the stored resource definitions for all the database resources and descriptors are deleted in the repository. This command is processed by the command master IMS. The resource list for all IMS systems that are defined to use the repository is updated.

*Example 2 for DELETE DEFN command*

TSO SPOC input:

```

DEL DEFN TARGET(REPO) TYPE(RTC) NAME(EMHTX2,RTC2)
FOR(IMSID(IMS2,IMS3))

```

TSO SPOC output:

```

Response for: DEL DEFN TARGET(REPO) TYPE(RTC) NAME(EMHTX2,RTC2)
              FOR(IMSID(IMS2,IMS3))
Name      Type      MbrName  CC  CCText                IMSid
IMS2      LIST      IMS1     0
IMS3      LIST      IMS1     0
EMHTX2    RTC        IMS1     0                        IMS2
EMHTX2    RTC        IMS1     0                        IMS3
RTC2      RTC        IMS1     1D3 REPOSITORY MEMBER NOT FOUND

```

When the PF4 key is pressed, the following is displayed:

Log for . . : DEL DEFN TARGET(REPO) TYPE(RTC) NAME(EMHTX2,RTC... More:

```

IMSpIex . . . . . : PLEX1
Routing . . . . . :
Start time. . . . : 2011.133 11:41:54.47
Stop time . . . . : 2011.133 11:41:54.55
Return code . . . : 0200000C
Reason code . . . : 00003008
Reason text . . . : None of the clients were successful.
Command master. . : IMS1

```

MbrName	Return Code	Reason Code	Reason text
IMS1	0000000C	00003000	At least one request successful

```

MbrName  Messages
-----  -----
IMS1     DFS3407E DELETE COMMAND COMPLETE FOR SOME, REPOTYPE=IMSRSC,
        REPONAME=IMSRSC_REPOSITORY

```

OM API input:

```

CMD(DEL DEFN TARGET(REPO) TYPE(RTC) NAME(EMHTX2,RTC2)
FOR(IMSID(IMS2,IMS3)))

```

OM API output:

```

<imsout>
<ctl>
<omname>OM10M </omname>
<omvsn>1.5.0</omvsn>
<xmlvsn>20 </xmlvsn>
<statime>2011.133 18:41:54.498030</statime>
<stotime>2011.133 18:41:54.547807</stotime>

```

```

<staseq>C7C3CF1A479EE33C</staseq>
<stoseq>C7C3CF1A53C5FFF0</stoseq>
<rqsttkn1>USRT011 10114154</rqsttkn1>
<rc>0200000C</rc>
<rsn>00003008</rsn>
<rsnmsg>CSLN054I</rsnmsg>
<rsntxt>None of the clients were successful.</rsntxt>
</ctl>
<cmderr>
<mbr name="IMS1  ">
<typ>IMS  </typ>
<styp>DBDC  </styp>
<rc>0000000C</rc>
<rsn>00003000</rsn>
<rsntxt>At least one request successful</rsntxt>
</mbr>
</cmderr>
<cmd>
<master>IMS1  </master>
<userid>USRT011 </userid>
<verb>DEL </verb>
<kwd>DEFN  </kwd>
<input>DEL DEFN TARGET(REPO) TYPE(RTC) NAME(EMHTX2,RTC2)
FOR(IMS2,IMS3) </input>
</cmd>
<cmdrsphdr>
<hdr s1b1="NAME" l1b1="Name" scope="LCL" sort="a" key="2" scroll="no"
len="8" dtype="CHAR" align="left" />
<hdr s1b1="TYPE" l1b1="Type" scope="LCL" sort="a" key="1" scroll="no"
len="8" dtype="CHAR" align="left" />
<hdr s1b1="MBR" l1b1="MbrName" scope="LCL" sort="a" key="3" scroll="no"
len="8" dtype="CHAR" align="left" />
<hdr s1b1="CC" l1b1="CC" scope="LCL" sort="n" key="0" scroll="yes"
len="4" dtype="INT" align="right" />
<hdr s1b1="CCTXT" l1b1="CCText" scope="LCL" sort="n" key="0"
scroll="yes" len="32" dtype="CHAR" align="left" skipb="yes" />
<hdr s1b1="IMS2" l1b1="IMS2" scope="LCL" sort="a" key="4" scroll="no"
len="8" dtype="CHAR" align="left" skipb="yes" />
</cmdrsphdr>
<cmdrspdata>
<rsp>NAME(IMS2  ) TYPE(LIST) MBR(IMS1) CC( 0) </rsp>
<rsp>NAME(IMS3  ) TYPE(LIST) MBR(IMS1) CC( 0) </rsp>
<rsp>NAME(RTC2  ) TYPE(RTC) MBR(IMS1) CC( 1D3) CCTXT(REPOSITORY
MEMBER NOT FOUND)</rsp>
<rsp>NAME(EMHTX2 ) TYPE(RTC) MBR(IMS1) CC( 0) IMS2(IMS2  ) </rsp>
<rsp>NAME(EMHTX2 ) TYPE(RTC) MBR(IMS1) CC( 0) IMS3(IMS3  ) </rsp>
</cmdrspdata>
<msgdata>
<mbr name="IMS1  ">
<msg>DFS3407E DELETE COMMAND COMPLETE FOR SOME, REPOTYPE=IMSRSC,
REPONAME=IMSRSC_REPOSITORY </msg>
</mbr>
</msgdata>
</imsout>

```

**Explanation:** Routing code resource EMHTX2 exists in the IMSRSC repository, and routing code resource RTC2 does not exist in the repository. The DELETE DEFN command completed successfully with some of the resources deleted. IMS1 processed the DELETE DEFN command on behalf of IMS2 and IMS3. Routing code resource EMHTX2 in the repository is deleted for both IMS2 and IMS3. The resource list of IMS2 and the resource list of IMS3 are updated to remove routing code resource EMHTX2 from the resource lists. IMS2 and IMS3 do not have routing code resource RTC2 in the repository. The completion code is 1D3, and the completion code text is REPOSITORY MEMBER NOT FOUND.

### Example 3 for DELETE DEFN command

TSO SPOC input:

```
DEL DEFN TARGET(REPO) TYPE(RTC) NAME(EMHTX2,RTC2)
FOR(IMSID(IMS2,IMS3))
```

TSO SPOC output:

```
Response for: DEL DEFN TARGET(REPO) TYPE(RTC) NAME(EMHTX2,RTC2)
              FOR(IMSID(IMS2,IMS3))
Name   Type   MbrName  CC CCText
IMS2   LIST   IMS1     1D0 NOT DONE DUE TO ERROR
IMS3   LIST   IMS1     1D0 NOT DONE DUE TO ERROR
EMHTX2 RTC     IMS1     1D3 REPOSITORY MEMBER NOT FOUND
RTC2   RTC     IMS1     1D3 REPOSITORY MEMBER NOT FOUND
```

When the PF4 key is pressed, the following is displayed:

```
Log for . . : DEL DEFN TARGET(REPO) TYPE(RTC) NAME(EMHTX2,RTC... More:  +>
```

```
IMSpIex . . . . . : PLEX1
Routing . . . . . :
Start time . . . . : 2011.133 12:07:41.08
Stop time . . . . . : 2011.133 12:07:41.14
Return code . . . . : 0200000C
Reason code . . . . : 00003008
Reason text . . . . : None of the clients were successful.
Command master. . . : IMS1
```

MbrName	Return Code	Reason Code	Reason text
IMS1	0000000C	00003004	No requests were successful

```
MbrName  Messages
-----  -----
IMS1     DFS3407E DELETE COMMAND COMPLETE FOR NONE, REQUEST=CSLRPDEL, RC=0300000C,
         RSN=00003004, ERRORTXT=, MBR=RSC, REPOTYPE=IMSRSC,
         REPONAME=IMSRSC_REPOSITORY
```

OM API input:

```
CMD(DEL DEFN TARGET(REPO) TYPE(RTC) NAME(EMHTX2,RTC2)
FOR(IMSID(IMS2,IMS3)))
```

OM API output:

```
<imsout>
<ctl>
<omname>OM10M  </omname>
<omvsn>1.5.0</omvsn>
<xmlvsn>20  </xmlvsn>
<statime>2011.133 19:07:41.108147</statime>
<stotime>2011.133 19:07:41.139772</stotime>
<staseq>C7C3D4DD3DFB3522</staseq>
<stoseq>C7C3D4DD45B3CC2C</stoseq>
<rqsttkn1>USRT011 10120741</rqsttkn1>
<rc>0200000C</rc>
<rsn>00003008</rsn>
<rsnmsg>CSLN054I</rsnmsg>
<rsntxt>None of the clients were successful.</rsntxt>
</ctl>
<cmderr>
<mbr name="IMS1  ">
<typ>IMS  </typ>
<styp>DBDC  </styp>
<rc>0000000C</rc>
<rsn>00003004</rsn>
<rsntxt>No requests were successful</rsntxt>
</mbr>
</cmderr>
```

```

<cmd>
<master>IMS1 </master>
<userid>USR011 </userid>
<verb>DEL </verb>
<kwd>DEFN </kwd>
<input>DEL DEFN TARGET(REPO) TYPE(RTC) NAME(EMHTX2,RTC2)
FOR(IMSID(IMS2,IMS3)) </input>
</cmd>
<cmdrsphdr>
<hdr s1b1="NAME" l1b1="Name" scope="LCL" sort="a" key="2" scroll="no"
len="8" dtype="CHAR" align="left" />
<hdr s1b1="TYPE" l1b1="Type" scope="LCL" sort="a" key="1" scroll="no"
len="8" dtype="CHAR" align="left" />
<hdr s1b1="MBR" l1b1="MbrName" scope="LCL" sort="a" key="3" scroll="no"
len="8" dtype="CHAR" align="left" />
<hdr s1b1="CC" l1b1="CC" scope="LCL" sort="n" key="0" scroll="yes"
len="4" dtype="INT" align="right" />
<hdr s1b1="CCTXT" l1b1="CCText" scope="LCL" sort="n" key="0"
scroll="yes" len="32" dtype="CHAR" align="left" skipb="yes" />
<hdr s1b1="IMSID" l1b1="IMSid" scope="LCL" sort="a" key="4" scroll="no"
len="8" dtype="CHAR" align="left" skipb="yes" />
</cmdrsphdr>
<cmdrspdata>
<rsp>NAME(IMS2 ) TYPE(LIST) MBR(IMS1) CC( 1D0) CCTXT(NOT DONE DUE TO ERROR) </rsp>
<rsp>NAME(IMS3 ) TYPE(LIST) MBR(IMS1) CC( 1D0) CCTXT(NOT DONE DUE TO ERROR) </rsp>
<rsp>NAME(EMHTX2 ) TYPE(RTC) MBR(IMS1) CC( 1D3) CCTXT(REPOSITORY MEMBER NOT FOUND) </rsp>
<rsp>NAME(RTC2 ) TYPE(RTC) MBR(IMS1) CC( 1D3) CCTXT(REPOSITORY MEMBER NOT FOUND) </rsp>
</cmdrspdata>
<msgdata>
<mbr name="IMS1 ">
<msg>DFS3407E DELETE COMMAND COMPLETE FOR NONE, REQUEST=CSLRPDEL,
RC=0300000C, RSN=00003004, ERRORTXT=, MBR=RSC, REPOTYPE=IMSRSC,
REPONAME=IMSRSC_REPOSITORY </msg>
</mbr>
</msgdata>
</imsout>

```

**Explanation:** Routing code resources EMHTX2 and RTC2 do not exist in the IMSRSC repository. The DELETE DEFN command completed successfully with none of the resources deleted. IMS1 processed the DELETE DEFN command on behalf of IMS2 and IMS3. IMS2 and IMS3 do not have routing code resources EMHTX2 and RTC2 in the repository. The completion code is 1D3, and the completion code text is REPOSITORY MEMBER NOT FOUND. The resource list of IMS2 and the resource list of IMS3 are not updated due to error. The completion code is 1D0, and the completion code text is NOT DONE DUE TO ERROR.

#### *Example 4 for DELETE DEFN command*

TSO SPOC input:

```

DEL DEFN TARGET(REPO) TYPE(RTC) NAME(EMH*,R*)
FOR(IMSID(IMS2,IMS3))

```

TSO SPOC output:

```

Response for: DEL DEFN TARGET(REPO) TYPE(RTC) NAME(EMH*,R*)
FOR(IMSID(IMS2,IMS3))

```

Name	Type	MbrName	CC	IMSid
IMS2	LIST	IMS1	0	
IMS3	LIST	IMS1	0	
EMHTX3	RTC	IMS1	0	IMS2
EMHTX3	RTC	IMS1	0	IMS3
RTC3	RTC	IMS1	0	IMS2
RTC3	RTC	IMS1	0	IMS3

When the PF4 key is pressed, the following is displayed:

```

Log for . . : DEL DEFN TARGET(REPO) TYPE(RTC) NAME(EMH*,R*) FO... More: >

IMSp1ex . . . . . : PLEX1
Routing . . . . . :

```

```

Start time. . . . : 2011.133 12:22:50.10
Stop time . . . . : 2011.133 12:22:50.17
Return code . . . : 00000000
Reason code . . . : 00000000
Reason text . . . :
Command master. . : IMS1

```

```

MbrName  Messages
-----

```

```

IMS1      DFS3406I DELETE COMMAND COMPLETE FOR ALL, REPOTYPE=IMSRSC,
          REPONAME=IMSRSC_REPOSITORY

```

OM API input:

```

CMD(DEL DEFN TARGET(REPO) TYPE(RTC) NAME(EMH*,R*)
FOR(IMSID(IMS2,IMS3)))

```

OM API output:

```

<imsout>
<ctl>
<omname>OM10M  </omname>
<omvsn>1.5.0</omvsn>
<xmlvsn>20  </xmlvsn>
<statime>2011.133 19:22:50.121164</statime>
<stotime>2011.133 19:22:50.167741</stotime>
<staseq>C7C3D84024FCC3A0</staseq>
<stoseq>C7C3D840305BDD22</stoseq>
<rqsttkn1>USRT011 10122250</rqsttkn1>
<rc>00000000</rc>
<rsn>00000000</rsn>
</ctl>
<cmd>
<master>IMS1  </master>
<userid>USRT011 </userid>
<verb>DEL </verb>
<kwd>DEFN      </kwd>
<input>DEL DEFN TARGET(REPO) TYPE(RTC) NAME(EMH*,R*)
FOR(IMSID(IMS2,IMS3)) </input>
</cmd>
<cmdrsphdr>
<hdr s1b1="NAME" l1b1="Name" scope="LCL" sort="a" key="2" scroll="no"
  len="8" dtype="CHAR" align="left" />
<hdr s1b1="TYPE" l1b1="Type" scope="LCL" sort="a" key="1" scroll="no"
  len="8" dtype="CHAR" align="left" />
<hdr s1b1="MBR" l1b1="MbrName" scope="LCL" sort="a" key="3" scroll="no"
  len="8" dtype="CHAR" align="left" />
<hdr s1b1="CC" l1b1="CC" scope="LCL" sort="n" key="0" scroll="yes"
  len="4" dtype="INT" align="right" />
<hdr s1b1="CCTXT" l1b1="CCText" scope="LCL" sort="n" key="0"
  scroll="yes" len="32" dtype="CHAR" align="left" skipb="yes" />
<hdr s1b1="IMSID" l1b1="IMSid" scope="LCL" sort="a" key="4" scroll="no"
  len="8" dtype="CHAR" align="left" skipb="yes" />
</cmdrsphdr>
<cmdrspdata>
<rsp>NAME(IMS2      ) TYPE(LIST) MBR(IMS1) CC(  0) </rsp>
<rsp>NAME(IMS3      ) TYPE(LIST) MBR(IMS1) CC(  0) </rsp>
<rsp>NAME(EMHTX3    ) TYPE(RTC) MBR(IMS1) CC(  0) IMSID(IMS2      ) </rsp>
<rsp>NAME(RTC3      ) TYPE(RTC) MBR(IMS1) CC(  0) IMSID(IMS2      ) </rsp>
<rsp>NAME(EMHTX3    ) TYPE(RTC) MBR(IMS1) CC(  0) IMSID(IMS3      ) </rsp>
<rsp>NAME(RTC3      ) TYPE(RTC) MBR(IMS1) CC(  0) IMSID(IMS3      ) </rsp>
</cmdrspdata>
<msgdata>
<mbr name="IMS1      ">
<msg>DFS3406I DELETE COMMAND COMPLETE FOR ALL, REPOTYPE=IMSRSC,

```

```

    REPONAME=IMSRSC_REPOSITORY                                </msg>
</mbr>
</msgdata>
</imsout>

```

**Explanation:** Routing code resources EMHTX3 and RTC3 exist in the IMSRSC repository. The DELETE DEFN command completed successfully with all of the resources deleted. IMS1 processed the DELETE DEFN command on behalf of IMS2 and IMS3. Routing code resource EMHTX3 that exists in the repository matched the wildcard resource name of EMH\*. Routing code resource RTC3 that exists in the repository matched the wildcard resource name of R\*. Routing code resources EMHTX3 and RTC3 exist in the repository for IMS2 and IMS3. Routing code resources EMHTX3 and RTC3 are deleted from the repository for IMS2 and IMS3. The resource list of IMS2 and the resource list of IMS3 are updated to remove routing code resources EMHTX3 and RTC3 from the resource lists.

*Example 5 for DELETE DEFN command*

Entry ET:

```
DELETE DEFN TYPE(IMSIDMBR) FOR(IMSID(IMS1))
```

Response ET:

Name	Type	MbrName	CC	CCText
IMS1	DB	IMS2	1D9	DEFINITIONS EXIST FOR RSCTYPE
IMS1	DBDESC	IMS2	1D9	DEFINITIONS EXIST FOR RSCTYPE
IMS1	IMSID	IMS2	1D0	NOT DONE DUE TO ERROR
IMS1	PGM	IMS2	1D9	DEFINITIONS EXIST FOR RSCTYPE
IMS1	PGMDESC	IMS2	1D9	DEFINITIONS EXIST FOR RSCTYPE
IMS1	RTC	IMS2	1D9	DEFINITIONS EXIST FOR RSCTYPE
IMS1	RTCDESC	IMS2	1D9	DEFINITIONS EXIST FOR RSCTYPE
IMS1	TRAN	IMS2	1D9	DEFINITIONS EXIST FOR RSCTYPE
IMS1	TRANDESC	IMS2	1D9	DEFINITIONS EXIST FOR RSCTYPE

**Explanation:** The DELETE DEFN TYPE(IMSIDMBR) FOR(IMSID(IMS1)) command deletes the IMS ID IMS1 from the repository IMSID member table. The command fails as resource definitions of all 8 types exist in the repository for the IMS ID IMS1. You must issue DELETE DEFN TARGET(REPO) FOR(IMSID(IMS1)) NAME(\*) TYPE(rsctype) where *rsctype* is DB|DBDESC|PGM|PGMDESC|RTC|RTCDESC|TRAN|TRANDESC to delete one or more resource definitions of the resource type in the repository for the IMS ID IMS1. After this command, the DELETE DEFN TYPE(IMSIDMBR) FOR(IMSID(IMS1)) command must be re-issued.

*Example 6 for DELETE DEFN command*

Entry ET:

```
DELETE DEFN TYPE(IMSIDMBR) FOR(IMSID(IMS1))
```

Response ET:

Name	Type	MbrName	CC
IMS1	IMSID	IMS2	0

**Explanation:** The DELETE DEFN TYPE(IMSIDMBR) FOR(IMSID(IMS1)) command deletes the IMS ID IMS1 from the repository IMSID member table. If the command is successful, IMS1 is no longer defined in the IMSRSC repository.

*Example 7 for DELETE DEFN command*

Entry ET:

```
DELETE DEFN TYPE(IMSIDMBR) FOR(IMSID(IMSX))
```

Response ET:

Name	Type	MbrName	CC	CCText
IMSX	IMSID	IMS1	1D3	REPOSITORY MEMBER NOT FOUND

**Explanation:** The DELETE DEFN TYPE(IMSIDMBR) FOR(IMSID(IMSX)) command deletes the IMS ID IMSX from the repository IMSID member table. The command is not successful, if the IMS IDIMSX is not defined in the IMSRSC repository.

*Example 8 for DELETE DEFN command*

TSO SPOC input:

```
DELETE DEFN TARGET(REPO) TYPE(LTERM) NAME(B0P06) FOR(IMSID(IMS1,IMS2))
```

TSO SPOC output:

```
(screen 1)
Response for: DELETE DEFN TARGET(REPO) TYPE(LTERM) NAME(B0P06)...
Name      Type      MbrName    CC  CCText      IMSid
IMS1      LIST      IMS1       0
IMS2      LIST      IMS1       1D3 REPOSITORY MEMBER NOT FOUND
B0P06     LTERM     IMS1       0      IMS1
```

OM API input:

```
CMD(DEL TARGET(REPO) TYPE(LTERM) NAME(B0P06) FOR(IMSID(IMS1,IMS2))
```

OM API output:

```
<cmdrspHdr>
<hdr s1bl="NAME" l1bl="Name" scope="LCL" sort="a" key="2" scroll="no"
  len="8" dtype="CHAR" align="left" />
<hdr s1bl="TYPE" l1bl="Type" scope="LCL" sort="a" key="1" scroll="no"
  len="8" dtype="CHAR" align="left" />
<hdr s1bl="MBR" l1bl="MbrName" scope="LCL" sort="a" key="3" scroll="no"
  len="8" dtype="CHAR" align="left" />
<hdr s1bl="CC" l1bl="CC" scope="LCL" sort="n" key="0" scroll="yes"
  len="4" dtype="INT" align="right" />
<hdr s1bl="CCTXT" l1bl="CCText" scope="LCL" sort="n" key="0"
  scroll="yes" len="32" dtype="CHAR" align="left" skipb="yes" />
<hdr s1bl="IMSID" l1bl="IMSid" scope="LCL" sort="a" key="4" scroll="no"
  len="8" dtype="CHAR" align="left" skipb="yes" />
<hdr s1bl="DTYP" l1bl="DelType" scope="LCL" sort="n" key="1" scroll="no"
  len="8" dtype="CHAR" align="left" skipb="yes" />
</cmdrspHdr>
<cmdrspdata>
<rsp>NAME(IMS1      ) TYPE(LIST) MBR(IMS1) CC( 0) </rsp>
<rsp>NAME(IMS2      ) TYPE(LIST) MBR(IMS1) CC( 1D3) CCTXT(REPOSITORY
  MEMBER NOT FOUND) </rsp>
<rsp>NAME(B0P06     ) TYPE(LTERM) MBR(IMS1) CC( 0) IMSID(IMS1      )
</rsp>
</cmdrspdata>
```

**Explanation:** MSC remote LTERM is defined to IMS1 and the IMSRSC repository for IMS1, but it is not defined to IMS2 locally or in the repository. This DELETE DEFN command fails with a completion code for IMS2, but succeeds for IMS1. LTERM B0P06 is not found in the resource list for IMS2.

**Related information:**

 [DFS3406I \(Messages and Codes\)](#)

 [DFS3407E \(Messages and Codes\)](#)

---

## DELETE IMSCON commands

Use the DELETE IMSCON commands to delete IMS Connect resource definitions during runtime processing without having to restart your IMS Connect instance.

The TYPE keyword specifies the type of IMS Connect resource to delete.

The DELETE IMSCON command is processed by every IMS Connect to which OM routes the command, whether or not OM has designated a particular IMS Connect as the command master.

IMS Connect resources that are deleted by use of this command will reappear after the next restart of IMS Connect, unless the corresponding configuration statement is also removed from the HWSCFGxx member of the IMS PROCLIB data set before the next restart.

Subsections:

- “DELETE IMSCON TYPE(DATASTORE) command”
- “DELETE IMSCON TYPE(IMSPLEX) command” on page 395
- “DELETE IMSCON TYPE(PORT) command” on page 400

**Related reference:**

 [IMS Connect type-2 commands \(Commands\)](#)

### DELETE IMSCON TYPE(DATASTORE) command

Use the DELETE IMSCON TYPE(DATASTORE) command to delete a communications path to an IMS data store in IMS Connect.

Subsections:

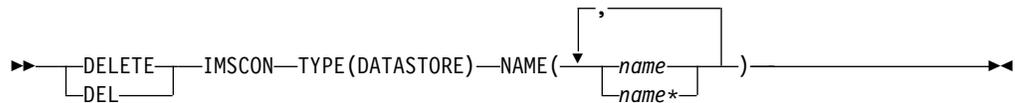
- “Environment”
- “Syntax” on page 391
- “Keywords” on page 391
- “Usage notes” on page 391
- “Equivalent WTOR and z/OS commands” on page 393
- “Output fields” on page 391
- “Return, reason, and completion codes” on page 392
- “Examples” on page 393

#### Environment

The DELETE IMSCON command is applicable only to IMS Connect. To issue this command, the following conditions must be satisfied:

- IMS Connect must be active and configured to communicate with the Common Service Layer (CSL) Structured Call Interface (SCI).
- A type-2 command environment with Structured Call Interface (SCI) and Operations Manager (OM) must be active.

## Syntax



## Keywords

The following keywords are valid for the `DELETED IMSCON TYPE(DATASTORE)` command.

### NAME

Specifies the 1-8 character name of the IMS data store. The name can consist of alphanumeric characters and must be unique within IMS Connect.

This name is equivalent to the ID parameter of the `DATASTORE` statement in the IMS Connect configuration member.

Wildcards are supported in the name. This parameter is required.

## Usage notes

You can issue the `DELETED IMSCON TYPE(DATASTORE)` command only through the Operations Manager (OM) API.

IMS Connect can process IMS Connect type-2 commands only if the IMSplex from which the commands were issued has a status of `ACTIVE`.

The communications path to the IMS data store must be stopped (in `NOTACTIVE` status) before it can be deleted by use of the `DELETED IMSCON TYPE(DATASTORE)` command.

IMS Connect resources that are deleted by use of this command will reappear after the next restart of IMS Connect, unless the corresponding configuration statement is also removed from the `HWSCFGxx` member of the IMS `PROCLIB` data set before the next restart.

## Output fields

### Short label

Contains the short label that is generated in the XML output.

### Long label

Contains the column heading displayed on the TSO SPOC screen.

### Keyword

Identifies the keyword on the command that caused the field to be generated. `N/A` (not applicable) is displayed for output fields that are always returned. `error` is displayed for output fields that are returned only in the case of an error.

## Meaning

Provides a brief description of the output field.

*Table 195. Output fields for the DELETE IMSCON TYPE(DATASTORE) command*

Short label	Long label	Keyword	Meaning
CC	CC	N/A	Completion code that indicates whether IMS Connect was able to process the command for the specified resource. The completion code is always returned. See Table 197.
CCTXT	CCText	N/A	Completion code text that briefly explains the meaning of the nonzero completion code. This field is returned only for an error completion code.
DS	DataStore	N/A	The data store name. The data store name is always returned.
MBR	MbrName	N/A	Identifier of the IMS Connect that built the output line. The identifier is always returned.

## Return, reason, and completion codes

The return and reason codes that can be returned as a result of the DELETE IMSCON TYPE(DATASTORE) command are standard for all commands entered through the OM API.

The following table includes the return and reason codes and a brief explanation of the codes. The return or reason code returned for the command might also indicate an error from a CSL request.

*Table 196. Return and reason codes for the DELETE IMSCON TYPE(DATASTORE) command*

Return code	Reason code	Meaning
X'00000000'	X'00000000'	The DELETE IMSCON TYPE(DATASTORE) command completed successfully. The command output contains a line for each resource, accompanied by its completion code. For details, see Table 197.
X'0C00000C'	X'00003000'	The command was successful for some resources but failed for others. The command output contains a line for each resource, accompanied by its completion code.
X'0C00000C'	X'00003004'	The command was not successful for any resource. The command output contains a line for each resource, accompanied by its completion code.

Errors unique to the processing of this command are returned as completion codes. A completion code is returned for each action against an individual resource.

*Table 197. Completion codes for the DELETE IMSCON TYPE(DATASTORE) command*

Completion code	Completion code text	Meaning
0		The DELETE IMSCON TYPE(DATASTORE) command completed successfully for the resources.
10	No resources found	The data store was not found.

Table 197. Completion codes for the DELETE IMSCON TYPE(DATASTORE) command (continued)

Completion code	Completion code text	Meaning
14	Resource is not stopped	The resource is not stopped. To delete this resource, it must be stopped first.

## Equivalent WTOR and z/OS commands

There are no equivalent WTOR and IMS Connect z/OS commands that perform similar functions as the DELETE IMSCON TYPE(DATASTORE) command.

## Examples

### Example 1 for DELETE IMSCON TYPE(DATASTORE) command

TSO SPOC input:

```
DELETE IMSCON TYPE(DATASTORE) NAME(IMS1)
```

TSO SPOC output:

```
DataStore MbrName      CC
IMS1      HWS1          0
```

OM API input:

```
CMD(DEL IMSCON TYPE(DATASTORE) NAME(IMS1))
```

OM API output:

```
<?xml version="1.0"?>
<!DOCTYPE imsout SYSTEM "imsout.dtd">
<imsout>
<ctl>
<omname>OM10M </omname>
<omvsn>1.7.0</omvsn>
<xmlvsn>20 </xmlvsn>
<statime>2014.064 18:26:57.848593</statime>
<stotime>2014.064 18:26:57.848849</stotime>
<staseq>CCCF07338D111E45</staseq>
<stoseq>CCCF07338D211145</stoseq>
<rqsttkn1>USRT011 10102657</rqsttkn1>
<rc>00000000</rc>
<rsn>00000000</rsn>
</ctl>
<cmd>
<master>HWS1 </master>
<userid>USRT011 </userid>
<verb>DEL </verb>
<kwd>IMSCON </kwd>
<input>DELETE IMSCON TYPE(DATASTORE) NAME(IMS1) </input>
</cmd>
<cmdrsphdr>
<hdr s1bl="DS" l1bl="DataStore" scope="LCL" sort="a" key="1"
scroll="no" len="8" dtype="CHAR" align="left" skipb="no" />
<hdr s1bl="MBR" l1bl="MbrName" scope="LCL" sort="a" key="2"
scroll="no" len="8" dtype="CHAR" align="left" skipb="no" />
<hdr s1bl="CC" l1bl="CC" scope="LCL" sort="n" key="0" scroll="yes"
len="4" dtype="INT" align="right" skipb="no" />
<hdr s1bl="CCTXT" l1bl="CCText" scope="LCL" sort="n" key="0"
scroll="yes" len="32" dtype="CHAR" align="left" skipb="yes" />
</cmdrsphdr>
```

```

| <cmdrspdata>
| <rsp>DS(IMS1 ) MBR(HWS1 ) CC( 0) </rsp>
| </cmdrspdata>
| </imsout>

```

**Explanation:** The communications to IMS data store IMS1 is deleted.

*Example 2 for DELETE IMSCON TYPE(DATASTORE) command*

TSO SPOC input:

```
DELETE IMSCON TYPE(DATASTORE) NAME(IM*,SYS3,ABC1)
```

TSO SPOC output:

DataStore	MbrName	CC	CCText
ABC1	HWS1	10	No resources found
IMS1	HWS1	0	
IMS2	HWS1	0	
SYS3	HWS1	14	Resource is not stopped

OM API input:

```
CMD(DEL IMSCON TYPE(DATASTORE) NAME(IM*,SYS3,ABC1))
```

OM API output:

```

| <?xml version="1.0"?>
| <!DOCTYPE imsout SYSTEM "imsout.dtd">
| <imsout>
| <ctl>
| <omname>OM10M </omname>
| <omvsn>1.7.0</omvsn>
| <xmlvsn>20 </xmlvsn>
| <statime>2014.064 18:42:19.322200</statime>
| <stotime>2014.064 18:42:19.322450</stotime>
| <staseq>CCCCF0AA256358E43</staseq>
| <stoseq>CCCCF0AA256452943</stoseq>
| <rqsttkn1>USRT011 10104219</rqsttkn1>
| <rc>0200000C</rc>
| <rsn>00003008</rsn>
| <rsnmsg>CSLN054I</rsnmsg>
| <rsntxt>None of the clients were successful.</rsntxt>
| </ctl>
| <cmderr>
| <mbr name="HWS1 ">
| <typ>IMSCON </typ>
| <rc>0C00000C</rc>
| <rsn>00003000</rsn>
| <rsntxt>At least one request successful </rsntxt>
| </mbr>
| </cmderr>
| <cmd>
| <master>HWS1 </master>
| <userid>USRT011 </userid>
| <verb>DEL </verb>
| <kwd>IMSCON </kwd>
| <input>DELETE IMSCON TYPE(DATASTORE) NAME(IM*,SYS3,ABC1) </input>
| </cmd>
| <cmdrsphdr>
| <hdr s1b1="DS" l1b1="DataStore" scope="LCL" sort="a" key="1"
| scroll="no" len="8" dtype="CHAR" align="left" skipb="no" />
| <hdr s1b1="MBR" l1b1="MbrName" scope="LCL" sort="a" key="2"
| scroll="no" len="8" dtype="CHAR" align="left" skipb="no" />
| <hdr s1b1="CC" l1b1="CC" scope="LCL" sort="n" key="0" scroll="yes"
| len="4" dtype="INT" align="right" skipb="no" />
| <hdr s1b1="CCTXT" l1b1="CCText" scope="LCL" sort="n" key="0"

```



## Keywords

The following keywords are valid for the DELETE IMSCON TYPE(IMSPLEX) command.

### NAME

Specifies a 1- to 5-character name of the IMSplex.

This name is equivalent to the TMEMBER parameter of the IMSPLEX statement in the IMS Connect configuration member.

Wildcards are supported in the name. This parameter is required.

## Usage notes

You can issue the DELETE IMSCON TYPE(IMSPLEX) command only through the Operations Manager (OM) API.

IMS Connect can process IMS Connect type-2 commands only if the IMSplex from which the commands were issued has a status of ACTIVE.

The communications path to the IMSplex must be stopped (in NOT ACTIVE state) before it can be deleted by use of the DELETE IMSCON TYPE(IMSPLEX) command.

IMS Connect communicates with an IMSplex to receive and respond to type-2 command requests. For this reason, you cannot use the DELETE command to delete the IMSplex that IMS Connect uses for type-2 command processing. When NAME(\*) is specified on the DELETE command, communications paths to all IMSplex systems are deleted except for the IMSplex that is used for type-2 command processing.

If the IMSplex is used by another resource such as MSC, ISC, or ODBM, the IMSplex cannot be deleted.

IMS Connect resources that are deleted by use of this command will reappear after the next restart of IMS Connect, unless the corresponding configuration statement is also removed from the HWSCFGxx member of the IMS PROCLIB data set before the next restart.

## Output fields

### Short label

Contains the short label that is generated in the XML output.

### Long label

Contains the column heading displayed on the TSO SPOC screen.

### Keyword

Identifies the keyword on the command that caused the field to be generated. N/A (not applicable) is displayed for output fields that are always returned. *error* is displayed for output fields that are returned only in the case of an error.

### Meaning

Provides a brief description of the output field.

Table 198. Output fields for the DELETE IMSCON TYPE(IMSPLEX) command

Short label	Long label	Keyword	Meaning
CC	CC	N/A	Completion code that indicates whether IMS Connect was able to process the command for the specified resource. The completion code is always returned. See Table 200.
CCTXT	CCText	N/A	Completion code text that briefly explains the meaning of the nonzero completion code. This field is returned only for an error completion code.
IMSPLX	IMSpIex	N/A	The IMSpIex name. The IMSpIex name is always returned.
MBR	MbrName	N/A	Identifier of the IMS Connect that built the output line. The identifier is always returned.

## Return, reason, and completion codes

The return and reason codes that can be returned as a result of the DELETE IMSCON TYPE(IMSPLEX) command are standard for all commands that are entered through the OM API.

The following table includes the return and reason codes and a brief explanation of the codes. The return or reason code returned for the command might also indicate an error from a CSL request.

Table 199. Return and reason codes for the DELETE IMSCON TYPE(IMSPLEX) command

Return code	Reason code	Meaning
X'00000000'	X'00000000'	The DELETE IMSCON TYPE(IMSPLEX) command completed successfully. The command output contains a line for each resource, accompanied by its completion code. For details, see Table 200.
X'0C000008'	X'00002018'	The required NAME parameter is not specified.
X'0C00000C'	X'00003000'	The command was successful for some resources but failed for others. The command output contains a line for each resource, accompanied by its completion code.
X'0C00000C'	X'00003004'	The command was not successful for any resource. The command output contains a line for each resource, accompanied by its completion code.

Errors unique to the processing of this command are returned as completion codes. A completion code is returned for each action against an individual resource.

Table 200. Completion codes for the DELETE IMSCON TYPE(IMSPLEX) command

Completion code	Completion code text	Meaning
0		The DELETE IMSCON TYPE(IMSPLEX) command completed successfully for the resources.
10	No resources found	The IMSpIex was not found.
14	Resource is not stopped	The resource is not stopped. To delete this resource, it must be stopped first.

Table 200. Completion codes for the DELETE IMSCON TYPE(IMSPLEX) command (continued)

Completion code	Completion code text	Meaning
41	NAME value is too long	The specified NAME value is too long. IMSplex name is maximum of 5 characters.
45	Cmd processing uses this IMSplex	The IMSplex that IMS Connect uses to process this command cannot be deleted.
46	ODBM requires this IMSplex	The IMSplex that is used for ODBM communications cannot be deleted.
47	MSC requires this IMSplex	The IMSplex that is used for MSC communications cannot be deleted.
48	ISC requires this IMSplex	The IMSplex that is used for ISC communications cannot be deleted.

## Equivalent WTOR and z/OS commands

There are no equivalent WTOR and IMS Connect z/OS commands that perform similar functions as the DELETE IMSCON TYPE(IMSPLEX) command.

## Examples

### Example 1 for DELETE IMSCON TYPE(IMSPLEX) command

TSO SPOC input:

```
DELETE IMSCON TYPE(IMSPLEX) NAME(PLEX2)
```

TSO SPOC output:

```
IMSplex  MbrName      CC
PLEX2    HWS1         0
```

OM API input:

```
CMD(DEL IMSCON TYPE(IMSPLEX) NAME(PLEX2))
```

OM API output:

```
<?xml version="1.0"?>
<!DOCTYPE imsout SYSTEM "imsout.dtd">
<imsout>
<ctl>
<omname>OM10M </omname>
<omvsn>1.7.0</omvsn>
<xmlvsn>20 </xmlvsn>
<statime>2014.227 20:45:11.305908</statime>
<stotime>2014.227 20:45:11.306154</stotime>
<staseq>CD9C161AF0EB4E45</staseq>
<stoseq>CD9C161AF0FAA045</stoseq>
<rqsttkn1>USRT011 10134511</rqsttkn1>
<rc>00000000</rc>
<rsn>00000000</rsn>
</ctl>
<cmd>
<master>HWS1 </master>
<userid>USRT011 </userid>
<verb>DEL </verb>
<kwd>IMSCON </kwd>
<input>DELETE IMSCON TYPE(IMSPLEX) NAME(PLEX2) </input>
```

```

| </cmd>
| <cmdrsphdr>
| <hdr s1b1="IMSPLEX" l1b1="IMSpIex" scope="LCL" sort="a" key="1" scroll="no"
|   len="8" dtype="CHAR" align="left" skipb="no" />
| <hdr s1b1="MBR" l1b1="MbrName" scope="LCL" sort="a" key="2" scroll="no"
|   len="8" dtype="CHAR" align="left" skipb="no" />
| <hdr s1b1="CC" l1b1="CC" scope="LCL" sort="n" key="0" scroll="yes" len="4"
|   dtype="INT" align="right" skipb="no" />
| <hdr s1b1="CCTXT" l1b1="CCText" scope="LCL" sort="n" key="0" scroll="yes"
|   len="32" dtype="CHAR" align="left" skipb="yes" />
| </cmdrsphdr>
| <cmdrspdata>
| <rsp>IMSPLEX(PLEX2 ) MBR(HWS1 ) CC( 0) </rsp>
| </cmdrspdata>
| </imsout>

```

**Explanation:** The communication path to IMSplex PLEX2 is deleted.

### *Example 2 for DELETE IMSCON TYPE(IMSPLEX) command*

TSO SPOC input:

```
DELETE IMSCON TYPE(IMSPLEX) NAME(PLEX*)
```

TSO SPOC output:

IMSpIex	MbrName	CC CCText
PLEX1	HWS1	45 Cmd processing uses this IMSplex
PLEX2	HWS1	0
PLEX3	HWS1	14 Resource is not stopped
PLEX4	HWS1	46 ODBM requires this IMSplex

OM API input:

```
CMD(DELETE IMSCON TYPE(IMSPLEX) NAME(PLEX*))
```

OM API output:

```

| <?xml version="1.0"?>
| <!DOCTYPE imsout SYSTEM "imsout.dtd">
| <imsout>
| <ctl>
| <omname>OM10M </omname>
| <omvsn>1.7.0</omvsn>
| <xmlvsn>20 </xmlvsn>
| <statime>2014.227 21:10:58.581664</statime>
| <stotime>2014.227 21:10:58.581933</stotime>
| <staseq>CD9C1BDE89CA05C0</staseq>
| <stoseq>CD9C1BDE89DADDC0</stoseq>
| <rqsttkn1>USRT011 10141058</rqsttkn1>
| <rc>0200000C</rc>
| <rsn>00003008</rsn>
| <rsnmsg>CSLN054I</rsnmsg>
| <rsntxt>None of the clients were successful.</rsntxt>
| </ctl>
| <cmderr>
| <mbr name="HWS1 ">
| <typ>IMSCON </typ>
| <rc>0C00000C</rc>
| <rsn>00003000</rsn>
| <rsntxt>At least one request successful </rsntxt>
| </mbr>
| </cmderr>
| <cmd>
| <master>HWS1 </master>
| <userid>USRT011 </userid>
| <verb>DEL </verb>

```

```

| <kwd>IMSCON          </kwd>
| <input>DELETE IMSCON TYPE(IMSPLEX) NAME(PLEX*) </input>
| </cmd>
| <cmdrsphdr>
| <hdr s1b1="IMSPLEX" l1b1="IMSpIex" scope="LCL" sort="a" key="1" scroll="no"
|   len="8" dtype="CHAR" align="left" skipb="no" />
| <hdr s1b1="MBR" l1b1="MbrName" scope="LCL" sort="a" key="2" scroll="no"
|   len="8" dtype="CHAR" align="left" skipb="no" />
| <hdr s1b1="CC" l1b1="CC" scope="LCL" sort="n" key="0" scroll="yes" len="4"
|   dtype="INT" align="right" skipb="no" />
| <hdr s1b1="CCTXT" l1b1="CCText" scope="LCL" sort="n" key="0" scroll="yes"
|   len="32" dtype="CHAR" align="left" skipb="yes" />
| </cmdrsphdr>
| <cmdrspdata>
| <rsp>IMSPLEX(PLEX1 ) MBR(HWS1          ) CC( 45) CCTXT(Cmd processing
|   uses this IMSpIex) </rsp>
| <rsp>IMSPLEX(PLEX2 ) MBR(HWS1          ) CC(  0) </rsp>
| <rsp>IMSPLEX(PLEX3 ) MBR(HWS1          ) CC( 14) CCTXT(Resource is not
|   stopped) </rsp>
| <rsp>IMSPLEX(PLEX4 ) MBR(HWS1          ) CC( 46) CCTXT(ODBM requires
|   this IMSpIex) </rsp>
| </cmdrspdata>
| </imsout>

```

**Explanation:** The command is attempting to delete all communication paths to IMSplex systems that start with the characters 'PLEX'. IMSplex PLEX1 cannot be deleted because it is the IMSplex used to send this command. IMSplex PLEX2 was deleted. IMSplex PLEX3 cannot be deleted because it is not stopped. IMSplex PLEX4 cannot be deleted because communication to ODBM requires that IMSplex.

## DELETE IMSCON TYPE(PORT) command

Use the DELETE IMSCON TYPE(PORT) command to delete a listening port in IMS Connect.

This command cannot be used to delete SSL or LOCAL ports.

Subsections:

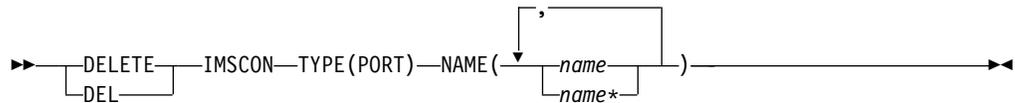
- "Environment"
- "Syntax" on page 401
- "Keywords" on page 401
- "Usage notes" on page 401
- "Equivalent WTOR and z/OS commands" on page 403
- "Output fields" on page 401
- "Return, reason, and completion codes" on page 402
- "Examples" on page 403

### Environment

The DELETE IMSCON command is applicable only to IMS Connect. To issue this command, the following conditions must be satisfied:

- IMS Connect must be active and configured to communicate with the Common Service Layer (CSL) Structured Call Interface (SCI).
- A type-2 command environment with Structured Call Interface (SCI) and Operations Manager (OM) must be active.

## Syntax



## Keywords

The following keywords are valid for the DELETE IMSCON TYPE(PORT) command.

### NAME

Specifies the port number in a 1- to 5-digit decimal number 1 - 65535. Wildcards are supported in the name.

## Usage notes

You can issue the DELETE IMSCON TYPE(PORT) command only through the Operations Manager (OM) API.

IMS Connect can process IMS Connect type-2 commands only if the IMSplex from which the commands were issued has a status of ACTIVE.

The listening port must be stopped (in NOTACTIVE status) before it can be deleted by use of the DELETE IMSCON TYPE(PORT) command.

IMS Connect resources that are deleted by use of this command will reappear after the next restart of IMS Connect, unless the corresponding configuration statement is also removed from the HWSCFGxx member of the IMS PROCLIB data set before the next restart.

## Output fields

### Short label

Contains the short label that is generated in the XML output.

### Long label

Contains the column heading displayed on the TSO SPOC screen.

### Keyword

Identifies the keyword on the command that caused the field to be generated. N/A (not applicable) is displayed for output fields that are always returned. *error* is displayed for output fields that are returned only in the case of an error.

### Meaning

Provides a brief description of the output field.

Table 201. Output fields for the DELETE IMSCON TYPE(PORT) command

Short label	Long label	Keyword	Meaning
CC	CC	N/A	Completion code that indicates whether IMS Connect was able to process the command for the specified resource. The completion code is always returned. See Table 203 on page 402.

Table 201. Output fields for the DELETE IMSCON TYPE(PORT) command (continued)

Short label	Long label	Keyword	Meaning
CCTXT	CCText	N/A	Completion code text that briefly explains the meaning of the nonzero completion code. This field is returned only for an error completion code.
MBR	MbrName	N/A	Identifier of the IMS Connect that built the output line. The identifier is always returned.
PORT	Port	N/A	The port number of the listening port to be deleted.

## Return, reason, and completion codes

The return and reason codes that can be returned as a result of the DELETE IMSCON TYPE(PORT) command are standard for all commands that are entered through the OM API.

The following table includes the return and reason codes and a brief explanation of the codes. The return or reason code returned for the command might also indicate an error from a CSL request.

Table 202. Return and reason codes for the DELETE IMSCON TYPE(PORT) command

Return code	Reason code	Meaning
X'00000000'	X'00000000'	The DELETE IMSCON TYPE(PORT) command completed successfully. The command output contains a line for each resource, accompanied by its completion code. For details, see Table 203.
X'0C00000C'	X'00003000'	The command was successful for some resources but failed for others. The command output contains a line for each resource, accompanied by its completion code.
X'0C00000C'	X'00003004'	The command was not successful for any resource. The command output contains a line for each resource, accompanied by its completion code.

Errors unique to the processing of this command are returned as completion codes. A completion code is returned for each action against an individual resource.

Table 203. Completion codes for the DELETE IMSCON TYPE(PORT) command

Completion code	Completion code text	Meaning
0		The DELETE IMSCON TYPE(PORT) command completed successfully for the resources.
10	No resources found	The port number is unknown to the client that is processing the request. The port number might have been typed in error or it might not be active at this time. If a wildcard was specified in the command, there were no matches for the name.
14	Resource is not stopped	The resource is not stopped. To delete this resource, it must be stopped first.
3B	Port type is invalid	The port type is invalid. SSL and local ports cannot be specified.

## Equivalent WTOR and z/OS commands

There are no equivalent WTOR and IMS Connect z/OS commands that perform similar functions as the DELETE IMSCON TYPE(PORT) command.

### Examples

#### *Example 1 for DELETE IMSCON TYPE(PORT) command*

TSO SPOC input:

```
DELETE IMSCON TYPE(PORT) NAME(8888)
```

TSO SPOC output:

Port	MbrName	CC
8888	HWS1	0

OM API input:

```
CMD(DEL IMSCON TYPE(PORT) NAME(8888))
```

OM API output:

```
<?xml version="1.0"?>
<!DOCTYPE imsout SYSTEM "imsout.dtd">
<imsout>
<ctl>
<omname>OM10M </omname>
<omvsn>1.7.0</omvsn>
<xmlvsn>20 </xmlvsn>
<statime>2013.283 22:48:48.511504</statime>
<stotime>2013.283 22:48:48.511747</stotime>
<staseq>CC17B16D1C21084E</staseq>
<stoseq>CC17B16D1C3030CE</stoseq>
<rqsttkn1>USRT011 10154848</rqsttkn1>
<rc>00000000</rc>
<rsn>00000000</rsn>
</ctl>
<cmd>
<master>HWS1 </master>
<userid>USRT011 </userid>
<verb>DEL </verb>
<kwd>IMSCON </kwd>
<input>DELETE IMSCON TYPE(PORT) NAME(8888) </input>
</cmd>
<cmdrsphdr>
<hdr s1b1="PORT" l1b1="Port" scope="LCL" sort="a" key="1" scroll="no"
  len="8" dtype="CHAR" align="left" skipb="no" />
<hdr s1b1="MBR" l1b1="MbrName" scope="LCL" sort="a" key="2" scroll="no"
  len="8" dtype="CHAR" align="left" skipb="no" />
<hdr s1b1="CC" l1b1="CC" scope="LCL" sort="n" key="0" scroll="yes"
  len="4" dtype="INT" align="right" skipb="no" />
<hdr s1b1="CCTXT" l1b1="CCText" scope="LCL" sort="n" key="0"
  scroll="yes" len="32" dtype="CHAR" align="left" skipb="yes" />
</cmdrsphdr>
<cmdrspdata>
<rsp>PORT(8888 ) MBR(HWS1 ) CC( 0) </rsp>
</cmdrspdata>
</imsout>
```

**Explanation:** Port 8888 is deleted.

#### *Example 2 for DELETE IMSCON TYPE(PORT) command*

TSO SPOC input:

```
DELETE IMSCON TYPE(PORT) NAME(8*,9999,7777)
```

TSO SPOC output:

Port	MbrName	CC	CCText
7777	HWS1	10	No resources found
8	HWS1	0	
8123	HWS1	14	Resource is not stopped
845	HWS1	0	
86	HWS1	0	
8888	HWS1	0	
9999	HWS1	0	

OM API input:

```
CMD(DEL IMSCON TYPE(PORT) NAME(8*,9999,7777))
```

OM API output:

```
<?xml version="1.0"?>
<!DOCTYPE imsout SYSTEM "imsout.dtd">
<imsout>
<ctl>
<omname>OM10M </omname>
<omvsn>1.7.0</omvsn>
<xmlvsn>20 </xmlvsn>
<statime>2013.283 22:55:36.537337</statime>
<stotime>2013.283 22:55:36.537599</stotime>
<staseq>CC17B2F23BCF9842</staseq>
<stoseq>CC17B2F23BDDFF242</stoseq>
<rqsttkn1>USRT011 10155536</rqsttkn1>
<rc>0200000C</rc>
<rsn>00003008</rsn>
<rsnmsg>CSLN054I</rsnmsg>
<rsntxt>None of the clients were successful.</rsntxt>
</ctl>
<cmderr>
<mbr name="HWS1 ">
<typ>IMSCON </typ>
<rc>0C00000C</rc>
<rsn>00003000</rsn>
<rsntxt>At least one request successful </rsntxt>
</mbr>
</cmderr>
<cmd>
<master>HWS1 </master>
<userid>USRT011 </userid>
<verb>DEL </verb>
<kwd>IMSCON </kwd>
<input>DELETE IMSCON TYPE(PORT) NAME(8*,9999,7777) </input>
</cmd>
<cmdrsphdr>
<hdr s1b1="PORT" l1b1="Port" scope="LCL" sort="a" key="1" scroll="no"
len="8" dtype="CHAR" align="left" skipb="no" />
<hdr s1b1="MBR" l1b1="MbrName" scope="LCL" sort="a" key="2" scroll="no"
len="8" dtype="CHAR" align="left" skipb="no" />
<hdr s1b1="CC" l1b1="CC" scope="LCL" sort="n" key="0" scroll="yes"
len="4" dtype="INT" align="right" skipb="no" />
<hdr s1b1="CCTXT" l1b1="CCText" scope="LCL" sort="n" key="0"
scroll="yes" len="32" dtype="CHAR" align="left" skipb="yes" />
</cmdrsphdr>
<cmdrspdata>
<rsp>PORT(8888 ) MBR(HWS1 ) CC( 0) </rsp>
<rsp>PORT(8123 ) MBR(HWS1 ) CC( 14) CCTXT(Resource is
not stopped) </rsp>
<rsp>PORT(845 ) MBR(HWS1 ) CC( 0) </rsp>
```

```

| <rsp>PORT(86 ) MBR(HWS1 ) CC( 0) </rsp>
| <rsp>PORT(8 ) MBR(HWS1 ) CC( 0) </rsp>
| <rsp>PORT(9999 ) MBR(HWS1 ) CC( 0) </rsp>
| <rsp>PORT(7777 ) MBR(HWS1 ) CC( 10) CCTXT(No resources
| found) </rsp>
| </cmdrspdata>
| </imsout>

```

**Explanation:** All ports whose port numbers start with 8 are deleted, except for 8123. Port 8123 cannot be deleted because it was not stopped. Port 9999 is also deleted. Port 7777 could not be found.

**Related reference:**

- [↗ CSLOMCMDB: command request \(System Programming APIs\)](#)
- [↗ IMS Connect Port Message Edit exit routine \(Exit Routines\)](#)
- [↗ IMSPLEX statement \(System Definition\)](#)
- [↗ ODACCESS statement \(System Definition\)](#)
- [↗ TCPIP statement \(System Definition\)](#)

---

## DELETE LE command

Use the DELETE command (with the LE keyword) to delete Language Environment (LE) runtime options that have previously been defined by the UPDATE command (with the LE keyword).

Subsections:

- “Environment”
- “Syntax”
- “Keywords” on page 406
- “Usage notes” on page 406
- “Output fields” on page 407
- “Return, reason, and completion codes” on page 407
- “Examples” on page 408

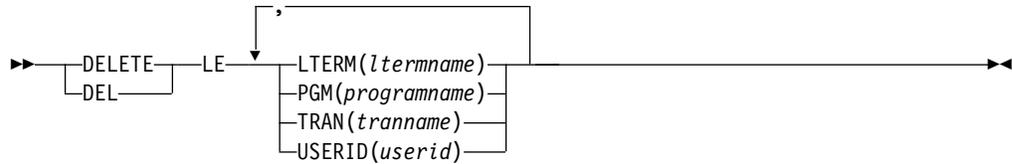
### Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

*Table 204. Valid environments for the DELETE command and keywords*

Command / Keywords	DB/DC	DBCTL	DCCTL
DELETE LE	X	X	X
LTERM	X	X	X
PGM	X	X	X
TRAN	X	X	X
USERID	X	X	X

### Syntax



## Keywords

### LTERM()

Specifies the 1 - 8 character name of the LTERMs that match the generic or wildcard parameter.

### PGM()

Specifies the 1 - 8 character name of the programs that match the generic or wildcard parameter.

### TRAN()

Specifies the 1 - 8 character name of the transactions that match the generic or wildcard parameter.

### USERID()

Specifies the 1 - 8 character name of the user IDs that match the generic or wildcard parameter.

## Usage notes

The delete can use filters on transaction code, LTERM name, user ID, or program name. Any combination of parameters can be used to qualify the application instance to which the runtime parameters are deleted. All matches found are deleted. The entry is logically deleted. The entry is then immediately available for reuse by the next UPDATE LE command. There is a small window of time where it is possible for an entry to be deleted and reused before LE uses it, that could cause LE to use the wrong set of parameters.

This command may be specified only through the Operations Manager API. The command syntax for this command is defined in XML and is available to automation programs that communicate with OM.

OM overrides the routing on the command and routes the command to all IMS systems in the IMSplex. The user-specified route list is ignored.

Rules for matching an entry which results in the delete of an entry:

- If a filter is specified in the command for a particular resource it must match the resource filter defined in the entry. The resource in the DELETE LE command can be specified with wildcards as defined previously.
- A resource filter that is not specified on a DELETE LE command will match on any filter for the specific resource defined in the entry. A non-specified filter is treated as a wildcard. For instance if the LTERM filter is not specified on a DELETE LE command it will match on any LTERM resource defined in an entry, as if LTERM(\*) was specified on the command.

If an existing set of LE runtime options are found for the specified TRAN, LTERM, USERID, or PGM, the existing parameter string is deleted. The string is deleted only when the specified filters are an exact match for the existing entry. The entry is logically deleted. The entry is available for reuse on the next UPDATE LE command.

The following keywords support a generic or wildcard character. A generic parameter is a 1 - 8 character alphanumeric name that includes one or more special characters and an asterisk or a percent sign. An asterisk can be replaced by a zero or more characters to create a valid resource name. A percent sign can be replaced by exactly one character to create a valid resource name.

**Requirement:** At least one of the resource filters (LTERM, PGM, TRAN, or USERID) must be specified on the UPDATE LE command.

## Output fields

The following table shows the output fields for an DELETE LE command. The columns in the table are as follows:

### Short label

Contains the short label generated in the XML output.

### Keyword

Identifies the keyword on the command that caused the field to be generated. N/A appears for output fields that are always returned.

### Meaning

Provides a brief description of the output field.

*Table 205. Output fields for the DELETE LE command*

Short label	Keyword	Meaning
CC	N/A	Completion code for the line of output. Completion code is always returned.
LTRM	LTERM	LTERM name affected by the DELETE.
MBR	N/A	IMSpIex member (IMS identifier) that built the output line. Member name is always returned.
PGM	PGM	Program name affected by the DELETE.
TRAN	TRAN	Transaction name affected by the DELETE.
UID	USERID	Userid affected by the DELETE.

## Return, reason, and completion codes

The following table includes the return and reason codes and a brief explanation of the codes. The return or reason code returned for the command might also indicate an error from a CSL request.

*Table 206. Return and reason codes for the DELETE LE command*

Return code	Reason code	Meaning
X'00000000'	X'00000000'	The DELETE LE command completed successfully.
X'00000008'	X'00002008'	No keywords were specified on the command. At least one keyword is required.
X'00000008'	X'0000200C'	No resources found to delete. Either the entry was previously deleted or a keyword filter was typed incorrectly.
X'00000008'	X'00002014'	An invalid character was specified in the filter name.
X'00000010'	X'00004040'	The parameter override header has not been initialized. Retry the command after restart is complete.
X'00000014'	X'00005000'	Unable to get storage from IMODULE GETSTOR.

Table 206. Return and reason codes for the DELETE LE command (continued)

Return code	Reason code	Meaning
X'00000014'	X'00005010'	Unable to obtain latch.
X'00000014'	X'00005FFF'	Internal IMS Error - should not occur.

The following table contains the completion code that can be returned on a DELETE LE command and the meaning of the code.

Table 207. Completion code for the DELETE LE command

Completion code	Meaning
0	The DELETE LE command completed successfully for the specified resource.

## Examples

The following are examples of the DELETE LE command:

### Example 1 for DELETE LE command

For this example, the following table shows the parameter override table entries prior to any DELETE LE commands being processed.

Table 208. Parameter override table entries for DELETE LE example

Entry#	TRAN	LTERM	USERID	PROGRAM	LERUNOPTS
1	PART			DFSSAM02	aaaa
2	PART		BETTY		bbbb
3	PART	TERM1			cccc
4				DFSSAM02	dddd
5	PART	TERM1	BARBARA		eeee
6	PART	TERM1	BOB		ffff
7		TERM2	BETTY		gggg
8	PART	TERM2			iiii

The following table shows the resulting actions when different filters are specified on a DELETE LE command.

Table 209. Example filters and resulting actions for DELETE LE command

Filters active on command	Resulting action
TRAN(PART)	Deletes entries #1, 2, 3, 5, 6, 8
TRAN(PART) LTERM(TERM1)	Deletes entries #3, 5, 6
LTERM(TERM2) USERID(BETTY)	Deletes entry #7
TRAN(PART) LTERM(TERM1) USERID(BETTY)	Does not delete any entries
TRAN(PART) LTERM(TERM*)	Deletes entries #3, 5, 6, 8
USERID(B*)	Deletes entries #2, 5, 6, 7

### Example 2 for DELETE LE command

The following TSO SPOC and OM API input and output sets up the scenario for the DELETE LE command examples. The UPDATE commands are used to add entries to the table; the QUERY command shows the resulting entries.

TSO SPOC input:

```
UPD LE TRAN(IAPMDI27) USERID(USRT003) SET(LERUNOPTS(cccc))
UPD LE TRAN(IAPMDI27) USERID(USRT001) SET(LERUNOPTS(bbbb))
UPD LE TRAN(IAPMDI26) USERID(USRT001) SET(LERUNOPTS(aaaa))
QRY LE SHOW(ALL)
```

TSO SPOC output (for QUERY command):

MbrName	CC	Trancode	Lterm	Userid	Program	LERunOpts
SYS3	0	IAPMDI26		USRT001		AAAA
SYS3	0	IAPMDI27		USRT001		BBBB
SYS3	0	IAPMDI27		USRT003		CCCC

OM API input:

```
CMD(UPD LE TRAN(IAPMDI27) USERID(USRT003) SET(LERUNOPTS(cccc)))
CMD(UPD LE TRAN(IAPMDI27) USERID(USRT001) SET(LERUNOPTS(bbbb)))
CMD(UPD LE TRAN(IAPMDI26) USERID(USRT001) SET(LERUNOPTS(aaaa)))
CMD(QRY LE SHOW(ALL))
```

OM API output (for QUERY command):

```
<imsout>
<ctl>
<omname>OM10M </omname>
<omvsn>1.1.0</omvsn>
<xmlvsn>1 </xmlvsn>
<statime>2002.163 18:10:14.974557</statime>
<stotime>2002.163 18:10:14.974985</stotime>
<staseq>B7C4CD747AC5DC85</staseq>
<stoseq>B7C4CD747AE09D25</stoseq>
<rqsttkn1>USRT002 10111014</rqsttkn1>
<rc>00000000</rc>
<rsn>00000000</rsn>
</ctl>
<cmd>
<master>SYS3 </master>
<userid>USRT002 </userid>
<verb>QRY </verb>
<kwd>LE </kwd>
<input>QRY LE SHOW(ALL) </input>
</cmd>
<cmdrsphdr>
<hdr s1bl="MBR" l1bl="MbrName" scope="LCL" sort="A" key="1" scroll="NO" len="8"
  dtype="CHAR" align="left" />
<hdr s1bl="CC" l1bl="CC" scope="LCL" sort="N" key="0" scroll="YES" len="4"
  dtype="INT" align="right" />
<hdr s1bl="TRAN" l1bl="Trancode" scope="LCL" sort="N" key="0" scroll="YES" len="8"
  dtype="CHAR" align="left" />
<hdr s1bl="LTRM" l1bl="Lterm" scope="LCL" sort="N" key="0" scroll="YES" len="8"
  dtype="CHAR" align="left" />
<hdr s1bl="UID" l1bl="Userid" scope="LCL" sort="N" key="0" scroll="YES" len="8"
  dtype="CHAR" align="left" />
<hdr s1bl="PGM" l1bl="Program" scope="LCL" sort="N" key="0" scroll="YES" len="8"
  dtype="CHAR" align="left" />
<hdr s1bl="PRM" l1bl="LERunOpts" scope="LCL" sort="N" key="0" scroll="YES" len="*"
  dtype="CHAR" align="left" />
</cmdrsphdr>
<cmdrspdata>
<rsp>MBR(SYS3 ) CC( 0) TRAN(IAPMDI26) LTRM( ) UID(USRT001) PGM( )
PRM(AAAA ) </rsp>
<rsp>MBR(SYS3 ) CC( 0) TRAN(IAPMDI27) LTRM( ) UID(USRT001) PGM( )
```

```

PRM(BBBB ) </rsp>
<rsp>MBR(SYS3 ) CC( 0) TRAN(IAPMDI27) LTRM( ) UID(USRT003 ) PGM( )
PRM(CCCC ) </rsp>
</cmdrspdata>
</imsout>

```

### Example 3 for DELETE LE command

TSO SPOC input:

```
DEL LE TRAN(IAPMDI27) USERID(USRT001)
```

TSO SPOC output:

MbrName	CC	Trancode	Lterm	Userid	Program
SYS3	0	IAPMDI27		USRT001	

OM API input:

```
CMD(DEL LE TRAN(IAPMDI27) USERID(USRT001))
```

OM API output:

```

<imsout>
<ctl>
<omname>OM10M </omname>
<omvsn>1.1.0</omvsn>
<xmlvsn>1 </xmlvsn>
<statime>2002.163 18:12:21.932813</statime>
<stotime>2002.163 18:12:21.933334</stotime>
<staseq>B7C4CDED8E70D786</staseq>
<stoseq>B7C4CDED8E916246</stoseq>
<rqsttkn1>USRT002 10111221</rqsttkn1>
<rc>00000000</rc>
<rsn>00000000</rsn>
</ctl>
<cmd>
<master>SYS3 </master>
<userid>USRT002 </userid>
<verb>DEL </verb>
<kwd>LE </kwd>
<input>DEL LE TRAN(IAPMDI27) USERID(USRT001) </input>
</cmd>
<cmdrsphdr>
<hdr s1bl="MBR" l1bl="MbrName" scope="LCL" sort="A" key="1" scroll="NO" len="8"
dtype="CHAR" align="left" />
<hdr s1bl="CC" l1bl="CC" scope="LCL" sort="N" key="0" scroll="YES" len="4"
dtype="INT" align="right" />
<hdr s1bl="TRAN" l1bl="Trancode" scope="LCL" sort="N" key="0" scroll="YES" len="8"
dtype="CHAR" align="left" />
<hdr s1bl="LTRM" l1bl="Lterm" scope="LCL" sort="N" key="0" scroll="YES" len="8"
dtype="CHAR " align="left" />
<hdr s1bl="UID" l1bl="Userid" scope="LCL" sort="N" key="0" scroll="YES" len="8"
dtype="CHAR " align="left" />
<hdr s1bl="PGM" l1bl="Program" scope="LCL" sort="N" key="0" scroll="YES" len="8"
dtype="CHAR" align="left" />
</cmdrsphdr>
<cmdrspdata>
<rsp>MBR(SYS3 ) CC( 0) TRAN(IAPMDI27) LTRM( ) UID(USRT001 ) PGM( )
</rsp>
</cmdrspdata>
</imsout>

```

Explanation: This command specifies two filters- transaction and userid. A table entry is deleted if both of these filters are defined and match what was specified in the command. This command deletes one entry from the table. That entry is shown in the command output.

#### Example 4 for DELETE LE command

TSO SPOC input:

```
DEL LE USERID(USRT00%)
```

TSO SPOC output:

MbrName	CC	Trancode	Lterm	Userid	Program
SYS3	0	IAPMDI26		USRT001	
SYS3	0	IAPMDI27		USRT003	

OM API input:

```
CMD(DEL LE USERID(USRT00%))
```

OM API output:

```
<imsout>
<ctl>
<omname>OM10M </omname>
<omvsn>1.1.0</omvsn>
<xmllvsn>1 </xmllvsn>
<statime>2002.163 18:19:45.616468</statime>
<stotime>2002.163 18:19:45.616972</stotime>
<staseq>B7C4CF94AFA54304</staseq>
<stoseq>B7C4CF94AFC4C704</stoseq>
<rqsttkn1>USRT002 10111945</rqsttkn1>
<rc>00000000</rc>
<rsn>00000000</rsn>
</ctl>
<cmd>
<master>SYS3 </master>
<userid>USRT002 </userid>
<verb>DEL </verb>
<kwd>LE </kwd>
<input>DEL LE USERID(USRT00%) </input>
</cmd>
<cmdrsphdr>
<hdr s1b1="MBR" l1b1="MbrName" scope="LCL" sort="A" key="1" scroll="NO" len="8"
dtype="CHAR" align="left" />
<hdr s1b1="CC" l1b1="CC" scope="LCL" sort="N" key="0" scroll="YES" len="4"
dtype="INT" align="right" />
<hdr s1b1="TRAN" l1b1="Trancode" scope="LCL" sort="N" key="0" scroll="YES" len="8"
dtype="CHAR" align="left" />
<hdr s1b1="LTRM" l1b1="Lterm" scope="LCL" sort="N" key="0" scroll="YES" len="8"
dtype="CHAR " align="left" />
<hdr s1b1="UID" l1b1="Userid" scope="LCL" sort="N" key="0" scroll="YES" len="8"
dtype="CHAR " align="left" />
<hdr s1b1="PGM" l1b1="Program" scope="LCL" sort="N" key="0" scroll="YES" len="8"
dtype="CHAR" align="left" />
</cmdrsphdr>
<cmdrspdata>
<rsp>MBR(SYS3 ) CC( 0) TRAN(IAPMDI26) LTRM( ) UID(USRT001 ) PGM( )
</rsp>
<rsp>MBR(SYS3 ) CC( 0) TRAN(IAPMDI27) LTRM( ) UID(USRT003 ) PGM( )
</rsp>
</cmdrspdata>
</imsout>
```

Explanation: This command specifies only one filter, userid. A wildcard is used, so any table entry where the USERID filter matches the string will be deleted. The command output shows that both of the remaining table entries were deleted.

**Related concepts:**

➡ How to interpret CSL request return and reason codes (System Programming APIs)

**Related reference:**

Chapter 2, "Command keywords and their synonyms," on page 63

---

## DELETE LTERM command

Use the DELETE LTERM command to delete a Multiple Systems Coupling (MSC) remote LTERM.

Subsections:

- "Environment"
- "Syntax"
- "Keywords"
- "Usage notes" on page 413
- "Output fields" on page 413
- "Return and reason codes" on page 414
- "Completion codes" on page 415
- "Examples" on page 415

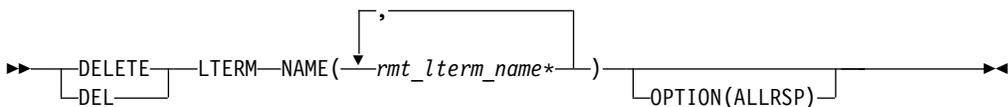
### Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

*Table 210. Valid environments for the DELETE LTERM command and keywords*

Command / Keywords	DB/DC	DBCTL	DCCTL
DELETE LTERM NAME() OPTION()	X		X

### Syntax



### Keywords

The following keywords are valid for the DELETE LTERM command:

#### NAME

Specify one or more 1-8 character names of MSC remote LTERMs. Wildcards are supported in the name. Examples of valid names are NAME(\*) and NAME(abc\*). If the NAME parameter specified is a specific or wildcard name, command responses are returned for all the resource names that are processed. For NAME(\*), command responses are returned only for the resource names that resulted in an error. OPTION(ALLRSP) can be specified with NAME(\*) to obtain the command responses for all the resource names that are processed.

#### OPTION

Specifies options for the command.

## ALLRSP

Indicates that the response lines are to be returned for all resources that are processed on the command. The default action is to return response lines only for the resources that resulted in an error. It is valid only with NAME(\*). ALLRSP is ignored for other NAME values.

## Usage notes

This command can be issued only through the OM API. This command applies to DB/DC systems and DCCTL systems. The DELETE LTERM command applies only to remote MSC LTERMs and is valid only when both MSC and the dynamic definition of MSC resources are enabled (MSC=Y is specified in the startup parameters and MSCRSCS=DYN is specified in the MSC section of the DFSDFxxx member).

The DELETE LTERM command is not valid on the XRF alternate, RSR tracker, or FDBR region.

This command is recoverable.

Before you attempt to delete an MSC remote LTERM, make sure that the remote LTERM is not in use. If the remote LTERM is in use, the delete fails. Issue the QUERY LTERM NAME(*ltermname*) SHOW(STATUS) command to check the status of the LTERM.

In a sysplex environment with multiple IMS systems, the delete might succeed on some IMS systems and fail on others.

Changes that are made to MSC resources by the DELETE command are logged to save them across warm and emergency restarts. The changes are not saved across a cold start of IMS unless the MSC resources are also deleted from the stage-1 system definition macro input.

If your MSC resources are defined by using stage-1 system definition macros, delete the resource definition by removing the macro that defines the resource from the stage-1 system definition input and regenerate the MSC resources into the DFSCLL3x member in the IMS.SDFSRESL data set.

## Output fields

The following table shows the DELETE LTERM output fields. The columns in the table are as follows:

### Short label

Contains the short label that is generated in the XML output.

### Long label

Contains the long label that is generated in the XML output.

### Keyword

Identifies keyword on the command that caused the field to be generated. N/A appears for output fields that are always returned. *error* appears for output fields that are returned only in case of an error.

### Meaning

Provides a brief description of the output field.

Table 211. Output fields for the DELETE LTERM command

Short label	Long label	Keyword	Meaning
CC	CC	N/A	Completion code.
CCTXT	CCText	<i>error</i>	Completion code text that briefly explains the nonzero completion code.
ERRT	ErrorText	<i>error</i>	Error text that provides diagnostic information. Error text can be returned for a nonzero completion code and further explains the completion code.
MBR	MbrName	N/A	IMSpIex member that built the output line.
LTERM	Lterm	N/A	The name of the remote logical terminal. The remote logical terminal name is always returned.

## Return and reason codes

The following table includes the return and reason codes and a brief explanation of the codes. The return or reason code returned for the command might also indicate an error from a CSL request.

Table 212. Return and reason codes for the DELETE LTERM command

Return code	Reason code	Meaning
X'00000000'	X'00000000'	The command completed successfully. The command output contains a line for each resource, which is accompanied by its completion code. If NAME(*) is specified without OPTION(ALLRSP), no output lines are returned. See the completion code table for details.
X'0000000C'	X'00003000'	Command was successful for some resources but failed for others. The command output contains a line for each resource, which is accompanied by its completion code. If NAME(*) is specified without OPTION(ALLRSP), only resources with nonzero completion codes are returned. See the completion code table for details.
X'0000000C'	X'00003004'	Command was not successful for any of the resources. The command output contains a line for each resource, which is accompanied by its completion code. See the completion code table for details.
X'00000010'	X'0000400C'	Command is not valid on the XRF alternate.
X'00000010'	X'00004014'	Command is not valid on the RSR tracker.
X'00000010'	X'0000402C'	Command is not valid because MSC is not enabled in the IMS system.
X'00000010'	X'0000402D'	Command is not allowed because dynamic definition is not enabled for MSC. MSCRSCS=DYN is not specified in the MSC section of the DFSDFxxx member of the IMS PROCLIB data set.
X'00000014'	X'00005004'	DFSOCMD response buffer could not be obtained.
X'00000014'	X'00005008'	DFSPOOL storage could not be obtained.
X'00000014'	X'00005010'	Unable to obtain latch.

## Completion codes

Errors unique to the processing of this command are returned as completion codes. The following table includes an explanation of the completion codes.

Table 213. Completion codes for the DELETE LTERM command

Completion code	Completion code text	Meaning
0		The command completed successfully for the resource.
10	NO RESOURCES FOUND	The resource name is invalid or the value that is specified on the NAME keyword does not match the name of any resource within this resource type.
17	ANOTHER CMD IN PROGRESS	Another command (such as DELETE or UPDATE) is in progress for this LTERM. This could also mean this command, if the resource is specified by more than one specific or wildcard parameter.
48	NOT ALLOWED FOR IMS RESOURCE	The resource name is invalid as a remote lterm. The resource is an IMS-defined resource.
90	INTERNAL ERROR	An IMS system service failure.

## Examples

The following are examples of the DELETE LTERM command:

### Example 1 for DELETE LTERM command

In the following example, the DELETE LTERM command is issued to delete MSC remote LTERMs LTERM20B, BADNAME, and LTERM23C. The delete is successful for remote LTERMs LTERM20B and LTERM23C, as shown by the completion code 0. The delete fails for BADNAME because no remote LTERM is named BADNAME.

TSO SPOC input:

```
DELETE LTERM NAME(LTERM20B,BADNAME,H0G5*,LTERM23C)
```

TSO SPOC output:

```
Lterm   MbrName   CC CCText
BADNAME IMS1      10 NO RESOURCES FOUND
H0G50   IMS1       0
H0G51   IMS1       0
H0G52   IMS1       0
LTERM20B IMS1      0
LTERM23C IMS1      0
```

OM API input:

```
CMD(DEL LTERM NAME(LTERM20B,BADNAME,H0G5*,LTERM23C))
```

OM API output:

```
<imsout>
<ctl>
<omname>OM10M </omname>
```

```

<omvsn>1.7.0</omvsn>
<xmlvsn>20 </xmlvsn>
<statime>2014.268 16:10:13.937131</statime>
<stotime>2014.268 16:10:13.938009</stotime>
<staseq>CDCF651DB8FEBD05</staseq>
<stoseq>CDCF651DB9359185</stoseq>
<rqsttkn1>USRT011 10091013</rqsttkn1>
<rc>0200000C</rc>
<rsn>00003008</rsn>
<rsnmsg>CSLN054I</rsnmsg>
<rsntxt>None of the clients were successful.</rsntxt>
</ctl>
<cmderr>
<mbr name="IMS1 ">
<typ>IMS </typ>
<styp>DBDC </styp>
<rc>0000000C</rc>
<rsn>00003000</rsn>
<rsntxt>At least one request successful</rsntxt>
</mbr>
</cmderr>
<cmd>
<master>IMS1 </master>
<userid>USRT011 </userid>
<verb>DEL </verb>
<kwd>LTERM </kwd>
<input>DELETE LTERM NAME(LTERM20B,BADNAME,H0G5*,LTERM23C) </input>
</cmd>
<cmdrsphdr>
<hdr s1b1="LTERM" l1b1="Lterm" scope="LCL" sort="a" key="1" scroll="n
o" len="8" dtype="CHAR" align="left" />
<hdr s1b1="MBR" l1b1="MbrName" scope="LCL" sort="a" key="2" scroll="no"
len="8" dtype="CHAR" align="left" />
<hdr s1b1="CC" l1b1="CC" scope="LCL" sort="a" key="3" scroll="yes"
len="4" dtype="INT" align="right" skipb="no" />
<hdr s1b1="CCTXT" l1b1="CCText" scope="LCL" sort="n" key="0"
scroll="yes" len="*" dtype="CHAR" skipb="yes" align="left" />
<hdr s1b1="ERRT" l1b1="ErrorText" scope="LCL" sort="n" key="0"
scroll="yes" len="*" dtype="CHAR" skipb="yes" align="left" />
</cmdrsphdr>
<cmdrspdata>
<rsp>LTERM(LTERM20B) MBR(IMS1 ) CC( 0) </rsp>
<rsp>LTERM(BADNAME ) MBR(IMS1 ) CC( 10) CCTXT(NO RESOURCES FOUND)
</rsp>
<rsp>LTERM(H0G52 ) MBR(IMS1 ) CC( 0) </rsp>
<rsp>LTERM(LTERM23C) MBR(IMS1 ) CC( 0) </rsp>
<rsp>LTERM(H0G51 ) MBR(IMS1 ) CC( 0) </rsp>
<rsp>LTERM(H0G50 ) MBR(IMS1 ) CC( 0) </rsp>
</cmdrspdata>
</imsout>

```

---

## DELETE MSLINK command

Use the DELETE MSLINK command to delete a Multiple Systems Coupling (MSC) logical link from an online IMS system.

Subsections:

- “Environment” on page 417
- “Syntax” on page 417
- “Keywords” on page 417
- “Usage notes” on page 417
- “Output fields” on page 418
- “Return and reason codes” on page 419

- “Completion codes” on page 419
- “Examples” on page 420

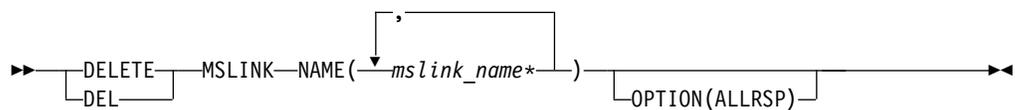
## Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

Table 214. Valid environments for the DELETE MSLINK command and keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
DELETE MSLINK NAME() OPTION()	X		X

## Syntax



## Keywords

The following keywords are valid for the DELETE MSLINK command:

### NAME

Specify one or more 1 - 8 character names of MSC logical links. Wildcards are supported in the name. Examples of valid names are NAME(\*) and NAME(abc\*). If the NAME parameter specified is a specific or wildcard name, command responses are returned for all the resource names that are processed. For NAME(\*), command responses are returned only for the resource names that resulted in an error. OPTION(ALLRSP) can be specified with NAME(\*) to obtain the command responses for all the resource names that are processed.

### OPTION

Specifies additional functions to be completed along with the command.

### ALLRSP

Indicates that the response lines are to be returned for all resources that are processed on the command. The default action is to return response lines only for the resources that resulted in an error. ALLRSP is valid only with NAME(\*). ALLRSP is ignored for other NAME values.

## Usage notes

Before attempting to delete a logical link, make sure that the logical link is not in use and that any MSC logical link paths that reference the logical link are deleted or reassigned to another logical link. If the logical link is in use or is referenced by a logical link path, the delete fails.

- To check the status of the link, issue the QUERY MSLINK NAME(*mslinknm*) SHOW(STATUS) command.
- To see which logical link paths (MSNAMEs) are assigned to the link, issue the QUERY MSLINK NAME(*mslinknm*) SHOW(MSNAME).

Changes that are made to MSC resources by the DELETE command are logged to save them across warm and emergency restarts. The changes are not saved across a cold start of IMS unless the MSC resources are also deleted from the stage-1 system definition macro input.

If your MSC resources are defined by using stage-1 system definition macros, delete the resource definition by removing the macro that defines the resource from the stage-1 system definition input and regenerate the MSC resources into the DFSCLL3x member in the IMS.SDFSRESL data set.

The DELETE MSLINK command is valid only when both MSC and the dynamic definition of MSC resources are enabled (MSC=Y is specified in the startup parameters and MSCRSCS=DYN is specified in the MSC section of the DFSDFxxx member). This command is recoverable.

This command can be issued only through the OM API. This command applies to DB/DC systems and DCCTL systems.

In a Sysplex environment with multiple IMS systems, the delete might succeed on some IMS systems and fail on others.

The DELETE MSLINK command is not valid on the XRF alternate or FDBR region.

The DELETE MSLINK command is not valid on the XRF alternate, RSR tracker, or FDBR region.

## Output fields

The following table shows the DELETE MSLINK output fields. The columns in the table are as follows:

### Short label

Contains the short label generated in the XML output.

### Long label

Contains the long label generated in the XML output.

### Keyword

Identifies keyword on the command that caused the field to be generated. N/A appears for output fields that are always returned. *error* appears for output fields that are returned only in case of an error.

### Meaning

Provides a brief description of the output field.

Table 215. Output fields for the DELETE MSLINK command

Short label	Long label	Keyword	Meaning
CC	CC	N/A	Completion code.
CCTXT	CCText	<i>error</i>	Completion code text that briefly explains the nonzero completion code.
ERRT	ErrorText	<i>error</i>	Error text that provides diagnostic information. Error text can be returned for a nonzero completion code and further explains the completion code.
MBR	MbrName	N/A	IMSpIex member that built the output line.
MSL	MSLink	N/A	MSC logical link name.

## Return and reason codes

The following table includes the return and reason codes and a brief explanation of the codes. The return or reason code returned for the command might also indicate an error from a CSL request.

*Table 216. Return and reason codes for the DELETE MSLINK command*

Return code	Reason code	Meaning
X'00000000'	X'00000000'	Command completed successfully. The command output contains a line for each resource, accompanied by its completion code. If NAME(*) is specified without OPTION(ALLRSP), no output lines are returned. See the completion code table for details.
X'0000000C'	X'00003000'	Command was successful for some resources but failed for others. The command output contains a line for each resource, accompanied by its completion code. If NAME(*) is specified without OPTION(ALLRSP), only resources with nonzero completion codes are returned. See the completion code table for details.
X'0000000C'	X'00003004'	Command was not successful for any of the resources. The command output contains a line for each resource, accompanied by its completion code. See the completion code table for details.
X'00000010'	X'0000400C'	Command is not valid on the XRF alternate.
X'00000010'	X'00004014'	Command is not valid on the RSR tracker.
X'00000010'	X'0000402C'	Command is not valid because MSC is not enabled in the IMS system.
X'00000010'	X'0000402D'	Command is not allowed because dynamic definition is not enabled for MSC. MSCRSCS=DYN is not specified in the MSC section of the DFSDFxxx member of the IMS PROCLIB data set.
X'00000014'	X'00005004'	DFSOCMD response buffer could not be obtained.
X'00000014'	X'00005008'	DFSPOOL storage could not be obtained.
X'00000014'	X'00005010'	Unable to obtain latch.

## Completion codes

Errors unique to the processing of this command are returned as completion codes. The following table includes an explanation of the completion codes.

*Table 217. Completion codes for the DELETE MSLINK command*

Completion code	Completion code text	Meaning
0		The command completed successfully for the resource.
10	NO RESOURCES FOUND	The resource name is invalid or the value that is specified on the NAME keyword does not match the name of any resource within this resource type.
90	INTERNAL ERROR	An IMS system failure occurred.

Table 217. Completion codes for the DELETE MSLINK command (continued)

Completion code	Completion code text	Meaning
20A	REFERENCED BY MSNAME	The logical link cannot be deleted because it is referenced by one or more logical link paths (MSNAMEs).

## Examples

The following are examples of the DELETE MSLINK command:

### Example 1 for DELETE MSLINK command

In the following example, the DELETE MSLINK command is used to try to delete MSC logical links LNK12M01, BADNAME, and LNKM0001. The delete is successful for logical link LNKM0001, as shown by the completion code 0. The delete fails for BADNAME because there is no logical link that is named BADNAME. The delete fails for logical link LNK12M01, because it is referenced by MSNAMEs LINK13M2 and LINK12M1.

TSO SPOC input:

```
DELETE MSLINK NAME(LNK12M01,BADNAME, LNKM0001)
```

TSO SPOC output:

```
MSLink  MbrName  CC CText
LNKM0001 IMS1      0
BADNAME IMS1      10 NO RESOURCES FOUND
LNK12M01 IMS1     20A REFERENCED BY MSNAME LINK13M2
LNK12M01 IMS1     20A REFERENCED BY MSNAME LINK12M1
```

OM API input:

```
CMD(DEL MSLINK NAME(LNK12M01,BADNAME, LNKM0001))
```

OM API output:

```
<ctl>
<omname>OM10M  </omname>
<omvsn>1.7.0</omvsn>
<xmlvsn>20  </xmlvsn>
<statime>2015.019 22:23:01.853686</statime>
<stotime>2015.019 22:23:01.890630</stotime>
<staseq>CE6190CAE61F656C</staseq>
<stoseq>CE6190CAEF246DE2</stoseq>
<rqsttkn1>USRT011 10142301</rqsttkn1>
<rc>0200000C</rc>
<rsn>00003008</rsn>
<rsnmsg>CSLN054I</rsnmsg>
<rsntxt>None of the clients were successful.</rsntxt>
</ctl>
<cmderr>
<mbr name="IMS1  ">
<typ>IMS  </typ>
<styp>DBDC  </styp>
<rc>0000000C</rc>
<rsn>00003000</rsn>
<rsntxt>At least one request successful</rsntxt>
</mbr>
</cmderr>
<cmd>
<master>IMS1  </master>
```

```

<userid>USRT011 </userid>
<verb>DEL </verb>
<kwd>MSLINK </kwd>
<input>DELETE MSLINK NAME(LNK12M01,BADNAME,LNKM0001) </input>
</cmd>
<cmdrsphdr>
<hdr s1b1="MSL" l1b1="MSLink" scope="LCL" sort="a" key="1" scroll="no "
 len="8" dtype="CHAR" align="left" />
<hdr s1b1="MBR" l1b1="MbrName" scope="LCL" sort="a" key="2" scroll="no"
 len="8" dtype="CHAR" align="left" />
<hdr s1b1="CC" l1b1="CC" scope="LCL" sort="n" key="0" scroll="yes"
 len="4" dtype="INT" align="right" skipb="no" />
<hdr s1b1="CCTXT" l1b1="CCText" scope="LCL" sort="n" key="0"
 scroll="yes" len="*" dtype="CHAR" skipb="yes" align="left" />
<hdr s1b1="ERRT" l1b1="ErrorText" scope="LCL" sort="n" key="0"
 scroll="yes" len="*" dtype="CHAR" skipb="yes" align="left" />
</cmdrsphdr>
<cmdrspdata>
<rsp>MSL(LNK12M01) MBR(IMS1 ) CC( 20A) CCTXT(REFERENCED BY MSNAME)
 ERRT(LINK13M2) </rsp>
<rsp>MSL(BADNAME ) MBR(IMS1 ) CC( 10) CCTXT(NO RESOURCES FOUND)
</rsp>
<rsp>MSL(LNKM0001) MBR(IMS1 ) CC( 0) </rsp>
<rsp>MSL(LNK12M01) MBR(IMS1 ) CC( 20A) CCTXT(REFERENCED BY MSNAME)
 ERRT(LINK12M1) </rsp>
</cmdrspdata>
</imsout>

```

---

## DELETE MSNAME command

Use the DELETE MSNAME command to delete a Multiple Systems Coupling (MSC) logical link path from an online IMS system.

Subsections:

- “Environment”
- “Syntax”
- “Keywords” on page 422
- “Usage notes” on page 422
- “Output fields” on page 423
- “Return and reason codes” on page 423
- “Completion codes” on page 424
- “Examples” on page 426

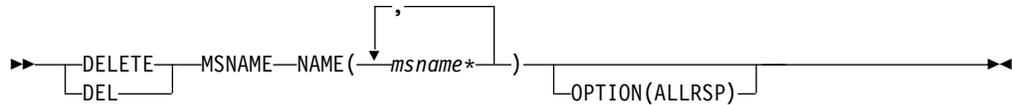
### Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

*Table 218. Valid environments for the DELETE MSNAME command and keywords*

Command / Keywords	DB/DC	DBCTL	DCCTL
DELETE MSNAME NAME() OPTION()	X		X

### Syntax



## Keywords

The following keywords are valid for the DELETED MSNAME command:

### NAME

Specify one or more 1 - 8 character names of MSC logical link paths. Wildcards are supported in the name. Examples of valid names are NAME(\*) and NAME(abc\*). If the NAME parameter specified is a specific or wildcard name, command responses are returned for all the resource names that are processed. For NAME(\*), command responses are returned only for the resource names that resulted in an error. OPTION(ALLRSP) can be specified with NAME(\*) to obtain the command responses for all the resource names that are processed.

### OPTION

Specifies options for the command.

#### ALLRSP

Indicates that the response lines are to be returned for all resources that are processed on the command. The default action is to return response lines only for the resources that result in an error. ALLRSP is valid only with NAME(\*). ALLRSP is ignored for other NAME values.

## Usage notes

Before you delete a logical link path, make sure that the logical link path is not in use and that any LTERMs or transactions that reference the MSNAME of the logical link path are deleted or reassigned to another MSNAME. If the logical link path is in use or is referenced by an LTERM or a transaction, the delete fails.

- Issue the QUERY MSNAME NAME(msname) SHOW(STATUS) command to check the status of the link.
- Issue the QUERY TRAN NAME(\*) SHOW(MSNAME) to see if any transactions reference the logical link path.
- Issue the QUERY LTERM NAME(\*) SHOW(MSNAME) to see if any LTERMs are assigned to the logical link path.

Changes that are made to MSC resources by the DELETED command are logged to save them across warm and emergency restarts. The changes are not saved across a cold start of IMS unless the MSC resources are also deleted from the stage-1 system definition macro input.

If your MSNAME resources are defined by using stage-1 systems definition macros, delete the MSNAME resource definition by removing the MSNAME macros that define the MSNAME resources from the stage-1 system definition input and regenerate the MSC MSNAME resources into the DFSCLC0x member in the IMS.SDFSRESL data set.

This command can be issued only through the OM API. This command applies to DB/DC systems and DCCTL systems.

In a Sysplex environment with multiple IMS systems, the delete might succeed on some IMS systems and fail on others.

The DELETE MSNAME command is valid only when both MSC and the dynamic definition of MSC resources are enabled (MSC=Y is specified in the startup parameters and MSCRSCS=DYN is specified in the MSC section of the DFSDFxxx member). This command is recoverable.

The DELETE MSNAME command is not valid on the XRF alternate, RSR tracker, or FDBR region.

In a shared queues environment, the DELETE MSNAME command deletes the MSC MSNAME (logical link path) locally in the IMS that has a physical link outside the IMSplex, and causes the dynamic MSNAME to be deleted in all the other IMSs in the shared queues group.

## Output fields

The following table shows the DELETE MSNAME output fields. The columns in the table are as follows:

### Short label

Contains the short label that is generated in the XML output.

### Long label

Contains the long label that is generated in the XML output.

### Keyword

Identifies keyword on the command that caused the field to be generated. N/A appears for output fields that are always returned. *error* appears for output fields that are returned only in case of an error.

### Meaning

Provides a brief description of the output field.

Table 219. Output fields for the DELETE MSNAME command

Short label	Long label	Keyword	Meaning
CC	CC	N/A	Completion code.
CCTXT	CCText	<i>error</i>	Completion code text that briefly explains the nonzero completion code.
ERRT	ErrorText	<i>error</i>	Error text that provides diagnostic information. Error text can be returned for a nonzero completion code and further explains the completion code.
MBR	MbrName	N/A	IMSplex member that built the output line.
MSN	MSName	N/A	The name of the logical link path.

## Return and reason codes

The following table includes the return and reason codes and a brief explanation of the codes. The return or reason code returned for the command might also indicate an error from a CSL request.

Table 220. Return and reason codes for the DELETE MSNAME command

Return code	Reason code	Meaning
X'00000000'	X'00000000'	The command completed successfully. The command output contains a line for each resource, which is accompanied by its completion code. If NAME(*) is specified without OPTION(ALLRSP), no output lines are returned. See the completion code table for details.
X'0000000C'	X'00003000'	Command was successful for some resources but failed for others. The command output contains a line for each resource, which is accompanied by its completion code. If NAME(*) is specified without OPTION(ALLRSP), only resources with nonzero completion codes are returned. See the completion code table for details.
X'0000000C'	X'00003004'	Command was not successful for any of the resources. The command output contains a line for each resource, which is accompanied by its completion code. See the completion code table for details.
X'00000010'	X'0000400C'	Command is not valid on the XRF alternate.
X'00000010'	X'00004014'	Command is not valid on the RSR tracker.
X'00000010'	X'0000402C'	Command is not valid because MSC is not enabled in the IMS system.
X'00000010'	X'0000402D'	Command is not allowed because dynamic definition is not enabled for MSC. MSCRSCS=DYN is not specified in the MSC section of the DFSDFxxx member of the IMS PROCLIB data set.
X'00000014'	X'00005004'	DFSOCMD response buffer could not be obtained.
X'00000014'	X'00005008'	DFSPPOOL storage could not be obtained.
X'00000014'	X'00005010'	Unable to obtain latch.
X'00000014'	X'00005058'	Synchronizing with DFSMNZ00 error.

## Completion codes

Errors unique to the processing of this command are returned as completion codes. The following table includes an explanation of the completion codes.

Table 221. Completion codes for the DELETE MSNAME command

Completion code	Completion code text	Meaning
0		The command completed successfully for the resource.
10	NO RESOURCES FOUND	The resource name is invalid or the value that is specified on the NAME keyword does not match the name of any resource within this resource type.

Table 221. Completion codes for the DELETE MSNAME command (continued)

Completion code	Completion code text	Meaning
17	ANOTHER CMD IN PROGRESS	Another command (such as DELETE or UPDATE) is in progress for this MSNAME. This could also mean this command, if the resource is specified by more than one specific or wildcard parameter.
48	NOT ALLOWED FOR IMS RESOURCE	An IMS system resource may not be deleted. For example, an IMS system MSNAME begins with the prefix DFSM or MSNS.
64	GETSTOR STORAGE ERROR	A DFSPOOL GET(CMDP) storage request was unsuccessful.
88	REFERENCED BY TRAN	The MSNAME cannot be deleted because it is referenced by one or more remote transactions. The transaction name is returned as error text. You can take either of these actions: <ul style="list-style-type: none"> <li>• Issue a DELETE TRAN command to delete the transaction.</li> <li>• Issue an UPDATE TRAN command to assign the transaction to another MSNAME.</li> </ul>
8D	RESOURCE IS NOT STOPPED	The resource is not stopped.
90	INTERNAL ERROR	An IMS system service failure occurred.
B7	USE MANAGER ERROR	IMS encountered an internal error using the USE manager.
109	COMMAND INVALID FOR DYNAMIC MSNAME	The DELETE command cannot complete because the specified MSNAME is a dynamic shared queues MSNAME.
10A	SOURCE MSLINK NOT STOPPED	The DELETE command cannot complete because the logical link that the specified MSNAME is assigned to is not stopped.
10E	LAST MSNAME FOR LOW SIDL	The logical link path cannot be deleted because it is the last MSNAME for the lowest local SID.
10F	SIDL STILL LOCAL IN AN RSMB	The logical link path cannot be deleted because there is a remote transaction with the SIDL value.
110	SIDR STILL REMOTE IN AN RSMB	The logical link path cannot be deleted because there is a remote transaction with the SIDR value.
145	MSG IN PROGRESS ACROSS LINK	The MSNAME cannot be deleted because there is a message in progress.

Table 221. Completion codes for the DELETE MSNAME command (continued)

Completion code	Completion code text	Meaning
20B	REFERENCED BY LTERM	<p>The MSNAME cannot be deleted because it is referenced by one or more remote LTERMs.</p> <p>The remote LTERM name is returned as error text.</p> <p>Suggested action:</p> <ul style="list-style-type: none"> <li>Issue a DELETE LTERM command to delete the remote LTERM.</li> </ul>

## Examples

The following are examples of the DELETE MSNAME command:

### Example 1 for DELETE MSNAME command

In the following example, the DELETE MSNAME command is used to delete the MSC logical link paths LINK12T3, LINK13M3, MSNM001, BADNAME, and MSNV001. The delete fails for all logical link paths because they are either not stopped or have a bad name.

TSO SPOC input:

```
DELETE MSNAME NAME(LINK13M3,MSN%001,BADNAME,LINK12T3)
```

TSO SPOC output:

```
MSName  MbrName  CC CCText
BADNAME IMS1    10 NO RESOURCES FOUND
LINK12T3 IMS1    8D RESOURCE IS NOT STOPPED
LINK13M3 IMS1    8D RESOURCE IS NOT STOPPED
MSNM001 IMS1    8D RESOURCE IS NOT STOPPED
MSNV001 IMS1    8D RESOURCE IS NOT STOPPED
```

OM API input:

```
CMD(DEL MSNAME NAME(LINK13M3,MSN%001,BADNAME,LINK12T3))
```

OM API output:

```
TBD <imsout>
<ctl>
<omname>OM10M </omname>
<omvsn>1.7.0</omvsn>
<xmlvsn>20 </xmlvsn>
<statime>2015.063 23:28:22.415541</statime>
<stotime>2015.063 23:28:22.416132</stotime>
<staseq>CE98F1761E8B5100</staseq>
<stoseq>CE98F1761EB04C00</stoseq>
<rqsttkn1>USRT011 10152822</rqsttkn1>
<rc>0200000C</rc>
<rsn>00003008</rsn>
<rsnmsg>CSLN054I</rsnmsg>
<rsntxt>None of the clients were successful.</rsntxt>
</ctl>
<cmderr>
<mbr name="IMS1 ">
<typ>IMS </typ>
```

```

<styp>DBDC </styp>
<rc>0000000C</rc>
<rsn>00003004</rsn>
<rsntxt>No requests were successful</rsntxt>
</mbr>
</cmderr>
<cmd>
<master>IMS1 </master>
<userid>USRT011 </userid>
<verb>DEL </verb>
<kwd>MSNAME </kwd>
<input>DELETE MSNAME NAME(LINK13M3,MSN%001,BADNAME,LINK12T3) </input>
</cmd>
<cmdrsphdr>
<hdr s1b1="MSN" l1b1="MSName" scope="LCL" sort="a" key="1" scroll="no "
  len="8" dtype="CHAR" align="left" />
<hdr s1b1="MBR" l1b1="MbrName" scope="LCL" sort="a" key="2" scroll="no"
  len="8" dtype="CHAR" align="left" />
<hdr s1b1="CC" l1b1="CC" scope="LCL" sort="a" key="3" scroll="yes"
  len="4" dtype="INT" align="right" skipb="no" />
<hdr s1b1="CCTXT" l1b1="CCText" scope="LCL" sort="n" key="0"
  scroll="yes" len="*" dtype="CHAR" skipb="yes" align="left" />
<hdr s1b1="ERRT" l1b1="ErrorText" scope="LCL" sort="n" key="0"
  scroll="yes" len="*" dtype="CHAR" skipb="yes" align="left" />
</cmdrsphdr>
<cmdrspdata>
<rsp>MSN(LINK13M3) MBR(IMS1 ) CC( 8D) CCTXT(RESOURCE IS NOT
  STOPPED) </rsp>
<rsp>MSN(MSNM001) MBR(IMS1 ) CC( 8D) CCTXT(RESOURCE IS NOT
  STOPPED) </rsp>
<rsp>MSN(BADNAME) MBR(IMS1 ) CC( 10) CCTXT(NO RESOURCES FOUND)
</rsp>
<rsp>MSN(LINK12T3) MBR(IMS1 ) CC( 8D) CCTXT(RESOURCE IS NOT
  STOPPED) </rsp>
<rsp>MSN(MSNV001) MBR(IMS1 ) CC( 8D) CCTXT(RESOURCE IS NOT
  STOPPED) </rsp>
</cmdrspdata>
</imsout>

```

---

## DELETE MSPLINK command

Use the DELETE MSPLINK command to delete a Multiple Systems Coupling (MSC) physical link from an online IMS system.

Subsections:

- "Environment"
- "Syntax" on page 428
- "Keywords" on page 428
- "Usage notes" on page 428
- "Output fields" on page 429
- "Return and reason codes" on page 429
- "Completion codes" on page 430
- "Examples" on page 431

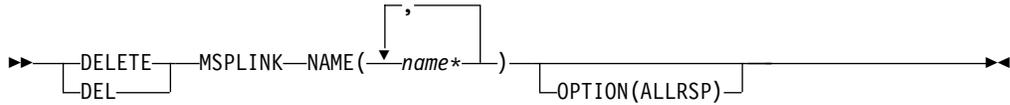
### Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

Table 222. Valid environments for the DELETE MSPLINK command and keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
DELETE MSPLINK NAME() OPTION()	X		X

## Syntax



## Keywords

The following keywords are valid for the DELETE MSPLINK command:

### NAME

Specify one or more 1 - 8 character names of MSC physical links. Wildcards are supported in the name. Examples of valid names are NAME(\*) and NAME(abc\*). If the NAME parameter specified is a specific or wildcard name, command responses are returned for all the resource names that are processed. For NAME(\*), command responses are returned only for the resource names that resulted in an error. OPTION(ALLRSP) can be specified with NAME(\*) to obtain the command responses for all the resource names that are processed.

### OPTION

Specifies additional functions to be completed along with the command.

#### ALLRSP

Indicates that the response lines are to be returned for all resources that are processed on the command. The default action is to return response lines only for the resources that result in an error. ALLRSP is valid only with NAME(\*). ALLRSP is ignored for other NAME values.

## Usage notes

Before you delete a physical link, make sure that the physical link is not in use and that any MSC logical link that are assigned to the physical link are deleted or reassigned to another physical link. If the physical link is in use or is referenced by a logical link, the delete fails.

- Issue the QUERY MSPLINK NAME(*mslinknm*) SHOW(STATUS) command to check the status of the link.
- Issue the QUERY MSPLINK NAME(*mslinknm*) SHOW(MSLINK) to see if any logical links are assigned to the link.

Changes that are made to MSC resources by the DELETE command are logged to save them across warm and emergency restarts. The changes are not saved across a cold start of IMS unless the MSC resources are also deleted from the stage-1 system definition macro input.

If your MSC resources are defined by using stage-1 system definition macros, delete the resource definition by removing the macro that defines the resource from the stage-1 system definition input and regenerate the MSC resources into the DFSCLL3x member in the IMS.SDFSRESL data set.

This command can be issued only through the OM API. This command applies to DB/DC systems and DCCTL systems.

The DELETE MSPLINK command is valid only when the dynamic definition of MSC resources is enabled (MSCRSCS=DYN is specified in the MSC section of the DFSDFxxx member). This command is recoverable.

The DELETE MSPLINK command is not valid on the XRF alternate, RSR tracker, or FDBR region.

In a Sysplex environment with multiple IMS systems, the delete might succeed on some IMS systems and fail on others.

## Output fields

The following table shows the DELETE MSPLINK output fields. The columns in the table are as follows:

### Short label

Contains the short label generated in the XML output.

### Long label

Contains the long label generated in the XML output.

### Keyword

Identifies keyword on the command that caused the field to be generated. N/A appears for output fields that are always returned. *error* appears for output fields that are returned only in case of an error.

### Meaning

Provides a brief description of the output field.

Table 223. Output fields for the DELETE MSPLINK command

Short label	Long label	Keyword	Meaning
CC	CC	N/A	The completion code.
CCTXT	CCText	<i>error</i>	The completion code text that briefly explains the nonzero completion code.
ERRT	ErrorText	<i>error</i>	Error text that provides diagnostic information. Error text can be returned for a nonzero completion code and further explains the completion code.
MBR	MbrName	N/A	IMSpIex member that built the output line.
MSCP	MSPLink	N/A	The name of the physical link.

## Return and reason codes

The following table includes the return and reason codes and a brief explanation of the codes. The return or reason code returned for the command might also indicate an error from a CSL request.

Table 224. Return and reason codes for the DELETE MSPLINK command

Return code	Reason code	Meaning
X'00000000'	X'00000000'	Command completed successfully. The command output contains a line for each resource, accompanied by its completion code. If NAME(*) is specified without OPTION(ALLRSP), no output lines are returned. See the completion code table for details.
X'0000000C'	X'00003000'	Command was successful for some resources but failed for others. The command output contains a line for each resource, accompanied by its completion code. If NAME(*) is specified without OPTION(ALLRSP), only resources with nonzero completion codes are returned. See the completion code table for details.
X'0000000C'	X'00003004'	Command was not successful for any of the resources. The command output contains a line for each resource, accompanied by its completion code. See the completion code table for details.
X'00000010'	X'0000400C'	Command is not valid on the XRF alternate.
X'00000010'	X'00004014'	Command is not valid on the RSR tracker.
X'00000010'	X'0000402C'	Command is not valid because MSC is not enabled in the IMS system.
X'00000010'	X'0000402D'	Command is not allowed because dynamic definition is not enabled for MSC. MSCRSCS=DYN is not specified in the MSC section of the DFSDFxxx member of the IMS PROCLIB data set.
X'00000014'	X'00005004'	DFSOCMD response buffer could not be obtained.
X'00000014'	X'00005008'	DFSPPOOL storage could not be obtained.
X'00000014'	X'00005010'	Unable to obtain latch.

## Completion codes

Errors unique to the processing of this command are returned as completion codes. The following table includes an explanation of the completion codes.

Table 225. Completion codes for the DELETE MSPLINK command

Completion code	Completion code text	Meaning
0		The command completed successfully for the resource.
10	NO RESOURCES FOUND	The resource name is invalid or the value that is specified on the NAME keyword does not match the name of any resource within this resource type.
17	ANOTHER CMD IN PROGRESS	Another command (such as DELETE or UPDATE) is in progress for this program. This could also mean this command, if the resource is specified by more than one specific or wildcard parameter.

Table 225. Completion codes for the DELETE MSPLINK command (continued)

Completion code	Completion code text	Meaning
8D	RESOURCE IS NOT STOPPED	The physical link cannot be deleted because the physical link is not stopped. To stop the link, issue the /PSTOP MSPLINK command or the type-2 UPDATE MSPLINK STOP(LOGON) command.
209	REFERENCED BY MSLINK	The physical link cannot be deleted because a logical link is assigned to the physical link.

## Examples

### Example 1 for DELETE MSPLINK command

In the following example, a DELETE MSPLINK command is used to delete the physical links PLNK12V, BADNAME, and PLNK12VU. The delete fails for all of the physical links for different reasons. One line is issued for each MSC logical link that references the physical link. The DELETE fails for physical link BADNAME because there is no physical link that is named BADNAME. The DELETE fails for physical links PLNK12V because it is referenced by an MSC logical link. The DELETE fails for physical link PLNK12VU because it is not stopped.

TSO SPOC input:

```
DELETE MSPLINK NAME(PLNK12V,BADNAME,PLNK12VU)
```

TSO SPOC output:

```
BADNAME IMS1      10 NO RESOURCES FOUND
PLNK12V  IMS1      8D RESOURCE IS NOT STOPPED
PLNK12V  IMS1      209 REFERENCED BY MSLINK   LNK12V01
PLNK12V  IMS1      209 REFERENCED BY MSLINK   LNK12V02
PLNK12V  IMS1      209 REFERENCED BY MSLINK   LNK12V03
PLNK12V  IMS1      209 REFERENCED BY MSLINK   DFSL0028
PLNK12VU IMS1      8D RESOURCE IS NOT STOPPED
```

OM API input:

```
CMD(DEL MSPLINK NAME(PLNK12V,BADNAME,PLNK12VU))
```

OM API output:

```
<imsout>
<ctl>
<omname>OM10M  </omname>
<omvsn>1.7.0</omvsn>
<xmlvsn>20  </xmlvsn>
<statime>2014.274 16:32:13.201064</statime>
<stotime>2014.274 16:32:13.201473</stotime>
<staseq>CDD6F538A2EA8600</staseq>
<stoseq>CDD6F538A3041D80</stoseq>
<rqsttkn1>USRT011 10093213</rqsttkn1>
<rc>0200000C</rc>
<rsn>00003008</rsn>
<rsnmsg>CSLN054I</rsnmsg>
<rsntxt>None of the clients were successful.</rsntxt>
</ctl>
<cmderr>
<mbr name="IMS1  ">
```

```

<typ>IMS      </typ>
<styp>DBDC    </styp>
<rc>0000000C</rc>
<rsn>00003004</rsn>
<rsntxt>No requests were successful</rsntxt>
</mbr>
</cmderr>
<cmd>
<master>IMS1  </master>
<userid>USRT011 </userid>
<verb>DEL </verb>
<kwd>MSPLINK </kwd>
<input>DELETE MSPLINK NAME(PLNK12V,BADNAME,PLNK12VU) </input>
</cmd>
<cmdrsphdr>
<hdr s1b1="MSP" l1b1="MSPLink" scope="LCL" sort="a" key="1" scroll="no"
  o" len="8" dtype="CHAR" align="left" />
<hdr s1b1="MBR" l1b1="MbrName" scope="LCL" sort="a" key="2" scroll="no"
  len="8" dtype="CHAR" align="left" />
<hdr s1b1="CC" l1b1="CC" scope="LCL" sort="n" key="0" scroll="yes"
  len="4" dtype="INT" align="right" skipb="no" />
<hdr s1b1="CCTXT" l1b1="CCText" scope="LCL" sort="n" key="0"
  scroll="yes" len="*" dtype="CHAR" skipb="yes" align="left" />
<hdr s1b1="ERRT" l1b1="ErrorText" scope="LCL" sort="n" key="0"
  scroll="yes" len="*" dtype="CHAR" skipb="yes" align="left" />
</cmdrsphdr>
<cmdrspdata>
<rsp>MSP(PLNK12V ) MBR(IMS1 ) CC( 8D) CCTXT(RESOURCE IS NOT
  STOPPED) </rsp>
<rsp>MSP(BADNAME ) MBR(IMS1 ) CC( 10) CCTXT(NO RESOURCES FOUND)
</rsp>
<rsp>MSP(PLNK12VU) MBR(IMS1 ) CC( 8D) CCTXT(RESOURCE IS NOT
  STOPPED) </rsp>
<rsp>MSP(PLNK12V ) MBR(IMS1 ) CC( 209) CCTXT(REFERENCED BY MSLINK)
  ERRT(LNK12V01) </rsp>
<rsp>MSP(PLNK12V ) MBR(IMS1 ) CC( 209) CCTXT(REFERENCED BY MSLINK)
  ERRT(LNK12V02) </rsp>
<rsp>MSP(PLNK12V ) MBR(IMS1 ) CC( 209) CCTXT(REFERENCED BY MSLINK)
  ERRT(LNK12V03) </rsp>
<rsp>MSP(PLNK12V ) MBR(IMS1 ) CC( 209) CCTXT(REFERENCED BY MSLINK)
  ERRT(DFSL0028) </rsp>
</cmdrspdata>
</imsout>

```

---

## DELETE OTMADESC command

Use the DELETE OTMADESC command to delete an existing OTMA destination descriptor without restarting IMS.

Subsections:

- “Environment” on page 433
- “Syntax” on page 433
- “Keywords” on page 433
- “Usage notes” on page 433
- “Output fields” on page 434
- “Return, reason, and completion codes” on page 434
- “Examples” on page 435

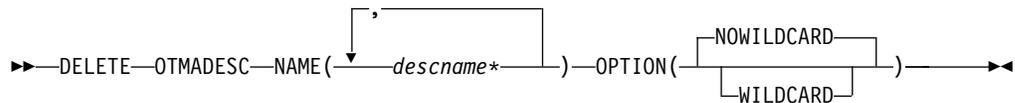
## Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the commands and keywords.

Table 226. Valid environments for the DELETE OTMADESC command and keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
DELETE OTMADESC	X		X
NAME	X		X
OPTION	X		X

## Syntax



## Keywords

The following keywords are valid for the DELETE OTMADESC command:

### NAME (*descname\**)

A required keyword parameter that specifies a 1- to 8-character destination routing descriptor name. The destination routing descriptor name can have an asterisk (\*) at the end to mask a group of names. The asterisk can be used by itself to mask all defined descriptor names in the system. Issuing the DELETE command with a name of asterisk only will delete all entries in the destination routing descriptors. If you intend to delete all entries in the table, you must specify DELETE NAME(\*) OPTION(WILDCARD). All entries including the entry of asterisk are deleted.

### OPTION(NOWILDCARD | WILDCARD)

Specify NOWILDCARD to delete the descriptor having a name with an asterisk as an entry from the table of destination routing descriptors. Specify WILDCARD to delete the groups of names that the asterisk is masking. If the name with the asterisk is also an entry in the table, the name with the asterisk itself is deleted.

## Usage notes

The DELETE OTMADESC command requires only the NAME keyword. If the NAME keyword contains the masked character of asterisk, it will not delete the group of names it is masking. It will only delete the name with the asterisk as its own entry in the destination routing descriptors. If the OPTION(WILDCARD) keyword is specified, then the group of names under a masked destination routing descriptor name will also be deleted; the name with the asterisk is also deleted. Log records will be created to track the dynamic changes to destination routing descriptors on restart of IMS.

The DELETE OTMADESC command is not valid in an XRF alternate or an RSR tracking environment.

## Output fields

The following table shows the DELETE OTMADESC output fields. The columns in the table are:

### Short label

Contains the short label generated in the XML output.

### Long label

Contains the column heading for the output field in the formatted output.

### Keyword

Identifies the keyword on the command that caused the field to be generated. N/A appears for output fields that are always returned. *error* appears for output fields that are returned only in case of an error.

### Meaning

Provides a brief description of the output field.

Table 227. Output fields for the DELETE OTMADESC command

Short label	Long label	Keyword	Meaning
CC	CC	N/A	Completion code for the line of output. The completion code indicates whether IMS was able to process the command for the specified resource. See "Return, reason, and completion codes" for more information. The completion code is always returned.
CCTXT	CCText	N/A	Completion code text that briefly explains the meaning of the non-zero completion code. This field is returned only for an error completion code.
DEST	DestName	NAME	Destination name.
MBR	MbrName	N/A	Member name.

## Return, reason, and completion codes

An IMS return and reason code is returned to OM by the DELETE OTMADESC command. The OM return and reason codes that may be returned as a result of the DELETE OTMADESC command are standard for all commands entered through the OM API.

The following table includes the return and reason codes and a brief explanation of the codes. The return or reason code returned for the command might also indicate an error from a CSL request.

Table 228. Return and reason codes for the DELETE OTMADESC command

Return code	Reason code	Meaning
X'00000000'	X'00000000'	Command completed successfully. The command output contains a line for each resource, accompanied by its completion code. See Table 229 on page 435 for details.
X'02000008'	X'00002000'	The command contains an invalid verb or no client is registered for the verb.
X'02000008'	X'00002004'	The command contains an invalid primary keyword or no client is registered for the keyword.
X'02000008'	X'00002028'	The command contains an invalid keyword.

Table 228. Return and reason codes for the DELETE OTMADESC command (continued)

Return code	Reason code	Meaning
X'02000008'	X'0000202C'	The command contains an unknown positional parameter.
X'02000008'	X'00002034'	The command contains an incomplete keyword parameter.
X'02000008'	X'00002038'	The command is missing a required parameter.
X'02000008'	X'0000203C'	The command contains an invalid keyword parameter value.

The following table includes an explanation of the completion codes. Errors unique to the processing of this command are returned as completion codes. A completion code is returned for each action against an individual resource.

Table 229. Completion codes for the DELETE OTMADESC command

Completion code	Completion code text	Meaning
0	Command completed successfully	The DELETE OTMADESC command completed successfully for the resource.
163	Descriptor not available for del	During delete processing, the entry in the table of destination routing descriptors might have already been deleted by another user.
166	No entries found	Command failed for DELETE OTMADESC, because no entries were found in the table of destination routing descriptors.

## Examples

The following are examples of the DELETE OTMADESC command:

### Example 1 for DELETE OTMADESC command

TSO SPOC input:

```
DELETE OTMADESC NAME(OTMACL*)
```

TSO SPOC output:

```
DestName  MbrName    CC
OTMACL*   IMSA          0
```

**Explanation:** The DELETE command will delete only OTMACL\* and not delete OTMACL99, because the default for the OPTION keyword is NOWILDCARD. It treats OTMACL\* as a standalone entry in the table of destination routing descriptors.

### Example 2 for DELETE OTMADESC command

TSO SPOC input:

```
DELETE OTMADESC NAME(OTMACL*) OPTION(WILDCARD)
```

TSO SPOC output:

DestName	MbrName	CC
OTMACL99	IMSA	0
OTMACL*	IMSA	0

**Explanation:** The DELETE command will delete both OTMACL99 and OTMACL\*, because the OPTION keyword is coded with WILDCARD. It treats OTMACL\* as a mask for a group of names, so it includes OTMACL99 and OTMACL\* as an entry in itself in the table of destination routing descriptors.

**Related concepts:**

➡ How to interpret CSL request return and reason codes (System Programming APIs)

**Related reference:**

Chapter 2, “Command keywords and their synonyms,” on page 63

---

## DELETE PGM command

Use the DELETE PGM command to delete program resources.

Subsections:

- “Environment”
- “Syntax”
- “Keywords”
- “Usage notes” on page 437
- “Output fields” on page 437
- “Return, reason, and completion codes” on page 438
- “Examples” on page 440

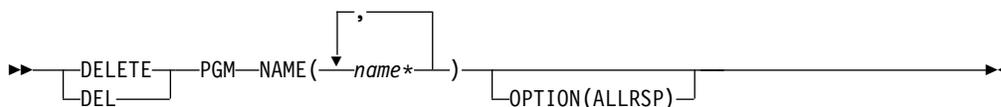
### Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

*Table 230. Valid environments for the DELETE PGM command and keywords*

Command / Keywords	DB/DC	DBCTL	DCCTL
DELETE PGM	X	X	X
NAME	X	X	X
OPTION	X	X	X

### Syntax



### Keywords

The following keywords are valid for the DELETE PGM command:

**NAME**

Specifies the 1-8 character name of the program. Wildcards are supported in the name. Examples of valid names are NAME(\*) and NAME(abc\*). If the NAME parameter specified is a specific or wildcard name, command responses are returned for all the resource names that are processed. For NAME(\*), command responses are returned only for the resource names that resulted in an error. OPTION(ALLRSP) can be specified with NAME(\*) to obtain the command responses for all the resource names that are processed.

**OPTION**

Specifies additional functions to be performed along with the command.

**ALLRSP**

Indicates that the response lines are to be returned for all resources that are processed on the command. The default action is to return response lines only for the resources that resulted in an error. It is valid only with NAME(\*). ALLRSP is ignored for other NAME values.

**Usage notes**

This command can only be issued through the Operations Manager API. This command applies to DB/DC systems, DBCTL systems, and DCCTL systems.

The DELETE PGM command is not valid on the XRF alternate, RSR tracker, or FDBR region. The DELETE PGM command is not valid if online change for MODBLKS is enabled (DFSDFxxx or DFSCGxxx defined with MODBLKS=OLC or MODBLKS not defined).

This command is recoverable.

Each program is deleted individually, unlike the online change process where either all programs are deleted or no programs are deleted. The runtime resource definition for a program can be deleted only if the program is not in use. If the program is in use, the delete fails. In a sysplex environment with multiple IMS systems, the delete might succeed on some IMS systems and fail on others.

The DELETE PGM command removes ACB members from the 31-bit nonresident pool as well as the 64-bit storage pool.

The resources remain deleted across an IMS warm start or emergency restart. To ensure that the resources remain deleted across a cold start, the existing resource definitions must be exported to a system RDDS and then imported from the RDDS at cold start. Or, if the IMSRSC repository is enabled, the DELETE DEFN command must be issued to delete the definition from the repository. If the resource definitions are to be imported from the MODBLKS data set at cold start instead of from an RDDS, the resource definitions must be deleted from system definition and a new MODBLKS data set must be generated.

The DELETE command is not allowed for IMS resources that are identified with a definition type of IMS. For DELETE PGM commands, this means Fast Path utility program DBF#FPU0.

**Output fields**

The following table shows the DELETE PGM output fields. The columns in the table are as follows:

**Short label**

Contains the short label generated in the XML output.

**Long label**

Contains the long label generated in the XML output.

**Keyword**

Identifies keyword on the command that caused the field to be generated. N/A appears for output fields that are always returned. *error* appears for output fields that are returned only in case of an error.

**Meaning**

Provides a brief description of the output field.

*Table 231. Output fields for the DELETE PGM command*

Short label	Long label	Keyword	Meaning
CC	CC	N/A	Completion code.
CCTXT	CCText	<i>error</i>	Completion code text that briefly explains the meaning of the non-zero completion code.
ERRT	ErrorText	<i>error</i>	Error text that provides diagnostic information. Error text can be returned for a non-zero completion code and further explains the completion code.
MBR	MbrName	N/A	IMSplex member that built the output line.
PGM	PgmName	PGM	Program name.

**Return, reason, and completion codes**

The following table includes the return and reason codes and a brief explanation of the codes. The return or reason code returned for the command might also indicate an error from a CSL request.

*Table 232. Return and reason codes for the DELETE PGM command*

Return code	Reason code	Meaning
X'00000000'	X'00000000'	Command completed successfully. The command output contains a line for each resource, accompanied by its completion code. If NAME(*) is specified without OPTION(ALLRSP), no output lines are returned. See the completion code table for details.
X'0000000C'	X'00003000'	Command was successful for some resources but failed for others. The command output contains a line for each resource, accompanied by its completion code. If NAME(*) is specified without OPTION(ALLRSP), only resources with non-zero completion codes are returned. See the completion code table for details.
X'0000000C'	X'00003004'	Command was not successful for any of the resources. The command output contains a line for each resource, accompanied by its completion code. See the completion code table for details.
X'00000010'	X'0000400C'	Command is not valid on the XRF alternate.
X'00000010'	X'00004014'	Command is not valid on the RSR tracker.
X'00000010'	X'00004120'	Online change phase is in progress.

Table 232. Return and reason codes for the DELETE PGM command (continued)

Return code	Reason code	Meaning
X'00000010'	X'00004300'	Command is not allowed because online change for MODBLKS is enabled (DFSDFxxx or DFSCGxxx defined with MODBLKS=OLC, or MODBLKS not defined).
X'00000014'	X'00005004'	DFSOCMD response buffer could not be obtained.
X'00000014'	X'00005008'	DFSPOOL storage could not be obtained.
X'00000014'	X'0000500C'	AWE could not be obtained.

Errors unique to the processing of this command are returned as completion codes. The following table includes an explanation of the completion codes.

Table 233. Completion codes for the DELETE PGM command

Completion code	Completion code text	Meaning	Error text
0		Command completed successfully for program.	
10	NO RESOURCES FOUND	Program name is invalid, or the wildcard parameter specified does not match any resource names.	
17	ANOTHER CMD IN PROGRESS	Another command (such as DELETE or UPDATE) is in progress for this program. This could also mean this command, if the resource is specified by more than one specific or wildcard parameter.	
48	NOT ALLOWED FOR IMS RESOURCE	DELETE command not allowed for IMS descriptor or resource. An example of an IMS resource include DBF#FPU0.	
73	PROGRAM SCHEDULED	Program is scheduled.	
7B	REFERENCED BY RTCODE	Routing code references the program to be deleted. The routing code name is returned as error text. <b>Suggested actions:</b> <ul style="list-style-type: none"> <li>Issue DELETE RTC command to delete the routing code.</li> </ul> <i>or</i> <ul style="list-style-type: none"> <li>Issue UPDATE RTC command to update the program to another name.</li> </ul>	rtcodename (8 chars)
88	REFERENCED BY TRAN	Transaction references the program to be deleted. The transaction name is returned as error text. <b>Suggested actions:</b> <ul style="list-style-type: none"> <li>Issue DELETE TRAN command to delete the transaction.</li> </ul> <i>or</i> <ul style="list-style-type: none"> <li>Issue UPDATE TRAN command to update the program to another name.</li> </ul>	trannname (8 char)

Table 233. Completion codes for the DELETE PGM command (continued)

Completion code	Completion code text	Meaning	Error text
1D7	IMPORT CHANGE LIST ERROR	Resource is in the process of being imported from the change list in the IMSRSC repository, or the import from the change list failed. The resource cannot be deleted until it is successfully imported.	

## Examples

The following are examples of the DELETE PGM command:

### Example 1 for DELETE PGM command

TSO SPOC input:

```
DELETE PGM NAME(DFSSAM02,AUTPSB2,BADNAME,DFSSAM04)
```

TSO SPOC output:

```
Response for: DELETE PGM NAME(DFSSAM02,AUTPSB2,BADNAME,DFSSAM04)
PgmName MbrName CC CText ErrorText
AUTPSB2 IMS1 0
BADNAME IMS1 10 NO RESOURCES FOUND
DFSSAM02 IMS1 88 REFERENCED BY TRAN PART
DFSSAM04 IMS1 88 REFERENCED BY TRAN ADDINV
DFSSAM04 IMS1 88 REFERENCED BY TRAN ADDPART
DFSSAM04 IMS1 88 REFERENCED BY TRAN DLETINV
DFSSAM04 IMS1 88 REFERENCED BY TRAN DLETPART
```

OM API input:

```
CMD(DEL PGM NAME(DFSSAM02,AUTPSB2,BADNAME,DFSSAM04))
```

OM API output:

```
<imsout>
<ctl>
<omname>OM10M </omname>
<omvsn>1.3.0</omvsn>
<xmlvsn>20 </xmlvsn>
<statime>2006.310 23:38:41.930125</statime>
<stotime>2006.310 23:38:41.931770</stotime>
<staseq>BFAACDEBAF38D540</staseq>
<stoseq>BFAACDEBAF9FA480</stoseq>
<rqsttkn1>USRT011 10153841</rqsttkn1>
<rc>0200000C</rc>
<rsn>00003008</rsn>
<rsnmsg>CSLN054I</rsnmsg>
<rsntxt>None of the clients were successful.</rsntxt>
</ctl>
<cmderr>
<mbr name="IMS1 ">
<typ>IMS </typ>
<styp>DBDC </styp>
<rc>0000000C</rc>
<rsn>00003000</rsn>
<rsntxt>At least one request successful</rsntxt>
</mbr>
</cmderr>
<cmd>
<master>IMS1 </master>
<userid>USRT011 </userid>
```

```

<verb>DEL </verb>
<kwd>PGM          </kwd>
<input>DELETE PGM NAME(DFSSAM02,AUTPSB2,BADNAME,DFSSAM04) </input>
</cmd>
<cmdrsphdr>
<hdr s1bl="PGM" l1bl="PgmName" scope="LCL" sort="a" key="1" scroll="no"
  len="8" dtype="CHAR" align="left" />
<hdr s1bl="MBR" l1bl="MbrName" scope="LCL" sort="a" key="2" scroll="no"
  len="8" dtype="CHAR" align="left" />
<hdr s1bl="CC" l1bl="CC" scope="LCL" sort="n" key="0" scroll="yes"
  len="4" dtype="INT" align="right" skipb="no" />
<hdr s1bl="CCTXT" l1bl="CCText" scope="LCL" sort="n" key="0"
  scroll="yes" len="*" dtype="CHAR" skipb="yes" align="left" />
<hdr s1bl="ERRT" l1bl="ErrorText" scope="LCL" sort="n" key="0"
  scroll="yes" len="*" dtype="CHAR" skipb="yes" align="left" />
</cmdrsphdr>
<cmdrspdata>
<rsp>PGM(DFSSAM02) MBR(IMS1) CC( 88) CCTXT(REFERENCED BY TRAN)
  ERRT(PART ) </rsp>
<rsp>PGM(AUTPSB2 ) MBR(IMS1) CC( 0) </rsp>
<rsp>PGM(BADNAME ) MBR(IMS1) CC( 10) CCTXT(NO RESOURCES FOUND) </rsp>
<rsp>PGM(DFSSAM04) MBR(IMS1) CC( 88) CCTXT(REFERENCED BY TRAN)
  ERRT(ADDINV ) </rsp>
<rsp>PGM(DFSSAM04) MBR(IMS1) CC( 88) CCTXT(REFERENCED BY TRAN)
  ERRT(ADDPART ) </rsp>
<rsp>PGM(DFSSAM04) MBR(IMS1) CC( 88) CCTXT(REFERENCED BY TRAN)
  ERRT(DLETINV ) </rsp>
<rsp>PGM(DFSSAM04) MBR(IMS1) CC( 88) CCTXT(REFERENCED BY TRAN)
  ERRT(DLETPART) </rsp>
</cmdrspdata>
</imsout>

```

**Explanation:** Delete programs DFSSAM02, AUTPSB2, BADNAME, and DFSSAM04. The delete is successful for program AUTPSB2, as shown by the completion code 0. The delete is rejected for DFSSAM02, because transaction PART refers to it. The delete fails for program DFSSAM04 fails, because transactions ADDPART, ADDINV, DLETPART, and DLETINV refer to it. Transactions PART, ADDPART, ADDINV, DLETPART, and DLETINV must either be deleted or updated to refer to another program before deleting programs DFSSAM02 and DFSSAM04. The delete fails for BADNAME, because there is no program named BADNAME.

**Related concepts:**

 How to interpret CSL request return and reason codes (System Programming APIs)

**Related tasks:**

 Deleting runtime application program resource and descriptor definitions with the DELETE command (System Definition)

**Related reference:**

Chapter 2, “Command keywords and their synonyms,” on page 63

## DELETE PGMDESC command

Use the DELETE PGMDESC command to delete program descriptors.

Subsections:

- “Environment” on page 442
- “Syntax” on page 442
- “Keywords” on page 442
- “Usage notes” on page 442

- “Output fields” on page 443
- “Return, reason, and completion codes” on page 443
- “Examples” on page 445

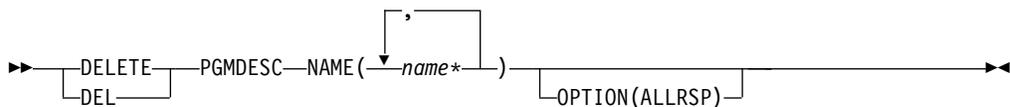
## Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

Table 234. Valid environments for the DELETE PGMDESC command and keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
DELETE PGMDESC	X	X	X
NAME	X	X	X
OPTION	X	X	X

## Syntax



## Keywords

The following keywords are valid for the DELETE PGMDESC command:

### NAME

Specifies the 1-8 character name of the program descriptor. Wildcards are supported in the name. Examples of valid names are NAME(\*) and NAME(abc\*). If the NAME parameter specified is a specific or wildcard name, command responses are returned for all the descriptor names that are processed. For NAME(\*), command responses are returned only for the descriptor names that resulted in an error. OPTION(ALLRSP) can be specified with NAME(\*) to obtain the command responses for all the descriptor names that are processed.

### OPTION

Specifies additional functions to be performed along with the command.

### ALLRSP

Indicates that the response lines are to be returned for all descriptors that are processed on the command. The default action is to return response lines only for the descriptors that resulted in an error. It is valid only with NAME(\*). ALLRSP is ignored for other NAME values.

## Usage notes

This command can only be issued through the Operations Manager API. This command applies to DB/DC systems, DBCTL systems, and DCCTL systems.

The DELETE PGMDESC command is not valid on the XRF alternate, RSR tracker, or FDBR region. The DELETE PGMDESC command is not valid if online change for MODBLKS is enabled (DFSDFxxx or DFSCGxxx defined with MODBLKS=OLC or MODBLKS not defined).

This command is recoverable.

Each descriptor is deleted individually. Individual deletion does not work like online change where either all descriptors are deleted or no descriptors are deleted. Descriptors can be successfully deleted if they are not currently in use. If a descriptor is in use, the delete fails. In a sysplex environment, the delete might succeed on some IMS systems and fail on others. A descriptor is in use if another command is in progress that references the descriptor.

The descriptors remain deleted across an IMS warm start or emergency restart. To ensure they remain deleted across a cold start, the existing descriptor definitions must be exported to an RDDS and imported from the RDDS at cold start.

The DELETE PGMDESC command is not allowed for IMS descriptors that are identified with a definition type of IMS. For DELETE PGMDESC commands, this means IMS-defined program descriptor DFSDSPG1.

## Output fields

The following table shows the DELETE PGMDESC output fields. The columns in the table are as follows:

### Short label

Contains the short label generated in the XML output.

### Long label

Contains the long label generated in the XML output.

### Keyword

Identifies keyword on the command that caused the field to be generated. N/A appears for output fields that are always returned. *error* appears for output fields that are returned only in case of an error.

### Meaning

Provides a brief description of the output field.

Table 235. Output fields for the DELETE PGMDESC command

Short label	Long label	Keyword	Meaning
CC	CC	N/A	Completion code.
CCTXT	CCText	<i>error</i>	Completion code text that briefly explains the meaning of the non-zero completion code.
DESC	DescName	PGMDESC	Program descriptor name.
ERRT	ErrorText	<i>error</i>	Error text that provides diagnostic information. Error text can be returned for a non-zero completion code and further explains the completion code.
MBR	MbrName	N/A	IMSplex member that built the output line.

## Return, reason, and completion codes

The following table includes the return and reason codes and a brief explanation of the codes. The return or reason code returned for the command might also indicate an error from a CSL request.

Table 236. Return and reason codes for the DELETE PGMDESC command

Return code	Reason code	Meaning
X'00000000'	X'00000000'	Command completed successfully. The command output contains a line for each descriptor, accompanied by its completion code. If NAME(*) is specified without OPTION(ALLRSP), no output lines are returned. See the completion code table for details.
X'0000000C'	X'00003000'	Command was successful for some descriptors but failed for others. The command output contains a line for each descriptor, accompanied by its completion code. If NAME(*) is specified without OPTION(ALLRSP), only descriptors with nonzero completion codes are returned. See the completion code table for details.
X'0000000C'	X'00003004'	Command was not successful for any of the descriptors. The command output contains a line for each descriptor, accompanied by its completion code. See the completion code table for details.
X'00000010'	X'0000400C'	Command is not valid on the XRF alternate.
X'00000010'	X'00004014'	Command is not valid on the RSR tracker.
X'00000010'	X'00004120'	Online change phase is in progress.
X'00000010'	X'00004300'	Command is not allowed because online change for MODBLKS is enabled (DFSDFxxx or DFSCGxxx defined with MODBLKS=OLC, or MODBLKS not defined).
X'00000014'	X'00005004'	DFSOCMD response buffer could not be obtained.
X'00000014'	X'00005008'	DFSPPOOL storage could not be obtained.
X'00000014'	X'0000500C'	AWE could not be obtained.

Errors unique to the processing of this command are returned as completion codes. The following table includes an explanation of the completion codes.

Table 237. Completion codes for the DELETE PGMDESC command

Completion code	Completion code text	Meaning
0		Command completed successfully for program descriptor.
10	NO RESOURCES FOUND	Program name is invalid, or the wildcard parameter specified does not match any program descriptor names.
17	ANOTHER CMD IN PROGRESS	Another command (such as DELETE or UPDATE) is in progress for this program descriptor. This could also mean this command, if the descriptor is specified by more than one specific or wildcard parameter.

Table 237. Completion codes for the DELETE PGMDESC command (continued)

Completion code	Completion code text	Meaning
27	REJECTED FOR DEFAULT DESCRIPTOR	DELETE command not allowed for user-defined default descriptor. You must issue an UPDATE command to set another descriptor as the default before deleting this descriptor.
48	NOT ALLOWED FOR IMS RESOURCE	DELETE command not allowed for IMS-defined descriptor. An example of an IMS-defined descriptor is DFSDSPG1.
1D7	IMPORT CHANGE LIST ERROR	Descriptor is in the process of being imported from the change list in the IMSRSC repository, or the import from the change list failed. The descriptor cannot be deleted until it is successfully imported.

## Examples

The following are examples of the DELETE PGMDESC command:

### Example 1 for DELETE PGMDESC command

TSO SPOC input:

```
DEL PGMDESC NAME(*) OPTION(ALLRSP)
```

TSO SPOC output:

```
Response for: DEL PGMDESC NAME(*) OPTION(ALLRSP)
DescName MbrName   CC CCText
DESC001  IMS1         0
DESC002  IMS1         0
DESC003  IMS1         0
DESC004  IMS1         0
DESC005  IMS1         0
DFSDSPG1 IMS1        48 NOT ALLOWED FOR IMS RESOURCE
DOPTDESC IMS1         0
FPEDESC  IMS1         0
GPSBDESC IMS1         0
PARLDESC IMS1         0
RESDESC  IMS1         0
TSLDESC  IMS1        27 REJECTED FOR DEFAULT DESCRIPTOR
```

OM API input:

```
CMD(DEL PGMDESC NAME(*) OPTION(ALLRSP))
```

OM API output:

```
<imsout>
<ctl>
<omname>OM10M  </omname>
<omvsn>1.3.0</omvsn>
<xmlvsn>20  </xmlvsn>
<statime>2006.310 23:09:24.532552</statime>
<stotime>2006.310 23:09:24.533243</stotime>
<staseq>BFAAC75FB314869E</staseq>
<stoseq>BFAAC75FB33FB7DE</stoseq>
<rqsttkn1>USRT011 10150924</rqsttkn1>
```

```

<rc>0200000C</rc>
<rsn>00003008</rsn>
<rsnmsg>CSLN054I</rsnmsg>
<rsntxt>None of the clients were successful.</rsntxt>
</ctl>
<cmderr>
<mbr name="IMS1  ">
<typ>IMS      </typ>
<styp>DBDC    </styp>
<rc>0000000C</rc>
<rsn>00003000</rsn>
<rsntxt>At least one request successful</rsntxt>
</mbr>
</cmderr>
<cmd>
<master>IMS1  </master>
<userid>USRT011 </userid>
<verb>DEL </verb>
<kwd>PGMDESC      </kwd>
<input>DEL PGMDESC NAME(*) OPTION(ALLRSP) </input>
</cmd>
<cmdrsphdr>
<hdr s1bl="DESC" l1bl="DescName" scope="LCL" sort="a" key="1"
  scroll="no" len="8" dtype="CHAR" align="left" />
<hdr s1bl="MBR" l1bl="MbrName" scope="LCL" sort="a" key="2" scroll="no"
  len="8" dtype="CHAR" align="left" />
<hdr s1bl="CC" l1bl="CC" scope="LCL" sort="n" key="0" scroll="yes"
  len="4" dtype="INT" align="right" skipb="no" />
<hdr s1bl="CCTXT" l1bl="CCText" scope="LCL" sort="n" key="0"
  scroll="yes" len="*" dtype="CHAR" skipb="yes" align="left" />
<hdr s1bl="ERRT" l1bl="ErrorText" scope="LCL" sort="n" key="0"
  scroll="yes" len="*" dtype="CHAR" skipb="yes" align="left" />
</cmdrsphdr>
<cmdrspdata>
<rsp>DESC(DESC004 ) MBR(IMS1) CC( 0) </rsp>
<rsp>DESC(DESC005 ) MBR(IMS1) CC( 0) </rsp>
<rsp>DESC(DESC001 ) MBR(IMS1) CC( 0) </rsp>
<rsp>DESC(DOPTDESC) MBR(IMS1) CC( 0) </rsp>
<rsp>DESC(RESDESC ) MBR(IMS1) CC( 0) </rsp>
<rsp>DESC(DFSDSPG1) MBR(IMS1) CC( 48) CCTXT(NOT ALLOWED FOR IMS
  RESOURCE) </rsp>
<rsp>DESC(TLSDESC ) MBR(IMS1) CC( 27) CCTXT(REJECTED FOR DEFAULT
  DESCRIPTOR) </rsp>
<rsp>DESC(DESC002 ) MBR(IMS1) CC( 0) </rsp>
<rsp>DESC(PARLDESC) MBR(IMS1) CC( 0) </rsp>
<rsp>DESC(DESC003 ) MBR(IMS1) CC( 0) </rsp>
<rsp>DESC(FPEDESC ) MBR(IMS1) CC( 0) </rsp>
<rsp>DESC(GPSBDESC) MBR(IMS1) CC( 0) </rsp>
</cmdrspdata>
</imsout>

```

**Explanation:** A DELETE PGMDESC command is issued to delete all program descriptors from an active IMS system. Many program descriptors are successfully deleted, as indicated by the completion code 0. Because the DELETE command attempted to delete all of the program descriptors, OPTION(ALLRSP) is specified to display the successfully deleted program descriptors. Program descriptor TLSDESC fails, because it is the default program descriptor. Program descriptor DFSDSPG1 fails, because it is the IMS-defined program descriptor.

**Related concepts:**

 [How to interpret CSL request return and reason codes \(System Programming APIs\)](#)

**Related reference:**

Chapter 2, "Command keywords and their synonyms," on page 63

---

## DELETE RTC command

Use the DELETE RTC command to delete Fast Path routing codes.

Subsections:

- “Environment”
- “Syntax”
- “Keywords”
- “Usage notes” on page 448
- “Output fields” on page 448
- “Return, reason, and completion codes” on page 449
- “Examples” on page 450

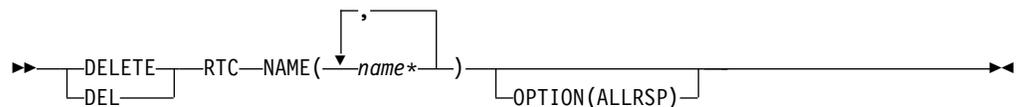
### Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

Table 238. Valid environments for the DELETE RTC command and keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
DELETE RTC	X		X
NAME	X		X
OPTION	X		X

### Syntax



### Keywords

The following keywords are valid for the DELETE RTC command:

#### NAME

Specifies the 1-8 character name of the routing code. Wildcards are supported in the name. Examples of valid names are NAME(\*) and NAME(abc\*). If the NAME parameter specified is a specific or wildcard name, command responses are returned for all the resource names that are processed. For NAME(\*), command responses are returned only for the resource names that resulted in an error. OPTION(ALLRSP) can be specified with NAME(\*) to obtain the command responses for all the resource names that are processed.

#### OPTION

Specifies additional functions to be performed along with the command.

#### ALLRSP

Indicates that the response lines are to be returned for all resources that are processed on the command. The default action is to return response lines only for the resources that resulted in an error. It is valid only with NAME(\*). ALLRSP is ignored for other NAME values.

## Usage notes

This command can only be issued through the Operations Manager API. This command applies to DB/DC systems and DCCTL systems. The DELETE RTC command applies if Fast Path is installed.

The DELETE RTC command is not valid on the XRF alternate, RSR tracker, or FDBR region. The DELETE RTC command is not valid if online change for MODBLKS is enabled (DFSDFxxx or DFSCGxxx defined with MODBLKS=OLC or MODBLKS not defined).

This command is recoverable.

Each routing code is deleted individually, unlike the online change process where either all routing codes are deleted or no routing codes are deleted. The runtime resource definition for a routing code can only be deleted if the routing code is not in use. If the routing code is in use, the delete fails. In a sysplex environment with multiple IMSs, the delete might succeed on some IMSs and fail on others.

The resources remain deleted across an IMS warm start or emergency restart. To ensure that the resources remain deleted across a cold start, the existing resource definitions must be exported to a system RDDS and then imported from the RDDS at cold start. Or, if the IMSRSC repository is enabled, the DELETE DEFN command must be issued to delete the definition from the repository. If the resource definitions are to be imported from the MODBLKS data set at cold start instead of from an RDDS, the resource definitions must be deleted from system definition and a new MODBLKS data set must be generated.

## Output fields

The following table shows the DELETE RTC output fields. The columns in the table are as follows:

### Short label

Contains the short label generated in the XML output.

### Long label

Contains the long label generated in the XML output.

### Keyword

Identifies keyword on the command that caused the field to be generated. N/A appears for output fields that are always returned. *error* appears for output fields that are returned only in case of an error.

### Meaning

Provides a brief description of the output field.

Table 239. Output fields for the DELETE RTC command

Short label	Long label	Keyword	Meaning
CC	CC	N/A	Completion code.
CCTXT	CCText	<i>error</i>	Completion code text that briefly explains the meaning of the non-zero completion code.
ERRT	ErrorText	<i>error</i>	Error text that provides diagnostic information. Error text can be returned for a non-zero completion code and further explains the completion code.

Table 239. Output fields for the DELETE RTC command (continued)

Short label	Long label	Keyword	Meaning
MBR	MbrName	N/A	IMSpIex member that built the output line.
RTC	RtcName	RTC	Routing code name.

## Return, reason, and completion codes

The following table includes the return and reason codes and a brief explanation of the codes. The return or reason code returned for the command might also indicate an error from a CSL request.

Table 240. Return and reason codes for the DELETE RTC command

Return code	Reason code	Meaning
X'00000000'	X'00000000'	Command completed successfully. The command output contains a line for each resource, accompanied by its completion code. If NAME(*) is specified without OPTION(ALLRSP), no output lines are returned. See the completion code table for details.
X'0000000C'	X'00003000'	Command was successful for some resources but failed for others. The command output contains a line for each resource, accompanied by its completion code. If NAME(*) is specified without OPTION(ALLRSP), only resources with non-zero completion codes are returned. See the completion code table for details.
X'0000000C'	X'00003004'	Command was not successful for any of the resources. The command output contains a line for each resource, accompanied by its completion code. See the completion code table for details.
X'00000010'	X'0000400C'	Command is not valid on the XRF alternate.
X'00000010'	X'00004014'	Command is not valid on the RSR tracker.
X'00000010'	X'00004024'	No Fast Path defined.
X'00000010'	X'00004120'	Online change phase is in progress.
X'00000010'	X'00004300'	Command is not allowed because online change for MODBLKS is enabled (DFSDFxxx or DFSCGxxx defined with MODBLKS=OLC, or MODBLKS not defined).
X'00000014'	X'00005004'	DFSOCMD response buffer could not be obtained.
X'00000014'	X'00005008'	DFSPOOL storage could not be obtained.
X'00000014'	X'0000500C'	AWE could not be obtained.

Errors unique to the processing of this command are returned as completion codes. The following table includes an explanation of the completion codes.

Table 241. Completion codes for the DELETE RTC command

Completion code	Completion code text	Meaning
0		Command completed successfully for routing code.

Table 241. Completion codes for the DELETE RTC command (continued)

Completion code	Completion code text	Meaning
10	NO RESOURCES FOUND	Routing code name is invalid, or the wildcard parameter specified does not match any routing code names.
17	ANOTHER CMD IN PROGRESS	Another command (such as DELETE or UPDATE) is in progress for this routing code. This could also mean this command, if the resource is specified by more than one specific or wildcard parameter.
39	FP=E TRAN FOR RTC EXISTS	Routing code was created by IMS for a Fast Path exclusive transaction and cannot be deleted by a DELETE RTC command. The routing code can only be deleted by a DELETE TRAN command for the Fast Path exclusive transaction.
73	PROGRAM SCHEDULED	Program is scheduled.
1D7	IMPORT CHANGE LIST ERROR	Resource is in the process of being imported from the change list in the IMSRSC repository, or the import from the change list failed. The resource cannot be deleted until it is successfully imported.

## Examples

The following are examples of the DELETE RTC command:

### Example 1 for DELETE RTC command

TSO SPOC input:

```
DEL RTC NAME(OLCFR102,GFP1,BADNAME,EMHTX32,EMHTX32)
```

TSO SPOC output:

```
Response for: DEL RTC NAME(OLCFR102,GFP1,BADNAME,EMHTX32,EMHTX32)
RtcName MbrName CC CText
BADNAME IMS1 10 NO RESOURCES FOUND
EMHTX32 IMS1 0
EMHTX32 IMS1 17 ANOTHER CMD IN PROGRESS
GFP1 IMS1 39 FP=E TRAN FOR RTC EXISTS
OLCFR102 IMS1 0
```

OM API input:

```
CMD(DEL RTC NAME(OLCFR102,GFP1,BADNAME,EMHTX32,EMHTX32))
```

OM API output:

```
<imsout>
<ctl>
<omname>OM10M </omname>
<omvsn>1.3.0</omvsn>
<xmlvsn>20 </xmlvsn>
<statime>2006.310 23:55:52.034690</statime>
<stotime>2006.310 23:55:52.035318</stotime>
```

```

<stoseq>BFAAD1C2119824D2</stoseq>
<stoseq>BFAAD1C211BF6E52</stoseq>
<rqsttkn1>USRT011 10155552</rqsttkn1>
<rc>0200000C</rc>
<rsn>00003008</rsn>
<rsnmsg>CSLN054I</rsnmsg>
<rsntxt>None of the clients were successful.</rsntxt>
</ctl>
<cmderr>
<mbr name="IMS1  ">
<typ>IMS  </typ>
<styp>DBDC  </styp>
<rc>0000000C</rc>
<rsn>00003000</rsn>
<rsntxt>At least one request successful</rsntxt>
</mbr>
</cmderr>
<cmd>
<master>IMS1  </master>
<userid>USRT011 </userid>
<verb>DEL </verb>
<kwd>RTC  </kwd>
<input>DEL RTC NAME(OLCFR102,GFP1,BADNAME,EMHTX32,EMHTX32) </input>
</cmd>
<cmdrsphdr>
<hdr s1bl="RTC" l1bl="RtcName" scope="LCL" sort="a" key="1" scroll="no"
len="8" dtype="CHAR" align="left" />
<hdr s1bl="MBR" l1bl="MbrName" scope="LCL" sort="a" key="2" scroll="no"
len="8" dtype="CHAR" align="left" />
<hdr s1bl="CC" l1bl="CC" scope="LCL" sort="n" key="0" scroll="yes"
len="4" dtype="INT" align="right" />
<hdr s1bl="CCTXT" l1bl="CCText" scope="LCL" sort="n" key="0"
scroll="yes" len="32" dtype="CHAR" align="left" skipb="yes" />
</cmdrsphdr>
<cmdrspdata>
<rsp>RTC(OLCFR102) MBR(IMS1) CC( 0) </rsp>
<rsp>RTC(GFP1 ) MBR(IMS1) CC( 39) CCTXT(FP=E TRAN FOR RTC EXISTS)
</rsp>
<rsp>RTC(BADNAME ) MBR(IMS1) CC( 10) CCTXT(NO RESOURCES FOUND) </rsp>
<rsp>RTC(EMHTX32 ) MBR(IMS1) CC( 0) </rsp>
<rsp>RTC(EMHTX32 ) MBR(IMS1) CC( 17) CCTXT(ANOTHER CMD IN PROGRESS)
</rsp>
</cmdrspdata>
</imsout>

```

**Explanation:** A DELETE RTC command is issued to delete several Fast Path routing codes from a running IMS system. The delete succeeds for some routing codes, as shown by completion code 0. The delete fails for routing code BADNAME, because no such routing code exists. The delete fails for routing code GFP1, because it was created for a Fast Path exclusive transaction called GFP1 and can only be deleted by a DELETE TRAN NAME(GFP1) command. Routing code EMHTX32 is successfully deleted, but routing code EMHTX32 is specified again, which fails because the DELETE RTC command is already in progress for EMHTX32.

**Related concepts:**

 [How to interpret CSL request return and reason codes \(System Programming APIs\)](#)

**Related tasks:**

 [Deleting runtime Fast Path routing code resource and descriptor definitions with the DELETE command \(System Definition\)](#)

**Related reference:**



lines only for the descriptors that resulted in an error. It is valid only with NAME(\*). ALLRSP is ignored for other NAME values.

## Usage notes

This command can only be issued through the Operations Manager API. This command applies to DB/DC systems and DCCTL systems. The DELETE RTCDESC command applies if Fast Path is installed.

The DELETE RTCDESC command is not valid on the XRF alternate, RSR tracker, or FDBR region. The DELETE RTCDESC command is not valid if online change for MODBLKS is enabled (DFSDFxxx or DFSCGxxx defined with MODBLKS=OLC or MODBLKS not defined).

This command is recoverable.

Each descriptor is deleted individually. Individual deletion does not work like online change where either all descriptors are deleted or no descriptors are deleted. Descriptors can be successfully deleted if they are not currently in use. If a descriptor is in use, the delete fails. In a sysplex environment, the delete might succeed on some IMS systems and fail on others. A descriptor is in use if another command is in progress that references the descriptor.

The descriptors remain deleted across an IMS warm start or emergency restart. To ensure they remain deleted across a cold start, the existing descriptor definitions must be exported to an RDDS and then imported from the RDDS at cold start.

The IMS-defined routing code descriptor (DBFDSRT1) cannot be deleted with the DELETE RTCDESC command.

## Output fields

The following table shows the DELETE RTCDESC output fields. The columns in the table are as follows:

### Short label

Contains the short label generated in the XML output.

### Long label

Contains the long label generated in the XML output.

### Keyword

Identifies keyword on the command that caused the field to be generated. N/A appears for output fields that are always returned. *error* appears for output fields that are returned only in case of an error.

### Meaning

Provides a brief description of the output field.

Table 243. Output fields for the DELETE RTCDESC command

Short label	Long label	Keyword	Meaning
CC	CC	N/A	Completion code.
CCTXT	CCText	<i>error</i>	Completion code text that briefly explains the meaning of the non-zero completion code.
DESC	DescName	RTCDESC	Routing code descriptor name.

Table 243. Output fields for the DELETE RTCDESC command (continued)

Short label	Long label	Keyword	Meaning
ERRT	ErrorText	<i>error</i>	Error text that provides diagnostic information. Error text can be returned for a non-zero completion code and further explains the completion code.
MBR	MbrName	N/A	IMSpIex member that built the output line.

## Return, reason, and completion codes

The following table includes the return and reason codes and a brief explanation of the codes. The return or reason code returned for the command might also indicate an error from a CSL request.

Table 244. Return and reason codes for the DELETE RTCDESC command

Return code	Reason code	Meaning
X'00000000'	X'00000000'	Command completed successfully. The command output contains a line for each descriptor, accompanied by its completion code. If NAME(*) is specified without OPTION(ALLRSP), no output lines are returned. See the completion code table for details.
X'0000000C'	X'00003000'	Command was successful for some descriptors but failed for others. The command output contains a line for each descriptor, accompanied by its completion code. If NAME(*) is specified without OPTION(ALLRSP), only descriptors with non-zero completion codes are returned. See the completion code table for details.
X'0000000C'	X'00003004'	Command was not successful for any of the descriptors. The command output contains a line for each descriptor, accompanied by its completion code. See the completion code table for details.
X'00000010'	X'0000400C'	Command is not valid on the XRF alternate.
X'00000010'	X'00004014'	Command is not valid on the RSR tracker.
X'00000010'	X'00004024'	No Fast Path defined.
X'00000010'	X'00004120'	Online change phase is in progress.
X'00000010'	X'00004300'	Command is not allowed because online change for MODBLKS is enabled (DFSDFxxx or DFSCGxxx defined with MODBLKS=OLC, or MODBLKS not defined).
X'00000014'	X'00005004'	DFSOCMD response buffer could not be obtained.
X'00000014'	X'00005008'	DFSPPOOL storage could not be obtained.
X'00000014'	X'0000500C'	AWE could not be obtained.

Errors unique to the processing of this command are returned as completion codes. The following table includes an explanation of the completion codes.

Table 245. Completion codes for the DELETE RTCDESC command

Completion code	Completion code text	Meaning
0		Command completed successfully for routing code descriptor.
10	NO RESOURCES FOUND	Routing code descriptor name is invalid, or the wildcard parameter specified does not match any routing code descriptor names.
17	ANOTHER CMD IN PROGRESS	Another command (such as DELETE or UPDATE) is in progress for this routing code descriptor. This could also mean this command, if the descriptor is specified by more than one specific or wildcard parameter.
27	REJECTED FOR DEFAULT DESCRIPTOR	DELETE command not allowed for user-defined default descriptor. You must issue an UPDATE command to set another descriptor as the default before deleting this descriptor.
48	NOT ALLOWED FOR IMS RESOURCE	DELETE command not allowed for IMS descriptor. An example of an IMS descriptor is DBFDSRT1.
1D7	IMPORT CHANGE LIST ERROR	Descriptor is in the process of being imported from the change list in the IMSRSC repository, or the import from the change list failed. The descriptor cannot be deleted until it is successfully imported.

## Examples

The following are examples of the DELETE RTCDESC command:

### Example 1 for DELETE RTCDESC command

TSO SPOC input:

```
DEL RTCDESC NAME(*) OPTION(ALLRSP)
```

TSO SPOC output:

```
Response for: DEL RTCDESC NAME(*) OPTION(ALLRSP)
DescName MbrName  CC CText
DBFDSRT1 IMS1     48 NOT ALLOWED FOR IMS RESOURCE
DESC001  IMS1         0
DESC002  IMS1         0
DESC003  IMS1         0
DESC004  IMS1         0
DESC005  IMS1         0
RTCDESC1 IMS1     27 REJECTED FOR DEFAULT DESCRIPTOR
```

OM API input:

```
CMD(DEL RTCDESC NAME(*) OPTION(ALLRSP))
```

OM API output:

```

<imsout>
<ctl>
<omname>OM10M </omname>
<omvsn>1.3.0</omvsn>
<xmlvsn>20 </xmlvsn>
<statime>2006.310 22:57:13.782312</statime>
<stotime>2006.310 22:57:13.783073</stotime>
<staseq>BFAAC4A6CD428992</staseq>
<stoseq>BFAAC4A6CD721DC0</stoseq>
<rqsttkn1>USRT011 10145713</rqsttkn1>
<rc>0200000C</rc>
<rsn>00003008</rsn>
<rsnmsg>CSLN054I</rsnmsg>
<rsntxt>None of the clients were successful.</rsntxt>
</ctl>
<cmderr>
<mbr name="IMS1 ">
<typ>IMS </typ>
<styp>DBDC </styp>
<rc>0000000C</rc>
<rsn>00003000</rsn>
<rsntxt>At least one request successful</rsntxt>
</mbr>
</cmderr>
<cmd>
<master>IMS1 </master>
<userid>USRT011 </userid>
<verb>DEL </verb>
<kwd>RTCDESC </kwd>
<input>DEL RTCDESC NAME(*) OPTION(ALLRSP) </input>
</cmd>
<cmdrsphdr>
<hdr slbl="DESC" llbl="DescName" scope="LCL" sort="a" key="1"
scroll="no" len="8" dtype="CHAR" align="left" />
<hdr slbl="MBR" llbl="MbrName" scope="LCL" sort="a" key="2" scroll="no"
len="8" dtype="CHAR" align="left" />
<hdr slbl="CC" llbl="CC" scope="LCL" sort="n" key="0" scroll="yes"
len="4" dtype="INT" align="right" />
<hdr slbl="CCTXT" llbl="CCText" scope="LCL" sort="n" key="0"
scroll="yes" len="32" dtype="CHAR" align="left" skip="yes" />
</cmdrsphdr>
<cmdrspdata>
<rsp>DESC(DESC004 ) MBR(IMS1) CC( 0) </rsp>
<rsp>DESC(DESC005 ) MBR(IMS1) CC( 0) </rsp>
<rsp>DESC(DESC001 ) MBR(IMS1) CC( 0) </rsp>
<rsp>DESC(RTCDESC1) MBR(IMS1) CC( 27) CCTXT(REJECTED FOR DEFAULT
DESCRIPTOR) </rsp>
<rsp>DESC(DBFDSRT1) MBR(IMS1) CC( 48) CCTXT(NOT ALLOWED FOR IMS
RESOURCE) </rsp>
<rsp>DESC(DESC002 ) MBR(IMS1) CC( 0) </rsp>
<rsp>DESC(DESC003 ) MBR(IMS1) CC( 0) </rsp>
</cmdrspdata>
</imsout>

```

**Explanation:** A DELETE RTCDESC command is issued to delete all routing code descriptors from an active IMS system. Many routing code descriptors are successfully deleted, as indicated by the completion code 0. Because the DELETE command attempted to delete all of the routing code descriptors, OPTION(ALLRSP) is specified to display the successfully deleted routing code descriptors. Routing code descriptor RTCDESC1 fails, because it is the default routing code descriptor. Routing code descriptor DBFDSRT1 fails, because it is the IMS-defined routing code descriptor.

**Related concepts:**

➡ How to interpret CSL request return and reason codes (System Programming APIs)

**Related reference:**

Chapter 2, “Command keywords and their synonyms,” on page 63

---

## DELETE TRAN command

Use the DELETE TRAN command to delete transactions.

Subsections:

- “Environment”
- “Syntax”
- “Keywords”
- “Usage notes” on page 458
- “Output fields” on page 458
- “Return, reason, and completion codes” on page 459
- “Examples” on page 461

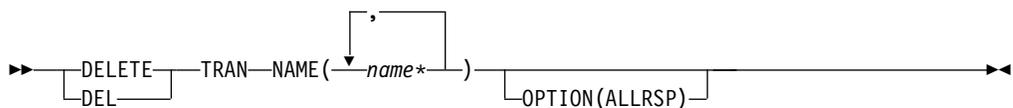
### Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

Table 246. Valid environments for the DELETE TRAN command and keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
DELETE TRAN	X		X
NAME	X		X
OPTION	X		X

### Syntax



### Keywords

The following keywords are valid for the DELETE TRAN command:

#### NAME

Specifies the 1-8 character name of the transaction. Wildcards are supported in the name. Examples of valid names are NAME(\*) and NAME(abc\*). If the NAME parameter specified is a specific or wildcard name, command responses are returned for all the resource names that are processed. For NAME(\*), command responses are returned only for the resource names that resulted in an error. OPTION(ALLRSP) can be specified with NAME(\*) to obtain the command responses for all the resource names that are processed.

#### OPTION

Specifies additional functions to be performed along with the command.

## **ALLRSP**

Indicates that the response lines are to be returned for all resources that are processed on the command. The default action is to return response lines only for the resources that resulted in an error. It is valid only with NAME(\*). ALLRSP is ignored for other NAME values.

## **Usage notes**

This command can only be issued through the OM API. This command applies to DB/DC systems and DCCTL systems.

The DELETE TRAN command is not valid on the XRF alternate, RSR tracker, or FDBR region. The DELETE TRAN command is not valid if online change for MODBLKS is enabled (DFSDFxxx or DFSCGxxx defined with MODBLKS=OLC or MODBLKS not defined).

This command is recoverable.

Each transaction is deleted individually, unlike the online change process where either all transactions are deleted or no transactions are deleted. The runtime resource definition for a transaction can only be deleted if the transaction is not in use. If the transaction is in use, the delete fails. In a sysplex environment with multiple IMS systems, the delete might succeed on some IMS systems and fail on others.

The resources remain deleted across an IMS warm start or emergency restart. To ensure that the resources remain deleted across a cold start, the existing resource definitions must be exported to a system RDDS and then imported from the RDDS at cold start. Or, if the IMSRSC repository is enabled, the DELETE DEFN command must be issued to delete the definition from the repository. If the resource definitions are to be imported from the MODBLKS data set at cold start instead of from an RDDS, the resource definitions must be deleted from system definition and a new MODBLKS data set must be generated.

The DELETE TRAN command stops all queuing and scheduling of the transaction.

If the transaction is a Fast Path exclusive transaction, the routing code by the same name is deleted as well.

## **Output fields**

The following table shows the DELETE TRAN output fields. The columns in the table are as follows:

### **Short label**

Contains the short label generated in the XML output.

### **Long label**

Contains the long label generated in the XML output.

### **Keyword**

Identifies keyword on the command that caused the field to be generated. N/A appears for output fields that are always returned. *error* appears for output fields that are returned only in case of an error.

### **Meaning**

Provides a brief description of the output field.

Table 247. Output fields for the DELETE TRAN command

Short label	Long label	Keyword	Meaning
CC	CC	N/A	Completion code.
CCTXT	CCText	<i>error</i>	Completion code text that briefly explains the nonzero completion code.
CONVID	ConvID	TRAN	Conversation id of conversation associated with transaction that caused the delete to fail with a completion code of C'1A'. This information may be used to exit the conversation, before attempting the delete again.
ERRT	ErrorText	<i>error</i>	Error text that provides diagnostic information. Error text can be returned for a nonzero completion code and further explains the completion code.
MBR	MbrName	N/A	IMSpIex member that built the output line.
LU	LUName	TRAN	APPC LU name associated with the transaction conversation that caused the delete to fail with a completion code of C'1A'. This information may be used to exit the conversation, before attempting the delete again.
NODE	NodeName	TRAN	Node name of static node associated with transaction conversation that caused the delete to fail with a completion code of C'1A'. This information may be used to exit the conversation, before attempting the delete again.
TMEM	TMember	TRAN	OTMA tmember name associated with transaction conversation that caused the delete to fail with a completion code of C'1A'. This information may be used to exit the conversation, before attempting the delete again.
TPIP	TPipe	TRAN	OTMA tpIpe name associated with the transaction conversation that caused the delete to fail with a completion code of C'1A'. This information may be used to exit the conversation, before attempting the delete again.
TRAN	Trancode	TRAN	Transaction name.
USER	UserName	TRAN	User name of dynamic user associated with the transaction conversation that caused the delete to fail with a completion code of C'1A'. This information may be used to exit the conversation, before attempting the delete again.

## Return, reason, and completion codes

The following table includes the return and reason codes and a brief explanation of the codes. The return or reason code returned for the command might also indicate an error from a CSL request.

Table 248. Return and reason codes for the DELETE TRAN command

Return code	Reason code	Meaning
X'00000000'	X'00000000'	Command completed successfully. The command output contains a line for each resource, accompanied by its completion code. If NAME(*) is specified without OPTION(ALLRSP), no output lines are returned. See the completion code table for details.
X'0000000C'	X'00003000'	Command was successful for some resources but failed for others. The command output contains a line for each resource, accompanied by its completion code. If NAME(*) is specified without OPTION(ALLRSP), only resources with nonzero completion codes are returned. See the completion code table for details.
X'0000000C'	X'00003004'	Command was not successful for any of the resources. The command output contains a line for each resource, accompanied by its completion code. See the completion code table for details.
X'00000010'	X'0000400C'	Command is not valid on the XRF alternate.
X'00000010'	X'00004014'	Command is not valid on the RSR tracker.
X'00000010'	X'00004120'	Online change phase is in progress.
X'00000010'	X'00004300'	Command is not allowed because online change for MODBLKS is enabled (DFSDFxxx or DFSCGxxx defined with MODBLKS=OLC, or MODBLKS not defined).
X'00000014'	X'00005004'	DFSOCMD response buffer could not be obtained.
X'00000014'	X'00005008'	DFSPOOL storage could not be obtained.
X'00000014'	X'0000500C'	AWE could not be obtained.

Errors unique to the processing of this command are returned as completion codes. The following table includes an explanation of the completion codes.

Table 249. Completion codes for the DELETE TRAN command

Completion code	Completion code text	Meaning
0		Command completed successfully for transaction.
10	NO RESOURCES FOUND	Transaction name is invalid, or the wildcard parameter specified does not match any transaction names.
17	ANOTHER CMD IN PROGRESS	Another command (such as DELETE or UPDATE) is in progress for this transaction or routing code associated with this transaction, if it is Fast Path exclusive. This could also mean this command, if the resource is specified by more than one specific or wildcard parameter.

Table 249. Completion codes for the DELETE TRAN command (continued)

Completion code	Completion code text	Meaning
1A	IN CONVERSATION	Transaction is in conversation.  The conversation ID and terminal in conversation are returned separately. The terminal can be a static node, node and user, dynamic user, APPC luname, or OTMA tmember and tpipe.  Suggested actions: Terminate the conversation.
4A	IN USE	Transaction is in use. Queuing is in progress, either terminal input or a program-to-program switch.
73	PROGRAM SCHEDULED	Program is scheduled.
85	SUSPENDED	Transaction is on the suspend queue.
87	TRAN QUEUEING	Transaction has messages queued (non-shared-queues environment).
89	TRAN SCHEDULED	Transaction is scheduled.
B3	ELIGIBLE FOR SCHEDULING	Transaction is eligible for scheduling and cannot be deleted. You may need to stop the transaction with an UPDATE TRAN STOP(Q,SCHD) command before attempting the DELETE again.
1D7	IMPORT CHANGE LIST ERROR	Resource is in the process of being imported from the change list in the IMSRSC repository, or the import from the change list failed. The resource cannot be deleted until it is successfully imported.

## Examples

The following are examples of the DELETE TRAN command:

### Example 1 for DELETE TRAN command

TSO SPOC input:

```
DEL TRAN NAME(AOP,BADNAME,APOL*,BAD*,AOP)
```

TSO SPOC output:

```
Response for: DEL TRAN NAME(AOP,BADNAME,APOL*,BAD*,AOP)
Trancode MbrName  CC CCText
AOP      IMS1      0
AOP      IMS1      17 ANOTHER CMD IN PROGRESS
APOL11   IMS1      0
APOL12   IMS1      0
APOL13   IMS1      0
APOL14   IMS1      0
APOL15   IMS1      0
APOL16   IMS1      0
```

```

APOL17 IMS1 0
APOL18 IMS1 0
BAD* IMS1 10 NO RESOURCES FOUND
BADNAME IMS1 10 NO RESOURCES FOUND

```

OM API input:

```
CMD(DEL TRAN NAME(AOP,BADNAME,APOL*,BAD*,AOP))
```

OM API output:

```

<imsout>
<ctl>
<omname>OM10M </omname>
<omvsn>1.3.0</omvsn>
<xmlvsn>20 </xmlvsn>
<statime>2006.311 00:04:00.354581</statime>
<stotime>2006.311 00:04:00.354544</stotime>
<staseq>BFAAD393C4515645</staseq>
<stoseq>BFAAD393C487EF05</stoseq>
<rqsttkn1>USRT011 10160400</rqsttkn1>
<rc>0200000C</rc>
<rsn>00003008</rsn>
<rsnmsg>CSLN054I</rsnmsg>
<rsntxt>None of the clients were successful.</rsntxt>
</ctl>
<cmderr>
<mbr name="IMS1 ">
<typ>IMS </typ>
<styp>DBDC </styp>
<rc>0000000C</rc>
<rsn>00003004</rsn>
<rsntxt>No requests were successful</rsntxt>
</mbr>
</cmderr>
<cmd>
<master>IMS1 </master>
<userid>USRT011 </userid>
<verb>DEL </verb>
<kwd>TRAN </kwd>
<input>DEL TRAN NAME(AOP,BADNAME,APOL*,BAD*,AOP) </input>
</cmd>
<cmdrsphdr>
<hdr s1b1="TRAN" l1b1="Trancode" scope="LCL" sort="a" key="1"
  scroll="no" len="8" dtype="CHAR" align="left" />
<hdr s1b1="MBR" l1b1="MbrName" scope="LCL" sort="a" key="2" scroll="no"
  len="8" dtype="CHAR" align="left" />
<hdr s1b1="CC" l1b1="CC" scope="LCL" sort="n" key="0" scroll="yes"
  len="4" dtype="INT" align="right" skipb="no" />
<hdr s1b1="CCTXT" l1b1="CCText" scope="LCL" sort="n" key="0"
  scroll="yes" len="*" dtype="CHAR" skipb="yes" align="left" />
<hdr s1b1="ERRT" l1b1="ErrorText" scope="LCL" sort="n" key="0"
  scroll="yes" len="*" dtype="CHAR" skipb="yes" align="left" />
<hdr s1b1="CONVID" l1b1="ConvID" scope="LCL" sort="n" key="0"
  scroll="yes" len="4" dtype="CHAR" skipb="yes" align="left" />
<hdr s1b1="NODE" l1b1="NodeName" scope="LCL" sort="n" key="0"
  scroll="yes" len="8" dtype="CHAR" skipb="yes" align="left" />
<hdr s1b1="USER" l1b1="UserName" scope="LCL" sort="n" key="0"
  scroll="yes" len="8" dtype="CHAR" skipb="yes" align="left" />
<hdr s1b1="LU" l1b1="LUName" scope="LCL" sort="n" key="0"
  scroll="yes" len="24" dtype="CHAR" skipb="yes" align="left" />
<hdr s1b1="TMEM" l1b1="TMember" scope="LCL" sort="n" key="0"
  scroll="yes" len="16" dtype="CHAR" skipb="yes" align="left" />
<hdr s1b1="TPIP" l1b1="TPipe" scope="LCL" sort="n" key="0" scroll="yes"
  len="8" dtype="CHAR" skipb="yes" align="left" />
</cmdrsphdr>
<cmdrspdata>
<rsp>TRAN(AOP ) MBR(IMS1) CC( 0) </rsp>

```

```

| <rsp>TRAN(BADNAME ) MBR(IMS1) CC( 10) CCTXT(NO RESOURCES FOUND) </rsp>
| <rsp>TRAN(APOL11 ) MBR(IMS1) CC( 0) </rsp>
| <rsp>TRAN(BAD* ) MBR(IMS1) CC( 10) CCTXT(NO RESOURCES FOUND) </rsp>
| <rsp>TRAN(AOP ) MBR(IMS1) CC( 17) CCTXT(ANOTHER CMD IN PROGRESS)
| </rsp>
| <rsp>TRAN(APOL12 ) MBR(IMS1) CC( 0) </rsp>
| <rsp>TRAN(APOL13 ) MBR(IMS1) CC( 0) </rsp>
| <rsp>TRAN(APOL14 ) MBR(IMS1) CC( 0) </rsp>
| <rsp>TRAN(APOL15 ) MBR(IMS1) CC( 0) </rsp>
| <rsp>TRAN(APOL16 ) MBR(IMS1) CC( 0) </rsp>
| <rsp>TRAN(APOL17 ) MBR(IMS1) CC( 0) </rsp>
| <rsp>TRAN(APOL18 ) MBR(IMS1) CC( 0) </rsp>
| </cmdrspdata>
| </imsout>

```

**Explanation:** A DELETE TRAN command is issued to delete several transactions from a running IMS system. The delete succeeds for some transactions, as shown by completion code 0. The delete fails for transaction BADNAME, because no such transaction exists. The delete fails for transaction name BAD\*, because no transactions start with BAD. Transaction AOP is successfully deleted, but transaction AOP is specified again, which fails because the DELETE TRAN command is already in progress for transaction AOP.

**Related concepts:**

 How to interpret CSL request return and reason codes (System Programming APIs)

**Related tasks:**

 Deleting runtime transaction resource and descriptor definitions with the DELETE command (System Definition)

**Related reference:**

Chapter 2, “Command keywords and their synonyms,” on page 63

## DELETE TRANDESC command

Use the DELETE TRANDESC command to delete transaction descriptors.

**Subsections:**

- “Environment”
- “Syntax” on page 464
- “Keywords” on page 464
- “Usage notes” on page 464
- “Output fields” on page 465
- “Return, reason, and completion codes” on page 465
- “Examples” on page 467

### Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

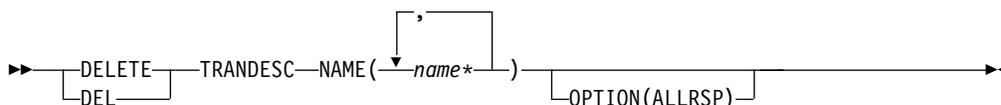
*Table 250. Valid environments for the DELETE TRANDESC command and keywords*

Command / Keywords	DB/DC	DBCTL	DCCTL
DELETE TRANDESC	X		X
NAME	X		X

Table 250. Valid environments for the DELETE TRANDESC command and keywords (continued)

Command / Keywords	DB/DC	DBCTL	DCCTL
OPTION	X		X

## Syntax



## Keywords

The following keywords are valid for the DELETE TRANDESC command:

### NAME

Specifies the 1-8 character name of the transaction descriptor. Wildcards are supported in the name. Examples of valid names are NAME(\*) and NAME(abc\*). If the NAME parameter specified is a specific or wildcard name, command responses are returned for all the descriptor names that are processed. For NAME(\*), command responses are returned only for the descriptor names that resulted in an error. OPTION(ALLRSP) can be specified with NAME(\*) to obtain the command responses for all the descriptor names that are processed.

### OPTION

Specifies additional functions to be performed along with the command.

### ALLRSP

Indicates that the response lines are to be returned for all descriptors that are processed on the command. The default action is to return response lines only for the descriptors that resulted in an error. It is valid only with NAME(\*). ALLRSP is ignored for other NAME values.

## Usage notes

This command can only be issued through the Operations Manager API. This command applies to DB/DC systems and DCCTL systems.

The DELETE TRANDESC command is not valid on the XRF alternate, RSR tracker, or FDBR region. The DELETE TRANDESC command is not valid if online change for MODBLKS is enabled (DFSDFxxx or DFSCGxxx defined with MODBLKS=OLC or MODBLKS not defined). This command is recoverable.

Each descriptor is deleted individually. Individual deletion does not work like online change where either all descriptors are deleted or no descriptors are deleted. Descriptors can be successfully deleted if they are not currently in use. If a descriptor is in use, the delete fails. In a sysplex environment, the delete might succeed on some IMS systems and fail on others. A descriptor is in use if another command is in progress that references the command.

The descriptors remain deleted across an IMS warm start or emergency restart. To ensure they remain deleted across a cold start, the existing descriptor definitions must be exported to an RDDS and then imported from the RDDS at cold start.

The DELETE TRANDESC command is not allowed for IMS descriptors that are identified with a definition type of IMS. For DELETE TRANDESC commands, this means IMS transaction descriptor DFSDSTR1.

## Output fields

The following table shows the DELETE TRANDESC output fields. The columns in the table are as follows:

### Short label

Contains the short label generated in the XML output.

### Long label

Contains the long label generated in the XML output.

### Keyword

Identifies keyword on the command that caused the field to be generated. N/A appears for output fields that are always returned. *error* appears for output fields that are returned only in case of an error.

### Meaning

Provides a brief description of the output field.

Table 251. Output fields for the DELETE TRANDESC command

Short label	Long label	Keyword	Meaning
CC	CC	N/A	Completion code.
CCTXT	CCText	<i>error</i>	Completion code text that briefly explains the nonzero completion code.
DESC	DescName	TRANDESC	Transaction descriptor name.
ERRT	ErrorText	<i>error</i>	Error text that provides diagnostic information. Error text can be returned for a nonzero completion code and further explains the completion code.
MBR	MbrName	N/A	IMSplex member that built the output line.

## Return, reason, and completion codes

The following table includes the return and reason codes and a brief explanation of the codes. The return or reason code returned for the command might also indicate an error from a CSL request.

Table 252. Return and reason codes for the DELETE TRANDESC command

Return code	Reason code	Meaning
X'00000000'	X'00000000'	Command completed successfully. The command output contains a line for each descriptor, accompanied by its completion code. If NAME(*) is specified without OPTION(ALLRSP), no output lines are returned. See the completion code table for details.
X'0000000C'	X'00003000'	Command was successful for some descriptors but failed for others. The command output contains a line for each descriptor, accompanied by its completion code. If NAME(*) is specified without OPTION(ALLRSP), only descriptors with nonzero completion codes are returned. See the completion code table for details.

Table 252. Return and reason codes for the DELETE TRANDESC command (continued)

Return code	Reason code	Meaning
X'0000000C'	X'00003004'	Command was not successful for any of the descriptors. The command output contains a line for each descriptor, accompanied by its completion code. See the completion code table for details.
X'00000010'	X'0000400C'	Command is not valid on the XRF alternate.
X'00000010'	X'00004014'	Command is not valid on the RSR tracker.
X'00000010'	X'00004120'	Online change phase is in progress.
X'00000010'	X'00004300'	Command is not allowed because online change for MODBLKS is enabled (DFSDFxxx or DFSCGxxx defined with MODBLKS=OLC, or MODBLKS not defined).
X'00000014'	X'00005004'	DFSOCMD response buffer could not be obtained.
X'00000014'	X'00005008'	DFSPOOL storage could not be obtained.
X'00000014'	X'0000500C'	AWE could not be obtained.

Errors unique to the processing of this command are returned as completion codes. The following table includes an explanation of the completion codes.

Table 253. Completion codes for the DELETE TRANDESC command

Completion code	Completion code text	Meaning
0		Command completed successfully for transaction descriptor.
10	NO RESOURCES FOUND	Transaction descriptor name is invalid, or the wildcard parameter specified does not match any transaction descriptor names.
17	ANOTHER CMD IN PROGRESS	Another command (such as DELETE or UPDATE) is in progress for this transaction descriptor. This could also mean this command, if the descriptor is specified by more than one specific or wildcard parameter.
27	REJECTED FOR DEFAULT DESCRIPTOR	DELETE command not allowed for user-defined default descriptor. You must issue an UPDATE command to set another descriptor as the default before deleting this descriptor.
48	NOT ALLOWED FOR IMS RESOURCE	DELETE command not allowed for IMS descriptor. An example of an IMS descriptor is DFSWSTR1.
1D7	IMPORT CHANGE LIST ERROR	Descriptor is in the process of being imported from the change list in the IMSRSC repository, or the import from the change list failed. The descriptor cannot be deleted until it is successfully imported.

## Examples

The following are examples of the DELETE TRANDESC command:

### *Example 1 for DELETE TRANDESC command*

TSO SPOC input:

```
DEL TRANDESC NAME(*) OPTION(ALLRSP)
```

TSO SPOC output:

```
Response for: DEL TRANDESC NAME(*) OPTION(ALLRSP)
DescName MbrName   CC CText
CONVDESC IMS1      0
DESC001  IMS1      0
DESC002  IMS1      0
DESC003  IMS1      0
DESC004  IMS1      0
DESC005  IMS1      0
DESC101  IMS1      0
DESC102  IMS1      0
DESC103  IMS1      0
DESC104  IMS1      0
DESC105  IMS1      0
DESC201  IMS1      0
DESC202  IMS1      0
DESC203  IMS1      0
DESC204  IMS1      0
DESC205  IMS1      0
DFSDSTR1 IMS1      48 NOT ALLOWED FOR IMS RESOURCE
FPEDESC  IMS1      27 REJECTED FOR DEFAULT DESCRIPTOR
FPPDESC  IMS1      0
MSCDESC  IMS1      0
```

OM API input:

```
CMD(DEL TRANDESC NAME(*) OPTION(ALLRSP))
```

OM API output:

```
<imsout>
<ctl>
<omname>OM10M </omname>
<omvsn>1.3.0</omvsn>
<xmlvsn>20 </xmlvsn>
<statime>2006.310 22:43:49.648480</statime>
<stotime>2006.310 22:43:49.650611</stotime>
<staseq>BFAAC1A7EB860302</staseq>
<stoseq>BFAAC1A7EC0B37D2</stoseq>
<rqsttkn1>USRT011 10144349</rqsttkn1>
<rc>02000000C</rc>
<rsn>00003008</rsn>
<rsnmsg>CSLN054I</rsnmsg>
<rsntxt>None of the clients were successful.</rsntxt>
</ctl>
<cmderr>
<mbr name="IMS1 ">
<typ>IMS </typ>
<styp>DBDC </styp>
<rc>00000000C</rc>
<rsn>00003004</rsn>
<rsntxt>No requests were successful</rsntxt>
</mbr>
</cmderr>
<cmd>
<master>IMS1 </master>
```

```

<userid>USRT011 </userid>
<verb>DEL </verb>
<kwd>TRANDESC </kwd>
<input>DEL TRANDESC NAME(*) OPTION(ALLRSP) </input>
</cmd>
<cmdrsphdr>
<hdr s1bl="DESC" l1bl="DescName" scope="LCL" sort="a" key="1"
  scroll="no" len="8" dtype="CHAR" align="left" />
<hdr s1bl="MBR" l1bl="MbrName" scope="LCL" sort="a" key="2" scroll="no"
  len="8" dtype="CHAR" align="left" />
<hdr s1bl="CC" l1bl="CC" scope="LCL" sort="n" key="0" scroll="yes"
  len="4" dtype="INT" align="right" skipb="no" />
<hdr s1bl="CCTXT" l1bl="CCText" scope="LCL" sort="n" key="0"
  scroll="yes" len="*" dtype="CHAR" skipb="yes" align="left" />
<hdr s1bl="ERRT" l1bl="ErrorText" scope="LCL" sort="n" key="0"
  scroll="yes" len="*" dtype="CHAR" skipb="yes" align="left" />
<hdr s1bl="CONVID" l1bl="ConvID" scope="LCL" sort="n" key="0"
  scroll="yes" len="4" dtype="CHAR" skipb="yes" align="left" />
<hdr s1bl="NODE" l1bl="NodeName" scope="LCL" sort="n" key="0"
  scroll="yes" len="8" dtype="CHAR" skipb="yes" align="left" />
<hdr s1bl="USER" l1bl="UserName" scope="LCL" sort="n" key="0"
  scroll="yes" len="8" dtype="CHAR" skipb="yes" align="left" />
<hdr s1bl="LU" l1bl="LUName" scope="LCL" sort="n" key="0"
  scroll="yes" len="24" dtype="CHAR" skipb="yes" align="left" />
<hdr s1bl="TMEM" l1bl="TMember" scope="LCL" sort="n" key="0"
  scroll="yes" len="16" dtype="CHAR" skipb="yes" align="left" />
<hdr s1bl="TPIP" l1bl="TPipe" scope="LCL" sort="n" key="0" scroll="yes"
  len="8" dtype="CHAR" skipb="yes" align="left" />
</cmdrsphdr>
<cmdrspdata>
<rsp>DESC(DESC102 ) MBR(IMS1) CC( 0) </rsp>
<rsp>DESC(DESC004 ) MBR(IMS1) CC( 0) </rsp>
<rsp>DESC(DESC205 ) MBR(IMS1) CC( 0) </rsp>
<rsp>DESC(DFSDSTR1) MBR(IMS1) CC( 48) CCTXT(NOT ALLOWED FOR IMS
  RESOURCE) </rsp>
<rsp>DESC(DESC201 ) MBR(IMS1) CC( 0) </rsp>
<rsp>DESC(FPPDESC ) MBR(IMS1) CC( 0) </rsp>
<rsp>DESC(DESC103 ) MBR(IMS1) CC( 0) </rsp>
<rsp>DESC(DESC005 ) MBR(IMS1) CC( 0) </rsp>
<rsp>DESC(DESC001 ) MBR(IMS1) CC( 0) </rsp>
<rsp>DESC(DESC202 ) MBR(IMS1) CC( 0) </rsp>
<rsp>DESC(DESC104 ) MBR(IMS1) CC( 0) </rsp>
<rsp>DESC(CONVDESC) MBR(IMS1) CC( 0) </rsp>
<rsp>DESC(DESC002 ) MBR(IMS1) CC( 0) </rsp>
<rsp>DESC(DESC203 ) MBR(IMS1) CC( 0) </rsp>
<rsp>DESC(MSCDESC ) MBR(IMS1) CC( 0) </rsp>
<rsp>DESC(DESC105 ) MBR(IMS1) CC( 0) </rsp>
<rsp>DESC(DESC101 ) MBR(IMS1) CC( 0) </rsp>
<rsp>DESC(FPEDESC ) MBR(IMS1) CC( 27) CCTXT(REJECTED FOR DEFAULT
  DESCRIPTOR) </rsp>
<rsp>DESC(DESC003 ) MBR(IMS1) CC( 0) </rsp>
<rsp>DESC(DESC204 ) MBR(IMS1) CC( 0) </rsp>
</cmdrspdata>
</imsout>

```

**Explanation:** A DELETE TRANDESC command is issued to delete all transaction descriptors from an active IMS system. Many transaction descriptors are successfully deleted, as indicated by the completion code 0. Because the DELETE command attempted to delete all of the transaction descriptors, OPTION(ALLRSP) is specified to display the successfully deleted transaction descriptors. The delete for transaction descriptor FPEDESC fails, because it is the default transaction descriptor. The delete for transaction descriptor DFSDSTR1 fails, because it is the IMS-defined transaction descriptor.

**Related concepts:**

 How to interpret CSL request return and reason codes (System Programming APIs)

**Related reference:**

Chapter 2, “Command keywords and their synonyms,” on page 63



---

## Chapter 24. /DEQUEUE command

The /DEQUEUE command with any of the following keywords is used to cancel a message that is being processed: LINE, LTERM, LUNAME, MSNAME, NODE, TRAN, or USER.

Also, a /DEQUEUE command dequeues messages from either the local queues, or in a shared-queues environment, the shared queues. An output message displays the total count of all messages dequeued.

Subsections:

- “Environment”
- “Syntax”
- “Keywords” on page 472
- “Usage notes” on page 476
- “Examples” on page 477

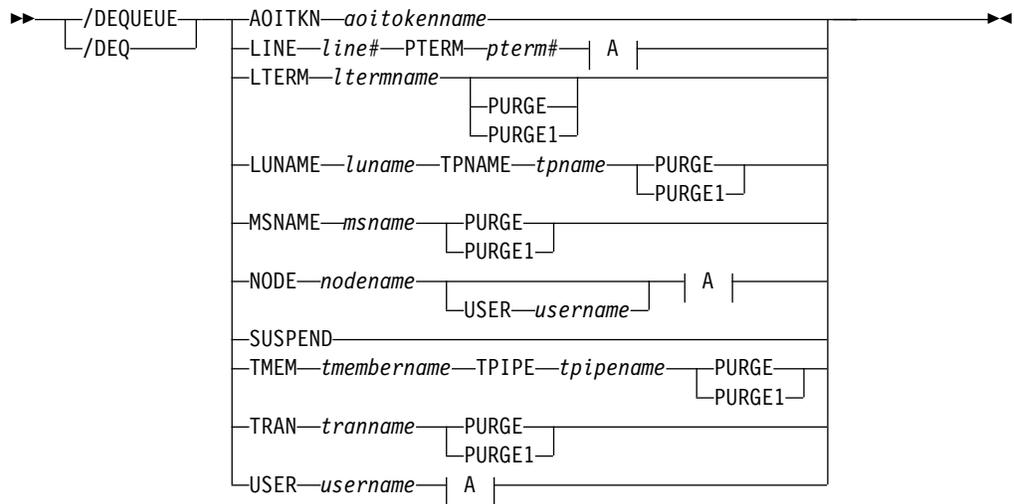
### Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

*Table 254. Valid environments for the /DEQUEUE command and keywords*

Command / Keywords	DB/DC	DBCTL	DCCTL
/DEQUEUE	X	X	X
AOITKN	X	X	X
LINE	X		X
LTERM	X		X
LUNAME	X		X
MSNAME	X		X
NODE	X		X
PTERM	X		X
PURGE	X		X
PURGE1	X		X
SUSPEND	X		X
TMEM	X		X
TPIPE	X		X
TPNAME	X		X
TRAN	X		X
USER	X		X

### Syntax



**A:**



## Keywords

The following keywords are valid for the /DEQUEUE command:

### AOITOKEN

Dequeues and discards all messages associated with the specified AOI token name.

### LINE PTERM

Cancels the output message currently being sent to the specified physical terminal on the specified communication line.

### LTERM

Cancels the output message currently being sent to the specified logical terminal.

/DEQUEUE NODE nodename LTERM is not valid for nodes with users. The user that contains the LTERM must be allocated and the session must be active.

If the PURGE or PURGE1 keyword is specified and the LTERM is associated with a user with DEADQ status, the DEADQ status will be reset.

If the DEADQ status is on, the following command turns it off for the user associated with LTERM ltermname:

```
/DEQUEUE LTERM ltermname (PURGE/PURGE1)
```

The DEADQ status is removed by signing the user on or by issuing the /DEQUEUE command to dequeue one or more messages.

| **Important:** Both the USER and the LTERM blocks must exist in either the IMS  
 | system or the Resource Manager (RM) resource structure. Otherwise, a DFS189  
 | error message might occur when the /DEQUEUE LTERM ltermname

(PURGE/PURGE1) command is issued for a dynamic LTERM in a shared-queue environment with STM=YES specified. To solve this problem, see the instructions in DFS189 (Messages and Codes).

When the /DEQUEUE LTERM ltermname (PURGE/PURGE1) command is successfully completed, issue a /START USER command to reset the user status so that the user structure can be deleted at the next checkpoint.

#### **LUNAME TPNAME**

Specifies the LU name and TP name of the LU 6.2 destination from which messages are discarded. A normal (non-qualified) or fully network qualified LUNAME is required for the LUNAME keyword.

**Note:** A fully qualified LU name must be specified in single quotation marks.

Either the PURGE or PURGE1 option must be specified, and the LU 6.2 destination must be stopped and not active; if both actions are not taken, error messages are issued.

In a shared-queues environment, only messages that are owned by the command processing IMS are dequeued. Inflight synchronous output messages are not dequeued. Any stranded synchronous output messages with affinity to the command processing IMS are dequeued.

If you used a side information entry name to place messages on the queues, to dequeue those messages, specify the side information entry name for the LU name and the character string DFSSIDE for the TP name.

#### **MSNAME**

Cancels the output message currently being sent to the specified MSC link. Either PURGE or PURGE1 is required for this link. The MSC link must be stopped by the /PSTOP LINK command before issuing the /DEQUEUE MSNAME command.

#### **NODE**

Cancels the output message currently being sent to the specified VTAM node.

The following command is not valid for nodes with users:

```
/DEQUEUE NODE nodename  
/DEQUEUE NODE nodename LTERM ltermname
```

The following command cancels the output message currently being sent to the specified ISC session, and the user must be allocated to the node and the session must be active:

```
/DEQUEUE NODE nodename USER username
```

#### **Restrictions for using NODE and USER parameters together:**

- Commands with the NODE USER keyword pair are valid only if:
  - The USER is signed on to the NODE
  - In an ISC environment, the USER is allocated to the NODE
  - The nodes and users already exist
- /DEQUEUE NODE USER LTERM commands are valid for ISC and non-ISC nodes and users.
- /DEQUEUE NODE USER PURGE commands and /DEQUEUE NODE USER LTERM PURGE1 commands are valid for ISC, LUP, and 3600 nodes only.

If PURGE or PURGE1 is specified, the node must be stopped and idle. Without PURGE and PURGE1, if an ETO node or an ISC static parallel session, the USER keyword is required.

If the PURGE keyword is specified, and the NODE is associated with a user with DEADQ status, the DEADQ status will be reset.

#### **PURGE**

Cancels all output messages enqueued to the specified LINE, LTERM, LUNAME, MSNAME, NODE, TMEM, TPIPE, or USER. The PURGE keyword allows cancellation of all output messages enqueued to an LTERM, or to a PTERM without regard to the LTERM. A /DEQUEUE command without PURGE cancels an in-progress output message on the specified terminal and, optionally, LTERM. If the NODE and LTERM keywords have both been supplied, the LTERM must be a valid LTERM of the user currently associated with the node.

Messages cannot be canceled with PURGE if they are destined for the MTO or for terminals or users in response mode or conversation mode. The /DEQUEUE PURGE command does not cancel messages destined for the master terminal operator because the master terminal cannot be stopped, disconnected, or idled. The /EXIT command should be used for a terminal in conversation mode.

If the terminal is a dynamic 3600/FINANCE, SLU P, or ISC terminal, the NODE keyword is valid only if the session is stopped and idle, and the LTERMs are still allocated to the terminal pending message resynchronization. For example, the following command is valid for the allocated user only if the specified ISC session is stopped and idled.

```
/DEQUEUE NODE nodename USER username (LTERM ltermname) PURGE
```

Messages queued to LTERMs associated with users that are not signed on can be purged by specifying the LTERM keyword without the NODE keyword. NODE with the USER keyword applies to ISC sessions and dynamic terminals and is valid only if the user is still allocated to the node. However, the line or node must be stopped and idled.

If USER is specified and the user had DEADQ status, the DEADQ status is removed. If LTERM or NODE is specified and the LTERM or node is associated with a user with DEADQ status, the DEADQ status is removed.

#### **PURGE1**

Cancels the first message queued to the specified LTERM. The PURGE1 and PURGE keywords are mutually exclusive.

The /DEQUEUE LTERM PURGE1 command removes the first message from the queue. The order of search for messages to be removed is:

1. Fast Path
2. Response mode
3. Exclusive mode messages
4. System messages
5. All other messages

Using a synonym for PURGE1 avoids the potential danger of omitting the trailing digit on PURGE1 and canceling all the messages enqueued. Synonyms for PURGE1 are SINGLE, SNGL, and FIRST.

When PURGE1 is specified, the terminal must be stopped and idle and not in conversation mode. If the message is in the process of being actively sent, The /DEQUEUE command without PURGE1 cancels the message.

### **SUSPEND**

Reschedules transactions that have been suspended by IMS. SUSPEND applies to IMS systems sharing data at the block level and to connected external subsystems (not CCTL subsystems). The /DEQUEUE SUSPEND command causes all message-driven transactions that have been suspended to be rescheduled. When a failed IMS system is restarted, the /DEQUEUE SUSPEND command informs the currently running IMS system to reschedule any message-driven transactions on its SUSPEND queue. However, when the failed subsystem has global locks retained and cannot communicate the release of those locks to the other subsystems, the /DEQUEUE SUSPEND command is used to reschedule these message-driven transactions in the other subsystems. This can occur when:

- The still-running IMS has transactions on its SUSPEND queue and the failed IMS has been restarted using the /ERESTART COLDSYS command.
- The still-running IMS terminates before the failed IMS is restarted. Thus, when the failed system is restarted, the other system, which has now terminated, is unable to receive the message to reschedule the transactions on its SUSPEND queue. The operator must issue the /DEQUEUE SUSPEND command to reschedule them.

Another use for the /DEQUEUE SUSPEND command is to release those transactions queued because the installation specified the requeue region error option in the external subsystem PROCLIB member for a particular subsystem. Transactions will be queued if the subsystem connection terminates in an abnormal fashion (for example, dependent region abend, subsystem abnormal termination).

In a shared-queues environment, a /DEQUEUE SUSPEND command moves all transactions on the Transaction Suspend queue to the Transaction Ready queue and makes them eligible for rescheduling. The “suspend” status for the transactions is also reset locally (on the IMS subsystem that entered the command), but is not reset across the sysplex. To reset the “suspend” status across a sysplex, use the /START TRAN command on each IMS subsystem.

You can also use the /START TRAN command to reschedule a specific transaction that has been suspended.

The /DISPLAY TRAN and /DISPLAY STATUS commands can be used to determine whether a transaction has messages on its suspend queue.

### **TMEM TPIPE**

Dequeues messages from the specified IMS Open Transaction Manager Access (OTMA) member and transaction pipe. Using the /DEQ command can make OTMA message resynchronization impossible. If resynchronization is in progress for the specified transaction pipe, IMS rejects the command and issues message DFS2392I.

In a shared-queues environment, only messages that are owned by the command processing IMS are dequeued. Inflight synchronous output messages are not dequeued. Any stranded synchronous output messages with affinity to the command processing IMS are dequeued.

## TRAN

Dequeues one or more messages from the transaction specified. The transaction name is a 1- to 8-byte name. The transaction must be stopped for messages to be dequeued.

**Restriction:** This keyword is valid only in a shared-queues environment.

**Recommendation:** Stop the transaction on all IMS subsystems in the shared queues group before issuing the /DEQUEUE TRAN command. If you do not, the IMS subsystem that processes the /DEQUEUE could dequeue a message it does not own and that another IMS subsystem might need.

A /DEQUEUE TRAN command does not dequeue messages enqueued for suspended transactions.

In a shared-queues environment, a /DEQUEUE TRAN command can dequeue transaction messages queued to the transaction from APPC or OTMA Clients. The command only dequeues the transaction messages. The APPC or OTMA transaction instance blocks (TIB) created at the IMS system are not deleted and the storage may remain allocated until the next IMS restart. For APPC, the TIB is not released and the APPC conversation (the client) hangs if there is no timeout value specified. The APPCIOT=(x,Y) timeout value for APPC would deallocate the APPC conversation and release the TIB after the timeout limit has been reached.

## USER (LTERM)

Dequeues all messages associated with all LTERMs (or a specific) LTERM assigned to that USER. The user must be stopped and not in conversation in order to enter /DEQUEUE USER with PURGE and PURGE1. The /DEQUEUE USER command with PURGE1 or the /DEQUEUE USER LTERM command without PURGE or PURGE1 requires the LTERM keyword. If a user is signed on, the /DEQUEUE USER LTERM command can be used to cancel an output message that is currently being processed. When the purge options are used, messages destined for the MTO and users in response mode, Fast Path, and conversation cannot be canceled. If the USER and LTERM keywords have both been supplied, the LTERM must be a valid LTERM of the specified USER.

If the user has DEADQ status, the DEADQ status is removed.

## Usage notes

The DEQUEUE command dequeues messages from an OTMA transaction pipe. CM0 output messages can be placed on the OTMA tpipe using multiple output queue names. A different queue name is used to queue I/O PCB output messages than is used to queue hold queue messages. In the non-super member environment, a CM0 output message could be on the I/O PCB output queue or the member's tpipe hold queue. In the super member environment, a CM0 output message could be on I/O PCB output queue or the super member hold queue.

If the member specified is a regular member, CM0 output messages are dequeued from the member's tpipe hold queue and the member's I/O PCB output queue. If the member specified is a super member, CM0 output messages are dequeued from the super member's output hold queue. The number of messages dequeued depends on whether PURGE or PURGE1 is specified. In the non-shared-queues environment, if the member specified is a regular member, messages are dequeued from the regular member's hold queue first, then from the regular member's I/O PCB output queue. In the shared-queues environment the order in which messages for the same destination are dequeued is not guaranteed. Hold queue messages

may be dequeued before I/O PCB output messages one time and I/O PCB output messages may be dequeued before hold queue messages the next.

If IMS cold starts and messages are queued to the same tpipe destination following the cold start, a new set of queue names is used to queue I/O PCB output messages and hold queue messages. In the shared-queues environment with super member enabled, all messages queued to a super member, are placed on the shared queues using the same queue name, even after a cold start of IMS. Messages on the super member's output queue do not have affinity to any IMS. They can be dequeued from any IMS in the IMSplex.

This command can be issued to an IMSplex using the Batch SPOC utility.

A /DEQUEUE command with any of the following keywords activates the Message Control Error exit routine, DFSCMUX0, before processing each message: LTERM, LUNAME TPNAME, NODE, or MSNAME. The user exit routine can request IMS to:

- Proceed with the command and purge the message. This is the default action for the Message Control Error exit routine.
- Suppress the command.
- Suppress the command and send an informational message to the entering terminal.
- Route the message to a different destination.

The /DEQUEUE NODE command dequeues messages from the local or shared queues for statically defined terminals. Without PURGE or PURGE1, the message actively being sent is dequeued.

The /DEQUEUE LTERM command dequeues messages from the local or shared queues. The user must be stopped (by a /STOP USER command), and not in conversation, in order to enter /DEQUEUE USER or /DEQUEUE LTERM with PURGE or PURGE1. In a shared queues and ETO environment where the user does not exist, /DEQUEUE LTERM with PURGE or PURGE1 creates a user structure in order to dequeue messages from the shared queues.

## Examples

The following are examples of the /DEQUEUE command:

### *Example 1 for /DEQUEUE command*

Entry ET:

```
/DEQUEUE AOITOKEN ABCDE
```

Response ET:

```
DFS058I DEQUEUE COMMAND COMPLETED
```

Explanation: The messages on AOITOKEN ABCDE are dequeued and discarded.

### *Example 2 for /DEQUEUE command*

Entry ET:

```
/DEQUEUE LINE 5 PTERM 2
```

Response ET:

```
DFS058I DEQUEUE COMMAND COMPLETED
```

Explanation: The message is terminated on successful completion of the current transmission.

*Example 3 for /DEQUEUE command*

Entry ET:

```
/DEQUEUE LTERM ABCDE PURGE
```

Response ET:

```
DFS976I 2 MESSAGE(S) PURGED
```

Explanation: Two messages are dequeued for LTERM ABCDE (the line is stopped and idle as required).

*Example 4 for /DEQUEUE command*

Entry ET:

```
/DEQUEUE LTERM ABCDE PURGE1
```

Response ET:

```
DFS058I DEQUEUE COMMAND COMPLETED
```

Explanation: The message is dequeued for LTERM ABCDE.

*Example 5 for /DEQUEUE command*

Entry ET:

```
/DEQ LUNAME 'NETWORK1.LUNAME1' TPNAME TPNAME1 PURGE
```

Response ET:

```
DFS1952 10:51:35 DESTINATION NOT FOUND
```

Explanation: An invalid APPC destination was entered.

*Example 6 for /DEQUEUE command*

Entry ET:

```
/DEQ LUNAME 'CICSNET.SERVER1' TPNAME DFSASYNCR PURGE1
```

Response ET:

```
DFS976I 14:10:25 1 MESSAGE(S) PURGED
```

Explanation: The message was dequeued from the APPC destination LUNAME CICSNET.SERVER, TPNAME DFSASYNCR.

**Related reference:**

Chapter 2, "Command keywords and their synonyms," on page 63

---

## Chapter 25. /DIAGNOSE commands

Use the /DIAGNOSE SNAP command to retrieve diagnostic information for system resources such as IMS control blocks, user-defined nodes, or user-defined transactions at any time without creating a console dump. Use the /DIAGNOSE SET command to enable or disable diagnostic features such as the logging facility that captures events related to APPC and OTMA synchronous transactions in a shared-queues environment.

The SET and SNAP keywords are mutually exclusive.

### Related concepts:

 /DIAGNOSE command SNAP function (Diagnosis)

---

## /DIAGNOSE SET command

Use the /DIAGNOSE SET command to enable or disable diagnostic features such as the logging facility that captures events related to APPC and OTMA synchronous transactions in a shared-queues environment.

Subsections:

- “Environment”
- “Syntax”
- “Keywords”
- “Examples” on page 480

### Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) from which the command can be issued.

Table 255. Valid environments for the /DIAGNOSE SET command and keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
/DIAGNOSE SET	X	X	X
AOSLOG	X		X
SYSOUT	X	X	X

### Syntax

→ /DIAGNOSE SET AOSLOG ( ON ) SYSOUT ( ON ) →  
/DIAG OFF OFF

### Keywords

You can use the /DIAGNOSE SET command to change the values of various IMS diagnostic options. The option to change is identified by the command keyword, and the desired state of the option is specified by the keyword value. For detailed information about option processing, see the specific keyword documentation. The SET keyword is mutually exclusive with the SNAP keyword.

The following keywords are valid for the /DIAGNOSE SET command:

#### **AOSLOG**

Specifies whether events related to APPC and OTMA synchronous transactions in a shared-queues environment are logged. Logged events are written to the OLDS as type X'6701' records.

If AOSLOG(ON) is specified in a non-shared-queues environment or when AOS=N is specified in the DFSDCxxx PROCLIB member, the command is rejected with a DFS2859I message.

**ON** Enables AOS logging.

#### **OFF**

Disables AOS logging.

#### **SYSOUT**

Enables or disables SYSOUT processing by the Diagnostic Services AWE server. If enabled, SYSOUT requests that are queued to the Diagnostic Services AWE server by the /DIAGNOSE command SYSOUT option are processed, and /DIAGNOSE SNAP command output is written to a SYSOUT data set. If disabled, SYSOUT requests that are queued to the Diagnostic Services AWE server are rejected.

**ON** Enables SYSOUT processing by the Diagnostic Services AWE server.

#### **OFF**

Disables SYSOUT processing by the Diagnostic Services AWE server.

### **Examples**

The following example shows the /DIAGNOSE SET command.

Entry ET:

```
/DIAGNOSE SET AOSLOG(ON)
```

Response ET:

```
DFS058I DIAGNOSE COMMAND COMPLETED
```

**Explanation:** The logging of events related to APPC and OTMA synchronous transactions in the shared-queues environment is enabled. The X'6701' records are captured and written to the IMS log.

**Related information:**

 [DFS2859I \(Messages and Codes\)](#)

---

## **/DIAGNOSE SNAP command**

Use the /DIAGNOSE SNAP command to capture storage information, and then format and display that information on the issuing LTERM. Optionally, the information can be written to the OLDS, trace data sets as type X'6701' records, or a SYSOUT data set. Storage information can be captured for IMS control blocks, user-defined nodes, transactions, programs, and database resources.

Subsections:

- "Environment" on page 481
- "Syntax" on page 481
- "Keywords" on page 483

- “Usage notes” on page 514
- “Examples” on page 515

## Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) from which the command can be issued.

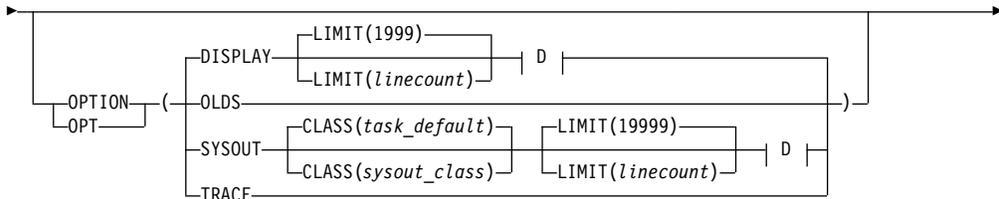
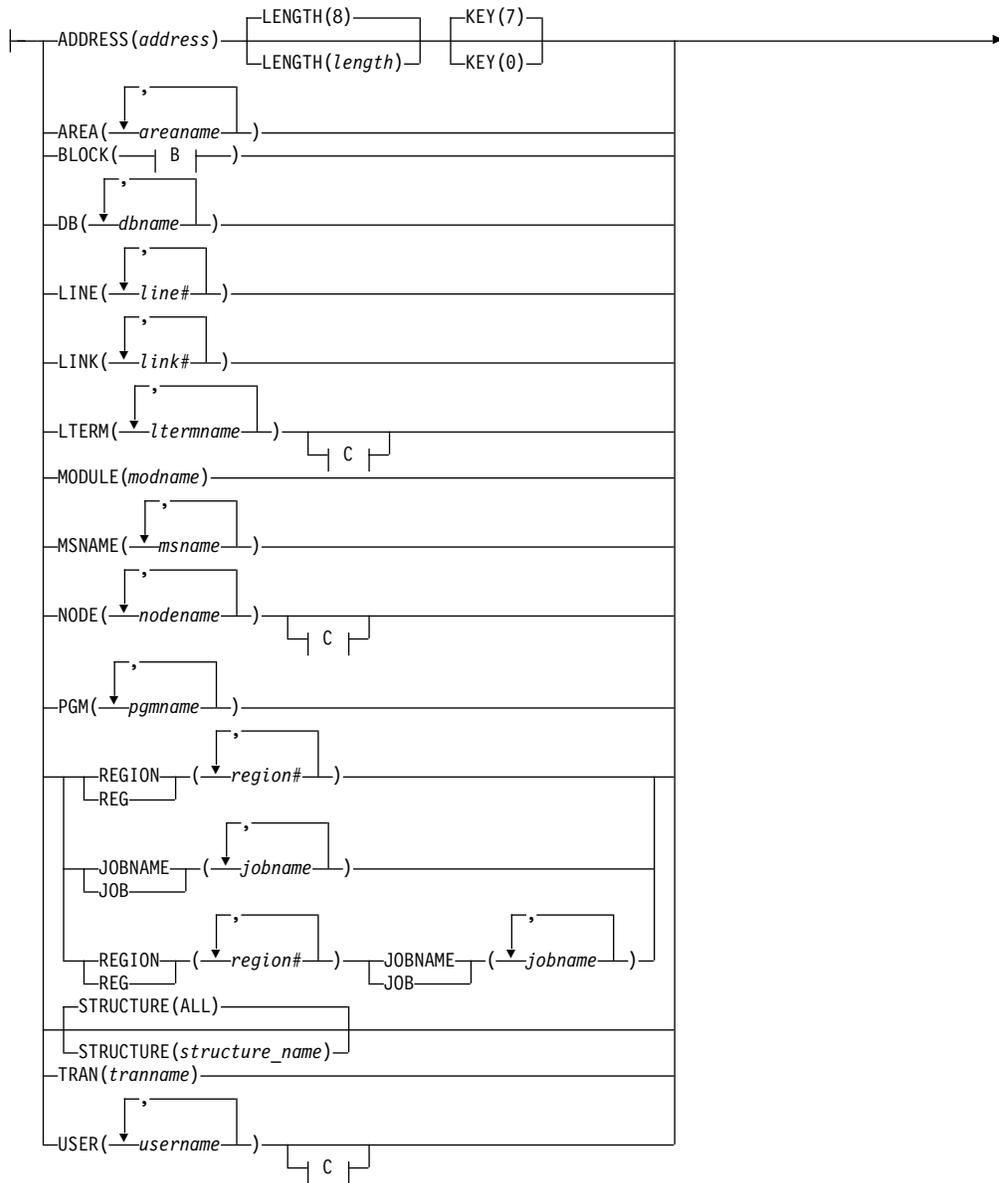
Table 256. Valid environments for the /DIAGNOSE SNAP command and keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
/DIAGNOSE SNAP	X	X	X
ADDRESS	X	X	X
AREA	X	X	
BLOCK	X	X	X
DB	X	X	
JOBNAME	X	X	X
LINE	X		X
LINK	X		X
LTERM	X		X
MODULE	X	X	X
MSNAME	X		X
NODE	X		X
OPTION	X	X	X
PGM	X	X	X
REGION	X	X	X
RM	X		X
SHOW	X	X	X
STRUCTURE	X		X
TRAN	X		X
USER	X		X

## Syntax



**A:**



**B:**

ALL
CATA
CMDE
CSCD
CSLA
CSVT
CTLASCB
CTLASSB
CTLJTCB
CTLTCB
DCCB
DFA
DGA
DGSD
DGSW
EDBT
ESCD
FRB
GDBT
IRCVECT
LCD
LCDM
LSCD
MSCD
MWA
PXPARDS
QSCD
RECA
RSR
RWCD
SCD
SDTT
SKCB
SQM
TIME
TRA
TSCD
UXDT

**C:**



**D:**



## Keywords

The following keywords are valid for the /DIAGNOSE SNAP command:

## ADDRESS

Captures information about a specific area of storage. The *address* value identifies the area and must specify a hexadecimal value between 0 and 7FFFFFFF.

## LENGTH

Specifies the length of the area of storage to capture. The LENGTH parameter is optional and has a default value of 8. If the LENGTH parameter is specified, the *length* value must be a numeric value from 1 to 65536.

## KEY

Specifies the key of the storage to capture. If the KEY parameter is included, the PSW key is changed to the value specified before accessing the storage. The KEY parameter is optional. Valid values for the KEY parameter are 0 and 7.

## AREA

Captures control block information for the Fast Path area specified in the *areaname* parameter. The *areaname* must be alphanumeric, no longer than 8 characters, and identify a currently defined Fast Path area. Multiple *areaname* parameters can be specified with each parameter separated by a comma or a blank.

The control blocks that can be captured for a Fast Path area are listed in the following table. Primary control blocks for an area, which are always present and available for capture, are identified by the column labeled "Primary". Optional control blocks for an area, which might be present and available for capture based on workload and other factors, are identified by the column labeled "Optional".

The /DIAGNOSE SNAP AREA() command is available in a DB/DC or DBCTL environment where Fast Path is defined.

The DEDB extended area control block (EMAC) is available only in an RSR tracker environment. If the SHOW(ALL) or SHOW(OPT) options are specified in a non-RST tracker environment, the EMAC is ignored. If the SHOW(EMAC) option is specified in a non-tracker environment, a DFS110I error message is issued in response.

Table 257. /DIAGNOSE SNAP AREA() control blocks

Name	Block description	Macro	Primary	Optional
ALDS	DEDB Area Name List Entry	DBFAREA	X	
DMAC	DEDB Area Control Block	DBFDMAC	X	
ADSC	DEDB Area Data Set Control Block	DBFADSC		X
DMHR	DEDB Buffer Header (SDEP)	DBFDMHR		X
EMAC	DEDB Extended Area Control Block	DBFEMAC		X
DDIR	Database Directory Block	DFSDDIR		X
DMCB	DEDB Master Control Block	DBFDMCB		X
MRMB	DEDB Randomizing Module Block	DBFDMRMB		X

Table 257. /DIAGNOSE SNAP AREA() control blocks (continued)

Name	Block description	Macro	Primary	Optional
DMSL	Data Space Map List	DBFDMSL		X
DSME	Data Space Mapping Entry	DBFDSME		X
DMAX	DMAC ERE Extension Block	DBFDMHV		X

**SHOW keyword parameters for the /DIAGNOSE SNAP AREA command**

**SHOW**

Used as a filtering mechanism to control which blocks are returned. Using the SHOW() option, a single block, list of blocks, or group of blocks identified by the keyword can be returned. The following example code shows this command option: /DIAG SNAP AREA(DEDDB01D1) SHOW(OPT).

If the SHOW() option is omitted from the /DIAGNOSE SNAP AREA() command, only the primary control blocks for the area are captured by default. The SNAP AREA() resource type supports the ALL, PRI, and OPT keywords, and all block names listed in Table 257 on page 484 as parameters to the SHOW() option. Multiple SHOW() option parameters can be specified with each parameter separated by a comma or a blank. A maximum of 16 SHOW() option parameters can be specified.

**ALL**

Captures all control blocks listed in Table 257 on page 484 that are available. Example: /DIAG SNAP AREA(DEDDB01D1) SHOW(ALL).

**PRI**

Captures the primary control blocks for the area. Primary control blocks are identified by the Primary column in Table 257 on page 484. Example: /DIAG SNAP AREA(DEDDB01D1) SHOW(PRI).

**OPT**

Captures all optional control blocks that are available. Optional control blocks are identified by the Optional column in Table 257 on page 484. Example: /DIAG SNAP AREA(DEDDB01D1) SHOW(OPT).

**blockname**

Captures the specified block by block name. Block names are defined in the Name column in Table 257 on page 484. Example: /DIAG SNAP AREA(DEDDB01D1) SHOW(ALDS).

**blockname,blockname**

Captures multiple blocks by block name. Example: /DIAG SNAP AREA(DEDDB01D1) SHOW(ALDS,DMAC).

**keyword,blockname**

Captures a group of blocks by keyword and individual blocks by block name. Example: /DIAG SNAP AREA(DEDDB01D1) SHOW(PRI,PDIR).

**BLOCK**

Captures information for a specific IMS control block. A control block name or ALL is required.

Multiple control block name parameters can be specified with each parameter separated by a comma or a blank.

Valid control block names include:

- ALL**  
Captures information for all valid control blocks currently available.
- CATA**  
Captures information for the Catalog Anchor control block. CATA is available only in a DB/DC or DBCTL environment.
- CMDE**  
Captures information for the Commands SCD Extension control block.
- CSCD**  
Captures storage information for the APPC/OTMA SMQ SCD Extension control block. CSCD is available only in a DB/DC or DCCTL environment.
- CSLA**  
Captures information for the Common Service Layer Anchor control block.
- CSVT**  
Captures information for the Common Callable Services Vector Table control block.
- CTLASCB**  
Captures information for the IMS control region Address Space Control control block.
- CTLASSB**  
Captures information for the IMS control region Address Space Secondary control block.
- CTLJTCB**  
Captures information for the IMS control region Job Step Task Control Block.
- CTLTCB**  
Captures information for the IMS control region Primary Task Control Block.
- DCCB**  
Captures information for the Data Communications control block. DCCB is available only in a DB/DC or DCCTL environment.
- DFA**  
Captures information for the Definition Anchor control block.
- DGA**  
Captures information for the Diagnostics Anchor control block.
- DGSD**  
Captures information for the Diagnostic Data Set Structures control block.
- DGSW**  
Captures information for the Diagnose Work Area Storage control block.
- EDBT**  
Captures information for the RSR FP Global DB Tracking control block. EDBT is available only in an IMS system where Fast Path is defined.

**ESCD**

Captures information for the Extended System Contents Directory control block. ESCD is available only in an IMS system where Fast Path is defined.

**FRB**

Captures information for the Fast Restart control block. FRB is available only in a DB/DC or DBCTL environment.

**GDBT**

Captures information for the System RSR Global DB Tracking control block. GDBT is available only in an DB/DC or DBCTL environment.

**IRCVECT**

Captures information for the Log SVC Vector Table control block.

**LCD**

Captures information for the Log Control Directory (Journal) control block.

**LCDM**

Captures information for the Log Control Directory (Monitor) control block.

**LSCD**

Captures information for the APPC SCD Extension control block. LSCD is available only in a DB/DC or DCCTL environment.

**MSCD**

Captures information for the MSC SCD Extension control block. MSCD is available only in a DB/DC or DCCTL environment.

**MWA**

Captures information for the Modify Work Area control block.

**PXPARMS**

Captures information for the Modify Parameter Anchor control block.

**QSCD**

Captures information for the Queue Manager SCD Extension control block. QSCD is available only in a DB/DC or DCCTL environment.

**RECA**

Captures information for the VTAM Receive Any I/O Buffers control blocks. RECA is available only in a DB/DC or DCCTL environment.

Because of the volume of data that can be produced by capturing the VTAM Receive Any I/O Buffers, the RECA option is not included in the BLOCK(ALL) option.

**RSR**

Captures information for the Remote Site Recovery Anchor control block. RSR is available only in a DB/DC or DBCTL environment.

**RWCD**

Captures information for the Database Recovery 2 Anchor control block. RWCD is available only in a DB/DC or DBCTL environment.

**SCD**

Captures information for the System Contents Directory control block.

**SDTT**

Captures information for the Shutdown Trace Table control block.

**SKCB**

Captures information for the Stack Control Block.

**SQM**

Captures information for the Shared Queues Master control block. SQM is available only in a shared-queues environment.

**TIME**

Captures information for the Timer Services SCD Extension control block.

**TRA**

Captures information for the Table Trace control block.

**TSCD**

Captures information for the OTMA SCD Extension control block. TSCD is available only in a DB/DC or DCCTL environment.

**UXDT**

Captures information for the User Exit Definition Table control block.

The following table shows BLOCK() resources listed by the area.

*Table 258. /DIAGNOSE SNAP BLOCK() - System control blocks*

<b>Name</b>	<b>Block description</b>	<b>Macro</b>
CMDE	Commands SCD Extension	DFSCMDE
CSLA	Common Service Layer Anchor Block	DFSCSLA
CSVT	Callable Services Vector Table	DFSCSVT
CTLASCB	CTL Address Space Control Block	IHAASCB
CTLASSB	CTL Address Space Secondary Block	IHAASSB
CTLJTCB	CTL Job Step Task Control Block	IKJTCB
CTLTCB	CTL Primary Task Control Block	IKJTCB
DFA	Definition Anchor Block	DFSDFA
DGA	Diagnostics Anchor Block	DFSDGA
DGSD	Diagnostic Data Set Structures	DFSDGSWA
DGSW	Diagnose Work Area Storage	DFSDGSWA
IRCVECT	SVC Vector Table	IRCVECT
LCD	Log Control Directory (Journal)	LCDSECT
LCDM	Log Control Directory (Monitor)	LCDSECT
MWA	Modify Work Area	DFSMWA
PXPARMS	Parameter Anchor Block	PARMBLK
SCD	System Contents Directory	ISCD
SDTT	Shutdown Trace Table	DFSSDTT
SKCB	Stack Control Block	DFSSKCB

Table 258. /DIAGNOSE SNAP BLOCK() - System control blocks (continued)

Name	Block description	Macro
TIME	Timer Services SCD Extension	DFSTIME
TRA	Table Trace Control Block	DFSTRABK
UXDT	User Exit Definition Table	DFSUSRXD

Table 259. /DIAGNOSE SNAP BLOCK() - DB control blocks

Name	Block description	Macro
CATA	Catalog Anchor Block	DFS3ACH
EDBT	RSR FP Global DB Tracking Block	DBFEDBT
ESCD	Extended System Contents Directory	DBFESCD
FRB	Fast Restart Block	DFSFRB
GDBT	RSR Global DB Tracking Block	DFSGDBT
RSR	Remote Site Recovery Anchor Block	DFSRSCD
RWCD	Database Recovery 2 Anchor Block	DFSRWCD

Table 260. /DIAGNOSE SNAP BLOCK() - TM control blocks

Name	Block description	Macro
CSCD	APPC/OTMA SMQ SCD Extension	DFSCSCD
DCCB	Data Communications Control Block	DFSDCCB
LSCD	APPC SCD Extension	DFSLSCD
MSCD	MSC SCD Extension	DFSMSCD
QSCD	Queue Manager SCD Extension	DFSQSCD
RECA	VTAM Receive Any I/O Buffers	BUFVTPRE
SQM	Shared Queues Master Control Block	DFSSQM
TSCD	OTMA SCD Extension	DFSTSCD

**DB** Captures control block information for the database specified in the *dbname* parameter. The *dbname* parameter specified must be alphanumeric, no longer than 8 characters, and identify a currently defined database. Multiple *dbname* parameters can be specified with each parameter separated by a comma or a blank.

The control blocks that can be captured for a database are listed in the following table. Primary control blocks for a database, which are always present and available for capture, are identified by the column labeled "Primary". Optional control blocks for a database, which might or might

not be present and available for capture depending on workload, database type and other factors, are identified by the column labeled "Optional".

Table 261. /DIAGNOSE SNAP DB() control blocks

Name	Block description	Macro	Primary	Optional
DDIR	Database Directory Block	DFSDDIR	X	
DDIREXT	Database Directory Block Extension	DFSDDIR		X
RSCX	Resource Extension Block	DFSRSCX		X
DYNALMBR	Dynamic Allocate Member	DFSMDA		X
DBQLE	Database Quiesce List Entry	DFSDBQQL		X
EEQE	Error Queue Element	DFSEEQE		X
RRE	Residual Recovery Element	DFSRRE		X
TDBC	Tracking Data Base Control Block	DFSTDBC		X
SDTE	Segment Delta Block Table Entry	DFS5FLDD		X
DMB	Data Management Block	DFSDMB		X
PSDB	Physical Segment Descriptor Block	DFSDMB		X
SDB	Segment Descriptor Block	DFSSDBM		X
FDB	Field Descriptor Block	DFSFDB		X
DMBCPAC	Segment Edit/Compression Block	DFSDMB		X
DMBSEC	DMB Secondary List	DFSDMB		X
DMBDACS	Randomizer Control Block	DFSDMB		X
DMBAMPPR	Access Method Prefix Block Prefix	DFSDMB		X
DMBAMP	Access Method Prefix Block	DFSDMB		X
DCBACBP	Primary DCB/ACB Block	DCBD		X
DCBACBS	Overflow DCB/ACB Block	DCBD		X
DMBXBLCK	Data Management Exit Block	DFSDMB		X
DMBXARRY	Exit Array Entry Block	DFSDMB		X
DMBXT	Exit Description Block	DFSDMB		X
DMCB	DEDB Master Control Block	DBFDMCB		X
BHDR	Main Storage Database Header	DBFBMSDB		X

**SHOW keyword parameters for the /DIAGNOSE SNAP DB command**

**SHOW**

Specifies which blocks are returned. Using the SHOW() option, a single block, list of blocks, or group of blocks identified by the keyword can be returned. Example: /DIAG SNAP DB(IVPDB1) SHOW(DDIR)

If the SHOW keyword is omitted from the /DIAGNOSE SNAP DB command, only the primary control blocks for the database are captured by default.

The SNAP DB() resource type supports the ALL, OPT, and PRI keywords, and all block names listed in Table 261 as parameters for the SHOW keyword. Multiple SHOW parameters can be specified with each parameter separated by a comma or a blank. A maximum of 16 SHOW parameters can be specified.

The effects of the SHOW option on the data captured by the /DIAGNOSE SNAP DB command are described in the following list:

**ALL**

Captures all control blocks listed in Table 261 on page 490 that are available. Example: /DIAG SNAP DB(IVPDB1) SHOW(ALL).

**OPT**

Captures all optional control blocks that are available. Optional control blocks are identified by the Optional column in Table 261 on page 490. Example: /DIAG SNAP DB(IVPDB1) SHOW(OPT).

**PRI**

Captures the primary control blocks for the database. Primary control blocks are identified by the Primary column in Table 261 on page 490. Example: /DIAG SNAP DB(IVPDB1) SHOW(PRI).

**blockname**

Captures the specified block by block name. Block names are defined in the Name column in Table 261 on page 490. Example: /DIAG SNAP DB(IVPDB1) SHOW(DDIR).

**blockname, blockname**

Captures multiple blocks by block name. Example: /DIAG SNAP DB(IVPDB1) SHOW(DDIR,DMCB).

**keyword, blockname**

Captures a group of blocks by keyword and individual blocks by block name. Example: /DIAG SNAP DB(IVPDB1) SHOW(PRI,DDIR).

*Usage notes for the /DIAGNOSE SNAP DB command*

The /DIAGNOSE SNAP DB command is available only in a DB/DC or DBCTL environment.

**LINE** Captures control block information for the communication line specified in the *line#* parameter. The *line#* parameter specified must be numeric, in the range 1 - 1000, and identify a currently defined communication line. Multiple *line#* parameters can be specified with each parameter separated by a comma or a blank.

The control blocks that can be captured for a communication line are listed in the following table. Primary control blocks for a communication line, which are always present and available for capture, are identified by the column labeled "Primary". Optional control blocks for a communication line, which might be present and available for capture depending on workload and other factors, are identified by the column labeled "Optional".

Table 262. /DIAGNOSE SNAP LINE() control blocks

Name	Block description	Macro	Primary	Optional
EPF	Event Control Block Prefix	IEPF	X	
CULE	Common Use List Element Block	DFSCULE		X
DSPWRK1	Dispatcher Work Area	IDSPWRK		X
SAP	Save Area Prefix	ISAP		X
SAVEAREA	Save Area Set	REQUATE		X
PROLOG	Module Prolog Information	N/A		X
CLB	Communication Line Block	ICLI	X	

Table 262. /DIAGNOSE SNAP LINE() control blocks (continued)

Name	Block description	Macro	Primary	Optional
CTB	Communication Terminal Block	ICLI		X
CTT	Communication Translate Table	ICLI		X
CRB	Communications Restart Block	ICLI		X
SPQB	Subpool Queue Block	ICLI		X
SPQBEXT	Subpool Queue Extension Block	ICLI		X
EMHB	Expedited Message Handler Block	DBFEMHB		X
SMB	Scheduler Message Block	IAPS		X
TIB	APPC Transaction Instance Block	DFSTIB		X
YTIB	OTMA Transaction Instance Block	DFSYTIB		
CNT	Communication Name Table	ICLI		X
ECNT	Extended Communication Name Table	DBFECNT		X
CRTCN	Command CART & 4-Byte Console ID	DFSMCSC		X
CCB	Conversational Control Block	ICLI		X
CIB	Communication Interface Block	ICLI		X
MSGBP	Basic 01/03 Message Prefix	QLOGMSGP		X
UOWE	Unit of Work Table Entry	DFSUOWE		X
INBUF	Input Line Buffer	N/A		X
OUTBUF	Output Line Buffer	N/A		X
RAQE	VTAM Receive Any IO Buffer (In)	BUFVTPRE		X
RAQERES	VTAM Receive Any IO Buffer (Out)	BUFVTPRE		X

**SHOW keyword parameters for the /DIAGNOSE SNAP LINE command**

**SHOW**

Used as a filtering mechanism to control which blocks are returned. Using the SHOW() option, a single block, list of blocks, or group of blocks identified by the keyword can be returned. Example: /DIAG SNAP LINE(1)

If the SHOW() option is omitted from the /DIAGNOSE SNAP LINE() command, only the primary control blocks for the communication line are captured by default. The SNAP LINE() resource type supports the ALL, PRI, and OPT keywords, and all block names listed in Table 262 on page 491 as parameters to the SHOW() option. Multiple SHOW() option parameters can be specified with each parameter separated by a comma or a blank. A maximum of 16 SHOW() option parameters can be specified.

**ALL**

Captures all control blocks listed in Table 262 on page 491 that are available. Example: /DIAG SNAP LINE(1) SHOW(ALL)

**PRI**

Captures the primary control blocks for the communication line. Primary control blocks are identified by the Primary column in Table 262 on page 491. Example: /DIAG SNAP LINE(9) SHOW(PRI)

**OPT**

Captures all optional control blocks that are available. Optional

control blocks are identified by the Optional column in Table 262 on page 491. Example: /DIAG SNAP LINE(9) SHOW(OPT)

***blockname***

Captures the specified block by block name. Block names are defined in the Name column in Table 262 on page 491. Example: /DIAG SNAP LINE(1) SHOW(CLB)

***blockname,blockname***

Captures multiple blocks by block name. Example: /DIAG SNAP LINE(1) SHOW(CLB,CTT)

***keyword,blockname***

Captures a group of blocks by keyword and individual blocks by block name. Example: /DIAG SNAP LINE(1) SHOW(PRI,PDIR)

***Usage notes for the /DIAGNOSE SNAP LINE command***

The /DIAGNOSE SNAP LINE command is available in a DB/DC or DCCTL environment.

The SHOW(PROLOG) option displays information about the modules that are associated with the save areas in the chain. The module entry point is obtained from the saved register 15 value of the previous save area. If SHOW(SAVEAREA) is combined with SHOW(PROLOG), the display consists of a save area followed by the related prolog of the module that uses the save area.

The following control blocks are available only in an IMS system where Fast Path is defined.

- ECNT (Extended Communication Name Table)
- EMHB (Expedited Message Handler Block)
- SMB (Scheduler Message Block)
- TIB (APPC Transaction Instance Block)
- YTIB (OTMA Transaction Instance Block)

If the SHOW(ALL) or SHOW(OPT) option is specified in a non-Fast Path environment, these blocks are ignored. If the SHOW(ECNT), SHOW(EMHB), SHOW(SMB), SHOW(TIB), or SHOW(YTIB) option is specified in a non-Fast Path environment, a DFS154I message is issued in response.

UOWE (Unit of Work Table Entry) block is available only in an IMS system where shared queues are defined. If the SHOW(ALL) or SHOW(APP) option is specified in a non-shared-queues environment, UOWE is ignored. If the SHOW(UOWE) option is specified in a non-shared-queues environment, a DFS154I message is issued in response.

TIB (APPC Transaction Instance Block) and YTIB (OTMA Transaction Instance Block) are mutually exclusive; only one is present at any given time. For SHOW() purposes, TIB and YTIB are treated as synonyms. Specifying either TIB or YTIB displays, if available, whichever of the two blocks is present.

The SHOW(RECANY) option displays information about both RAQE (VTAM Receive Any IO Buffer (In)) and RAQERES (VTAM Receive Any IO Buffer (Out)) control blocks.

The SHOW(SA) option is a synonym for the SHOW(SAVEAREA) option.

**LINK** Captures control block information for the logical link specified in the *link#*

parameter. The *link#* parameter specified must be numeric, in the range 1 - 936, and identify a currently defined logical link. Multiple *link#* parameters can be specified with each parameter separated by a comma or a blank.

The control blocks that can be captured for a logical link are listed in the following table. Primary control blocks for a logical link, which are always present and available for capture, are identified by the column labeled "Primary". Optional control blocks for a logical link, which might be present and available for capture depending on workload and other factors, are identified by the column labeled "Optional".

Table 263. /DIAGNOSE SNAP LINK() control blocks

Name	Block description	Macro	Primary	Optional
EPF	Event Control Block Prefix	IEPF	X	
CULE	Common Use List Element Block	DFSCULE		X
DSPWRK1	Dispatcher Work Area	IDSPWRK		X
SAP	Save Area Prefix	ISAP		X
SAVEAREA	Save Area Set	REQUATE		X
PROLOG	Module Prolog Information	N/A		X
LLB	Link Line Block	ICLI	X	
LTB	Link Terminal Block	ICLI		X
CTT	Communication Translate Table	ICLI		X
CRB	Communications Restart Block	ICLI		X
SPQB	Subpool Queue Block	ICLI		X
SPQBEXT	Subpool Queue Extension Block	ICLI		X
EMHB	Expedited Message Handler Block	DBFEMHB		X
SMB	Scheduler Message Block	IAPS		X
TIB	APPC Transaction Instance Block	DFSTIB		X
YTIB	OTMA Transaction Instance Block	DFSYTIB		X
LNB	Link Name Block	ICLI		X
ECNT	Extended Communication Name Table	DBFECNT		X
CRTCN	Command CART & 4-Byte Console ID	DFSMCSC		X
CCB	Conversational Control Block	ICLI		X
CIB	Communication Interface Block	ICLI		X
MSGBP	Basic 01/03 Message Prefix	QLOGMSGP		X
UOWE	Unit of Work Table Entry	DFSUOWE		X
INBUF	Input Line Buffer	N/A		X
OUTBUF	Output Line Buffer	N/A		X
LCB	Link Control Block	LCB		X
LXB	Link Extension Block	LXB		X

### *SHOW keyword parameters for the /DIAGNOSE SNAP LINK command*

#### **SHOW**

Used as a filtering mechanism to control which blocks are returned. Using the SHOW() option, a single block, list of blocks, or group of blocks identified by the keyword can be returned. The following example code shows this command option: /DIAG SNAP LINK(9)

If the SHOW() option is omitted from the /DIAGNOSE SNAP LINK() command, only the primary control blocks for the logical line are captured by default. The SNAP LINK() resource type supports the ALL, PRI, and OPT keywords, and all block names that are listed in Table 263 on page 494 as parameters to the SHOW() option. Multiple SHOW() option parameters can be specified with each parameter separated by a comma or a blank. A maximum of 16 SHOW() option parameters can be specified.

**ALL**

Captures all control blocks listed in Table 263 on page 494 that are available. Example: /DIAG SNAP LINK(9) SHOW(ALL)

**PRI**

Captures the primary control blocks for the logical line. Primary control blocks are identified by the Primary column in Table 263 on page 494. Example: /DIAG SNAP LINK(9) SHOW(PRI)

**OPT**

Captures all optional control blocks that are available. Optional control blocks are identified by the Optional column in Table 263 on page 494. Example: /DIAG SNAP LINE(9) SHOW(OPT)

***blockname***

Captures the specified block by block name. Block names are defined in the Name column in Table 263 on page 494. Example: /DIAG SNAP LINK(9) SHOW(CLB)

***blockname,blockname***

Captures multiple blocks by block name. Example: /DIAG SNAP LINK(9) SHOW(LLB,CTT)

***keyword,blockname***

Captures a group of blocks by keyword and individual blocks by block name. Example: /DIAG SNAP LINK(9) SHOW(PRI,PDIR)

***Usage notes for the /DIAGNOSE SNAP LINK command***

The /DIAGNOSE SNAP LINK() command is available only in a DB/DC or DCCTL environment.

The SHOW(PROLOG) option displays information about the modules that are associated with the save areas in the chain. The module entry point is obtained from the saved register 15 value of the previous save area. If SHOW(SAVEAREA) is combined with SHOW(PROLOG), the display consists of a save area followed by the related prolog of the module that uses the save area.

The following control blocks are available only in an IMS system where Fast Path is defined.

- ECNT (Extended Communication Name Table)
- EMHB (Expedited Message Handler Block)
- SMB (Scheduler Message Block)
- TIB (APPC Transaction Instance Block)
- YTIB (OTMA Transaction Instance Block)

If the SHOW(ALL) or SHOW(OPT) option is specified in a non-Fast Path environment, these blocks are ignored. If the SHOW(ECNT),

SHOW(EMHB), SHOW(SMB), SHOW(TIB), or SHOW(YTIB) option is specified in a non-Fast Path environment, a DFS154I message is issued in response.

UOWE (Unit of Work Table Entry) block is available only in an IMS system where shared queues are defined. If the SHOW(ALL) or SHOW(APP) option is specified in a non-shared-queues environment, UOWE is ignored. If the SHOW(UOWE) option is specified in a non-shared-queues environment, a DFS154I message is issued in response.

TIB (APPC Transaction Instance Block) and YTIB (OTMA Transaction Instance Block) are mutually exclusive; only one is present at any given time. For SHOW() purposes, TIB and YTIB are treated as synonyms. Specifying either TIB or YTIB displays, if available, whichever of the two blocks is present.

The SHOW(SA) option is a synonym for the SHOW(SAVEAREA) option.

### LTERM

Captures control block information for the logical terminal specified in the *ltermname* parameter. The *ltermname* parameter specified must be alphanumeric, no longer than 8 characters, and identify a currently defined logical terminal. Multiple *ltermname* parameters can be specified with each parameter separated by a comma or a blank.

The control blocks that can be captured for logical terminals are listed in the following table. Primary control blocks for a logical terminal, which are always present and available for capture, are identified by the column labeled "Primary". Optional control blocks for a logical terminal, which might or might not be present and available for capture depending on workload and other factors, are identified by the column labeled "Optional".

Table 264. /DIAGNOSE SNAP LTERM() control blocks

Name	Block description	Macro	Primary	Optional
CNT	Communication Name Table (Target)	ICLI	X <sup>1</sup>	
RCNT	Remote Communication Name Table	RCNT	X <sup>2</sup>	
LNB	Link Name Block	LNB	X <sup>2</sup>	
EPF	Event Control Block Prefix	IEPF		X
CULE	Common Use List Element Block	DFSCULE		X
DSPWRK1	Dispatcher Work Area	IDSPWRK		X
SAP	Save Area Prefix	ISAP		X
SAVEAREA	Save Area Set	REQUATE		X
PROLOG	Module Prolog Information	N/A		X
CLB	Communication Line Block	ICLI		X
CTB	Communication Terminal Block	ICLI		X
CTT	Communication Translate Table	ICLI		X
CRB	Communications Restart Block	ICLI		X
SPQB	Subpool Queue Block	ICLI		X
SPQBEXT	Subpool Queue Extension Block	ICLI		X
EMHB	Expedited Message Handler Block	DBFEMHB		X
SMB	Scheduler Message Block	IAPS		X

Table 264. /DIAGNOSE SNAP LTERM() control blocks (continued)

Name	Block description	Macro	Primary	Optional
TIB	APPC Transaction Instance Block	DFSTIB		X
YTIB	OTMA Transaction Instance Block	DFSYTIB		X
CNT	Communication Name Table	ICLI		X
ECNT	Extended Communication Name Table	DBFECNT		X
CRTCN	Command CART & 4-Byte Console ID	DFSMCSC		X
CCB	Conversational Control Block	ICLI		X
CIB	Communication Interface Block	ICLI		X
MSGBP	Basic 01/03 Message Prefix	QLOGMSGP		X
UOWE	Unit of Work Table Entry	DFSUOWE		X
INBUF	Input Line Buffer	N/A		X
OUTBUF	Output Line Buffer	N/A		X
RAQE	VTAM Receive Any IO Buffer (In)	BUFVTPRE		X
RAQRES	VTAM Receive Any IO Buffer (Out)	BUFVTPRE		X

<sup>1</sup> Primary blocks for a local logical terminal

<sup>2</sup> Primary blocks for a remote logical terminal

### *SHOW keyword parameters for the /DIAGNOSE SNAP LTERM command*

#### **SHOW**

Specifies which blocks are returned. If the SHOW keyword is omitted from the /DIAGNOSE SNAP LTERM command, only the primary control blocks for the logical terminal are captured by default. The SNAP LTERM() resource type supports the ALL, DEF (or DEFAULT), OPT, PRI, and TAR (or TARGET) keywords, and all block names that are listed in Table 264 on page 496 as parameters for the SHOW keyword. Multiple SHOW parameters can be specified with each parameter separated by a comma or a blank. A maximum of 32 SHOW parameters can be specified.

The effects of the SHOW option on the data that is captured by the /DIAGNOSE SNAP LTERM command are described in the following list:

#### **ALL**

Captures all control blocks listed in Table 264 on page 496 that are available. Example: /DIAG SNAP LTERM(MTRL) SHOW(ALL).

#### **DEF**

Captures information about default blocks for the current logical terminal. DEF, or DEFAULT, is equivalent to the PRI keyword or without any SHOW parameter. Example: /DIAGNOSE SNAP LTERM(MTRL) SHOW(DEF).

#### **OPT**

Captures all optional control blocks that are available. Optional control blocks are identified by the Optional column in Table 264 on page 496. Example: /DIAG SNAP LTERM(MTRL) SHOW(OPT).

#### **PRI**

Captures the primary control blocks for the logical terminal.

Primary control blocks are identified by the Primary column in Table 264 on page 496. Example: /DIAG SNAP LTERM(MTRL) SHOW(PRI).

#### **TARGET**

Captures the target CNT (Communication Name Table) for a local logical terminal (LTERM). TARGET can be abbreviated as TAR.

If the SHOW(TARGET) option is specified and the only resource specified is a remote logical terminal, a DFS2859I informational message is issued with the following reason text: NO VALID BLOCKS SPECIFIED FOR LTERM(*ltermname*). If other SHOW() options are included that are applicable to a remote logical terminal, those blocks are displayed, and the TARGET option is ignored.

If the SHOW(TARGET) option is specified and both local and remote logical terminal resources are specified, a DFS2859I message is issued for the remote logical terminal.

Example: /DIAGNOSE SNAP LTERM(MTRL) SHOW(TAR).

#### ***blockname***

Captures the specified block by block name. Block names are defined in the Name column in Table 264 on page 496. Example: /DIAG SNAP LTERM(MTRL) SHOW(CLB).

#### ***blockname,blockname***

Captures multiple blocks by block name. Example: /DIAG SNAP LTERM(MTRL) SHOW(CLB,CTB).

#### ***keyword,blockname***

Captures a group of blocks by keyword and individual blocks by block name. Example: /DIAG SNAP LTERM(MTRL) SHOW(PRI,CTB).

#### ***Usage notes for the /DIAGNOSE SNAP LTERM command***

The /DIAGNOSE SNAP LTERM command is available in a DB/DC or DCCTL environment.

The SHOW(PROLOG) option displays information about the modules that are associated with the save areas in the chain. The module entry point is obtained from the saved register 15 value of the previous save area. If SHOW(SAVEAREA) is combined with SHOW(PROLOG), the display consists of a save area followed by the related prolog of the module that uses the save area.

The following control blocks are available only in an IMS system where Fast Path is defined.

- ECNT (Extended Communication Name Table)
- EMHB (Expedited Message Handler Block)
- SMB (Scheduler Message Block)
- TIB (APPC Transaction Instance Block)
- YTIB (OTMA Transaction Instance Block)

If the SHOW(ALL) or SHOW(OPT) option is specified in a non-Fast Path environment, these blocks are ignored. If the SHOW(ECNT), SHOW(EMHB), SHOW(SMB), SHOW(TIB), or SHOW(YTIB) option is specified in a non-Fast Path environment, a DFS154I message is issued in response.

UOWE (Unit of Work Table Entry) block is available only in an IMS system where shared queues are defined. If the SHOW(ALL) or SHOW(APP) option is specified in a non-shared-queues environment, UOWE is ignored. If the SHOW(UOWE) option is specified in a non-shared-queues environment, a DFS154I message is issued in response.

If the SHOW(RCNT), SHOW(LNB), or SHOW(RCNT,LNB) option is specified and the only resource specified is a local logical terminal, a DFS2859I informational message is issued. If other SHOW() options are included that are applicable to a local logical terminal, those blocks are displayed, and the RCNT or LNB option is ignored.

If the SHOW(RCNT), SHOW(LNB), or SHOW(RCNT,LNB) option is specified and both local and remote logical terminal resources are specified, a DFS2859I message is issued for the local logical terminal.

TIB (APPC Transaction Instance Block) and YTIB (OTMA Transaction Instance Block) are mutually exclusive; only one is present at any given time. For SHOW() purposes, TIB and YTIB are treated as synonyms. Specifying either TIB or YTIB displays, if available, whichever of the two blocks is present.

The SHOW(RECANY) option displays information about both RAQE (VTAM Receive Any IO Buffer (In)) and RAQERES (VTAM Receive Any IO Buffer (Out)) control blocks.

The SHOW(SA) option is a synonym for the SHOW(SAVEAREA) option.

## MODULE

Returns the entry point address and captures in-storage prolog information about the IMS module specified in the *modname* parameter. The standard IMS module in-storage prolog information contains the module name, product level, assembly date and time, last APAR ID, module maintenance level, BPE version, and release (for BPE-based modules), and copyright statement.

**Note:** The name of the IMS module that appears in the in-storage prolog information might not be the same as the loaded module name if binder options were used to declare a different name for the loaded module.

MODULE(*modname*) enables IBM Software Support representatives to see the maintenance level of the installed software.

## MSNAME

Captures control block information for the MSNAME specified in the *msname* parameter. The *msname* parameter specified must be alphanumeric, no longer than 8 characters, and identify a currently defined MSNAME. Multiple *msname* parameters can be specified with each parameter separated by a comma or a blank.

The control blocks that can be captured for MSNAMEs are listed in the following table. Primary control blocks for an MSNAME, which are always present and available for capture, are identified by the column labeled "Primary". Optional control blocks for an MSNAME, which might or might not be present and available for capture depending on workload and other factors, are identified by the column labeled "Optional".

Table 265. /DIAGNOSE SNAP MSNAME() control blocks

Name	Block description	Macro	Primary	Optional
LNB	Link Name Block (Target)	LNB	X	

Table 265. /DIAGNOSE SNAP MSNAME() control blocks (continued)

Name	Block description	Macro	Primary	Optional
EPF	Event Control Block Prefix	IEPF		X
CULE	Common Use List Element Block	DFSCULE		X
DSPWRK1	Dispatcher Work Area	IDSPWRK		X
SAP	Save Area Prefix	ISAP		X
SAVEAREA	Save Area Set	REQUATE		X
PROLOG	Module Prolog Information	N/A		X
LLB	Link Line Block	ICLI		X
LTB	Link Terminal Block	ICLI		X
CTT	Communication Translate Table	ICLI		X
CRB	Communications Restart Block	ICLI		X
SPQB	Subpool Queue Block	ICLI		X
SPQBEXT	Subpool Queue Extension Block	ICLI		X
EMHB	Expedited Message Handler Block	DBFEMHB		X
SMB	Scheduler Message Block	IAPS		X
TIB	APPC Transaction Instance Block	DFSTIB		X
YTIB	OTMA Transaction Instance Block	DFSYTIB		X
LNB	Link Name Block	ICLI		X
ECNT	Extended Communication Name Table	DBFECNT		X
CRTCN	Command CART & 4-Byte Console ID	DFSMCSC		X
CCB	Conversational Control Block	ICLI		X
CIB	Communication Interface Block	ICLI		X
MSGBP	Basic 01/03 Message Prefix	QLOGMSGP		X
UOWE	Unit of Work Table Entry	DFSUOWE		X
INBUF	Input link buffer	N/A		X
OUTBUF	Output link buffer	N/A		X
LCB	Link Control Block	LCB		X
LXB	Link Extension Block	LXB		X

**SHOW keyword parameters for the /DIAGNOSE SNAP MSNAME command**

**SHOW**

Specifies which blocks are to be returned. If the SHOW keyword is omitted from the /DIAGNOSE SNAP NODE command, only the primary control blocks for the MSNAME are captured by default. The SNAP MSNAME resource type supports the ALL, PRI, OPT, and TAR (or TARGET) keywords, and all block names that are listed in Table 265 on page 499 as parameters for the SHOW keyword. Multiple SHOW parameters can be specified with each parameter separated by a comma or a blank. A maximum of 32 SHOW parameters can be specified.

The effects of the SHOW option on the data that is captured by the /DIAGNOSE SNAP MSNAME command are described in the following list:

**ALL**

Captures all control blocks listed in Table 265 on page 499 that are available. Example: /DIAGNOSE SNAP MSNAME(MSN1) SHOW(ALL).

**OPT**

Captures all optional control blocks that are available. Optional control blocks are identified by the Optional column in Table 265 on page 499. Example: /DIAGNOSE SNAP MSNAME(MSN1) SHOW(OPT).

**PRI**

Captures the primary control blocks for the MSNAME. Primary control blocks are identified by the Primary column in Table 265 on page 499. Example: /DIAGNOSE SNAP MSNAME(MSN1) SHOW(PRI).

**TARGET**

Captures the target LNB (Link Name Block) for the MSNAME. TARGET can be abbreviated as TAR. Example: /DIAGNOSE SNAP MSNAME(MTR1) SHOW(TAR).

***blockname***

Captures the specified block by block name. Block names are defined in the Name column in Table 265 on page 499. Example: /DIAGNOSE SNAP MSNAME(MSN1) SHOW(CLB).

***blockname,blockname***

Captures multiple blocks by block name. Example: /DIAGNOSE SNAP MSNAME(MSN1) SHOW(CLB,CTB).

***keyword,blockname***

Captures a group of blocks by keyword and individual blocks by block name. Example: /DIAG SNAP MSNAME(MSN1) SHOW(PRI,CTB).

***Usage notes for the /DIAGNOSE SNAP MSNAME command***

The /DIAGNOSE SNAP MSNAME command is available only in a DB/DC or DCCTL environment.

The following control blocks are available only in an IMS system where Fast Path is defined.

- ECNT (Extended Communication Name Table)
- EMHB (Expedited Message Handler Block)
- SMB (Scheduler Message Block)
- TIB (APPC Transaction Instance Block)
- YTIB (OTMA Transaction Instance Block)

If the SHOW(ALL) or SHOW(OPT) option is specified in a non-Fast Path environment, these blocks are ignored. If the SHOW(ECNT), SHOW(EMHB), SHOW(SMB), SHOW(TIB), or SHOW(YTIB) option is specified in a non-Fast Path environment, a DFS154I message is issued in response.

UOWE (Unit of Work Table Entry) block is available only in an IMS system where shared queues are defined. If the SHOW(ALL) or SHOW(APP) option is specified in a non-shared-queues environment, UOWE is ignored. If the SHOW(UOWE) option is specified in a non-shared-queues environment, a DFS154I message is issued in response.

TIB (APPC Transaction Instance Block) and YTIB (OTMA Transaction Instance Block) are mutually exclusive; only one is present at any given

time. For SHOW() purposes, TIB and YTIB are treated as synonyms. Specifying either TIB or YTIB displays, if available, whichever of the two blocks is present.

The SHOW(PROLOG) option displays information about the modules that are associated with the save areas in the chain. The module entry point is obtained from the saved register 15 value of the previous save area. If SHOW(SAVEAREA) is combined with SHOW(PROLOG), the display consists of a save area followed by the related prolog of the module that uses the save area.

The SHOW(SA) option is a synonym for the SHOW(SAVEAREA) option.

## NODE

Captures control block information for the node name specified in the *nodename* parameter. The *nodename* parameter specified must be alphanumeric, no longer than 8 characters, and identify a currently defined node name. Multiple *nodename* parameters can be specified with each parameter separated by a comma or a blank.

The control blocks that can be captured for node names are listed in Table 266. Primary control blocks for a node name, which are always present and available for capture, are identified by the column labeled "Primary". Optional control blocks for a node name, which might or might not be present and available for capture depending on workload and other factors, are identified by the column labeled "Optional".

Table 266. /DIAGNOSE SNAP NODE() control blocks

Name	Block description	Macro	Primary	Optional
EPF	Event Control Block Prefix	IEPF	X	
CULE	Common Use List Element Block	DFSCULE		X
DSPWRK1	Dispatcher Work Area	IDSPWRK		X
SAP	Save Area Prefix	ISAP		X
SAVEAREA	Save Area Set	REQUATE		X
PROLOG	Module Prolog Information	N/A		X
CLB	Communication Line Block	ICLI	X	
CTB	Communication Terminal Block	ICLI		X
CTT	Communication Translate Table	ICLI		X
CRB	Communications Restart Block	ICLI		X
SPQB	Subpool Queue Block	ICLI		X
SPQBEXT	Subpool Queue Extension Block	ICLI		X
EMHB	Expedited Message Handler Block	DBFEMHB		X
SMB	Scheduler Message Block	IAPS		X
TIB YTIB	APPC Transaction Instance Block OTMA Transaction Instance Block	DFSTIB DFSYTIB		X
CNT	Communication Name Table	ICLI		X
ECNT	Extended Communication Name Table	DBFECNT		X
CRTCN	Command CART & 4-Byte Console ID	DFSMCSC		X
CCB	Conversational Control Block	ICLI		X
CIB	Communication Interface Block	ICLI		X
MSGBP	Basic 01/03 Message Prefix	QLOGMSGP		X

Table 266. /DIAGNOSE SNAP NODE() control blocks (continued)

Name	Block description	Macro	Primary	Optional
UOWE	Unit of Work Table Entry	DFSUOWE		X
INBUF	Input Line Buffer	N/A		X
OUTBUF	Output Line Buffer	N/A		X
RAQE	VTAM Receive Any IO Buffer (In)	BUFVTPRE		X
RAQERES	VTAM Receive Any IO Buffer (Out)	BUFVTPRE		X

*SHOW keyword parameters for the /DIAGNOSE SNAP NODE command*

**SHOW**

Specifies which blocks are returned. If the SHOW keyword is omitted from the /DIAGNOSE SNAP NODE command, only the primary control blocks for the node name are captured by default. The SNAP NODE resource type supports the ALL, OPT, and PRI keywords, and all block names that are listed in Table 266 on page 502 as parameters for the SHOW keyword. Multiple SHOW parameters can be specified with each parameter separated by a comma or a blank. A maximum of 32 SHOW parameters can be specified.

The effects of the SHOW option on the data that is captured by the /DIAGNOSE SNAP NODE command are described in the following list:

**ALL**

Captures all control blocks listed in Table 266 on page 502 that are available. Example: /DIAGNOSE SNAP NODE(L3270A) SHOW(ALL).

**OPT**

Captures all optional control blocks that are available. Optional control blocks are identified by the Optional column in Table 266 on page 502. Example: /DIAGNOSE SNAP NODE(L3270A) SHOW(OPT).

**PRI**

Captures the primary control blocks for the node. Primary control blocks are identified by the Primary column in Table 266 on page 502. Example: /DIAGNOSE SNAP NODE(L3270A) SHOW(PRI).

**blockname**

Captures the specified block by block name. Block names are defined in the Name column in Table 266 on page 502. Example: /DIAGNOSE SNAP NODE(L3270A) SHOW(CLB).

**blockname,blockname**

Captures multiple blocks by block name. Example: /DIAGNOSE SNAP NODE(L3270A,L3270B) SHOW(CLB,CTB).

**keyword,blockname**

Captures a group of blocks by keyword and individual blocks by block name. Example: /DIAG SNAP LTERM(MTRL) SHOW(PRI,CTB).

*Usage notes for the /DIAGNOSE SNAP NODE command*

The /DIAGNOSE SNAP NODE command is available only in a DB/DC or DCCTL environment.

The following control blocks are available only in an IMS system where Fast Path is defined.

- ECNT (Extended Communication Name Table)
- EMHB (Expedited Message Handler Block)
- SMB (Scheduler Message Block)
- TIB (APPC Transaction Instance Block)
- YTIB (OTMA Transaction Instance Block)

If the SHOW(ALL) or SHOW(OPT) option is specified in a non-Fast Path environment, these blocks are ignored. If the SHOW(ECNT), SHOW(EMHB), SHOW(SMB), SHOW(TIB), or SHOW(YTIB) option is specified in a non-Fast Path environment, a DFS154I message is issued in response.

UOWE (Unit of Work Table Entry) block is available only in an IMS system where shared queues are defined. If the SHOW(ALL) or SHOW(APP) option is specified in a non-shared-queues environment, UOWE is ignored. If the SHOW(UOWE) option is specified in a non-shared-queues environment, a DFS154I message is issued in response.

TIB (APPC Transaction Instance Block) and YTIB (OTMA Transaction Instance Block) are mutually exclusive; only one is present at any given time. For SHOW() purposes, TIB and YTIB are treated as synonyms. Specifying either TIB or YTIB displays, if available, whichever of the two blocks is present.

The SHOW(PROLOG) option displays information about the modules that are associated with the save areas in the chain. The module entry point is obtained from the saved register 15 value of the previous save area. If SHOW(SAVEAREA) is combined with SHOW(PROLOG), the display consists of a save area followed by the related prolog of the module that uses the save area.

The SHOW(RECANY) option displays information about both RAQE (VTAM Receive Any IO Buffer (In)) and RAQERES (VTAM Receive Any IO Buffer (Out)) control blocks.

The SHOW(SA) option is a synonym for the SHOW(SAVEAREA) option.

## **OPTION**

Specifies the destination for the resource information captured by the SNAP function. The OPTION parameter is optional.

### **DISPLAY**

SNAP output is formatted and displayed on the issuing LTERM. DISPLAY is the default.

You can specify the following keywords with the DISPLAY option:

### **LIMIT**

Specifies a limit for the number of lines of formatted SNAP data to display in response to the command. LIMIT is a subparameter of the DISPLAY option. The LIMIT parameter is optional and has a default value of 1999. The *linecount* parameter must be numeric and in the range 1 - 9999.

### **FORMAT**

Specifies the format of the output to be produced. The default is FORMAT(LONG). Valid values for the FORMAT() parameter are:

### **LOC**

### **LOCATION**

Only the block name, description, and location are displayed.

### **LONG**

The complete display is produced, which includes the block name, description, location, and a dump of the complete storage area for the block in both hexadecimal and character format.

### **OLDS**

SNAP data is written to the OLDS.

### **SYSOUT**

SNAP data is queued to the diagnostic asynchronous work element (AWE) service and written to a SYSOUT data set.

You can specify the following keywords with the SYSOUT option:

### **CLASS**

Specifies an output class for the SYSOUT data set. The attributes of each output class are installation-specific and defined during JES initialization. The *sysout\_class* parameter must be a single alphanumeric character A - Z or 0 - 9. Any defined, valid JES output class can be specified. The characteristics of the SYSOUT data set are: DSORG=PS, RECFM=FB, LRECL=80 BLKSIZE=4080. The CLASS parameter is optional and defaults to the default SYSOUT class assigned to the IMS control region.

### **LIMIT**

Specifies a limit for the number of lines of formatted SNAP data to process.

The LIMIT parameter is optional and has a default value of 19999. If specified, the *linecount* parameter must be numeric and in the range 1 - 99999.

### **FORMAT**

Specifies the format of the output to be produced. The default is FORMAT(LONG). Valid values for the FORMAT() parameter are:

### **LOC**

### **LOCATION**

Only the block name, description, and location are displayed.

### **LONG**

The complete display is produced, which includes the block name, description, location, and a dump of the complete storage area for the block in both hexadecimal and character format.

### **TRACE**

If TRACE is specified, SNAP data is written to the trace data sets.

**Important:** If you do not issue the /TRACE SET ON TABLE DIAG OPTION LOG command before issuing the /DIAGNOSE command with the TRACE option, the /DIAGNOSE command automatically turns on the DIAG trace tables, writes the output to the trace tables, and then turns off the DIAG trace tables. One disadvantage of this method is that the output from only one /DIAGNOSE command can be written to the trace data sets; that is, each new command overwrites the data from the last command. To capture data from a series of /DIAGNOSE commands in a trace data set, issue the commands in this order:

1. /TRACE SET ON TABLE DIAG OPTION LOG command

2. /DIAGNOSE commands
3. /TRACE SET OFF TABLE DIAG command

**PGM** Captures control block information for the program specified in the *pgmname* parameter. The *pgmname* parameter specified must be alphanumeric, no longer than eight characters, and identify a currently defined program. Multiple *pgmname* parameters can be specified with each parameter separated by a comma or a blank.

The control blocks that can be captured for a program are listed in the following table. Primary control blocks for a logical line, which are always present and available for capture, are identified by the column labeled "Primary". Optional control blocks for a logical link, which might be present and available for capture based on workload and other factors, are identified by the column labeled "Optional".

Table 267. /DIAGNOSE SNAP PGM() control blocks

Name	Block description	Macro	Primary	Optional	Work	No Work
PDIR	Program Directory Block	DFSPDIR	X			X
RSCX	Resource Extension Block	DFSRSCX		X		X
INTLIST	Intent List	INTLIST		X		X
PSB	Program Specification Block	DFSPSB		X		X
PST	Partition Specification Table	IPST		X		X
CNT	Communication Name Table	ICLI		X		X
SMB	Scheduler Message Block	IAPS		X		X
DMBL	Data management Block List	DFSDMBL		X		X
XPCB	Program Communication Block Index Maintenance	DFSPCBS		X		X
PCB	Program Communication Block	DFSPCBS		X		X
PCBX	Program Communication Extension Block	DFSPCBS		X		X
EPCB	Program Communication Block Extension	DBFEPCB		X		X
MSG6P	Message Prefix (LU62)	DFS62PRE		X		X
MSGOP	Message Prefix (OTMA)	DFSYPRE				
PSBPRM	User Parameter List Block	IDLI		X		X
WKCDL	Data Capture Segment Work Area	N/A		X	X	
WKNDX	Index Maintenance Work Area	N/A		X	X	
WKXIO	Index I/O Work Area	N/A		X	X	
WKSEG	Segment Work Area	N/A		X	X	
WKIOA	I/O Work Area	N/A		X	X	
WKSSA	Segment Search Argument Work Area	N/A		X	X	
WKIFP	Fast Path Control Block Work Area	N/A		X	X	

**SHOW keyword parameters for the /DIAGNOSE SNAP PGM command**

**SHOW**

Specifies which blocks are returned. Using the SHOW() option, a single block, list of blocks, or group of blocks identified by the keyword can be returned. Example: /DIAG SNAP PGM(APOL1)

If the SHOW() option is omitted from the /DIAGNOSE SNAP PGM() command, only the primary control blocks for the program are captured by default. The SNAP PGM() resource type supports the ALL, OPT, PRI, WORK, and NOWORK keywords, and all block names that are listed in Table 267 on page 506 as parameters to the SHOW() option. Multiple SHOW() option parameters can be specified with each parameter separated by a comma or a blank. A maximum of 32 SHOW() option parameters can be specified.

**ALL**

Captures all control blocks listed in Table 267 on page 506 that are available. Example: /DIAG SNAP PGM(APOL1) SHOW(ALL)

**PRI**

Captures the primary control blocks for the program. Primary control blocks are identified by the Primary column in Table 267 on page 506. Example: /DIAG SNAP PGM(APOL1) SHOW(PRI)

**OPT**

Captures all optional control blocks that are available. Optional control blocks are identified by the Optional column in Table 267 on page 506. Example: /DIAG SNAP PGM(APOL1) SHOW(OPT)

***blockname***

Captures the specified block by block name. Block names are defined in the Name column in Table 267 on page 506. Example: /DIAG SNAP PGM(APOL1) SHOW(PDIR)

***blockname,blockname***

Captures multiple blocks by block name. Example: /DIAG SNAP PGM(APOL1) SHOW(PDIR,PSB)

***keyword,blockname***

Captures a group of blocks by keyword and individual blocks by block name. Example: /DIAG SNAP PGM(APOL1) SHOW(PRI,PDIR)

**WORK**

Captures all working storage blocks that are available. Working storage blocks are identified by the Work column in Table 267 on page 506. Example: /DIAG SNAP PGM(APOL1) SHOW(WORK)

**NOWORK**

Captures all non-work control blocks that are available. Non-work control blocks are identified by the No Work column in Table 267 on page 506. Example: /DIAG SNAP PGM(APOL1) SHOW(NOWORK)

***Usage notes for the /DIAGNOSE SNAP PGM command***

EPCB (Program Communication Block Extension) is available only in an IMS system where Fast Path is defined. If the SHOW(ALL) or SHOW(OPT) option is specified in a non-FP environment, the EPCB is ignored. If the SHOW(EPCB) option is specified in a non-FP environment, a DFS154I error message is issued in response

MSG6P (Message Prefix for LU62) and MSGOP (Message Prefix for OTMA) are mutually exclusive. Only one is present at any point. For SNAP PGM()

SHOW() purposes, the MSG6P and MSGOP are treated as synonyms. Specifying either MSG6P or MSGOP displays, if available, whichever of the two blocks is present.

## REGION

Captures control block information for the dependent region specified in the *region#* parameter. The *region#* parameter specified must be numeric, in the range 1 - 999, and identify a currently active dependent region. Multiple *region#* parameters can be specified with each parameter separated by a comma or a blank.

The dependent region might also be identified using the SNAP JOBNAME(*jobname*) format of the SNAP REGION() resource type. The *jobname* parameter specified must be alphanumeric, no longer than eight characters, and identify a currently active dependent region. Multiple *jobname* parameters can be specified with each parameter separated by a comma or a blank. The REGION(*region#*) and JOBNAME(*jobname*) formats can both be specified on the same command.

The control blocks that can be captured for a dependent region are listed in the following table. Primary control blocks for a dependent region, which are always present and available for capture, are identified by the column labeled "Primary". Optional control blocks for a dependent region, which might be present and available for capture based on workload and other factors, are identified by the column labeled "Optional".

Table 268. /DIAGNOSE SNAP REGION() control blocks

Name	Block description	Macro	Primary	Optional	System	Application
VTD	SVC Vector Table Directory Entry	DFSVDIR	X		X	
ASCB	MVS Address Space Control Block	IHAASCB		X	X	
ASSB	MVS Address Space Secondary Block	IHAASSB		X	X	
DPDIR	Dependent Region Directory Block	DFSDPDIR		X	X	
IWALE	Internal Work Area List Elements Block	DFSQALE		X	X	
LESEP	Local External Entry Table Prefix Block	DFSLESEP		X	X	
DRAT	DRA Thread Control Block	DFS DRAT		X	X	
IDT	Identify Table Entry	DFSIDT	X		X	
TCB	MVS Task Control Block (IDT)	IKJTCB		X	X	
TCB	Task Control Block (PST)	IKJTCB		X	X	
SAP	Save Area Prefix block	ISAP	X		X	
DSPWRK1	Dispatch Block: Work Area Part 1 (current)	IDSPWRK		X	X	
DSPWRK2	Dispatch Block: Work Area Part 2 (current)	IDSPWRK		X	X	
TCB	MVS Task Control Block (CDSP)	IKJTCB		X	X	
RB	MVS Associated Request Block (CDSP)	IHARB		X	X	
RBP	MVS Associated Request Block Prefix (CDSP)	IHARB		X	X	
XSB	MVS Extended Status Block (CDSP)	IHAXSB		X	X	

Table 268. /DIAGNOSE SNAP REGION() control blocks (continued)

Name	Block description	Macro	Primary	Optional	System	Application
DSPWRK1	Dispatcher Block: Work Area Part 1 (home)	IDSPWRK		X	X	
DSPWRK2	Dispatcher Block: Work Area Part 2 (home)	IDSPWRK		X	X	
DSPPST	PST Dispatching Control Block	IDSPWRK		X	X	
XMCI	Cross-Memory Control Block, ITASK Level	DFSXMC		X	X	
SSVPL	System Service Parameter List Block	DFSSSVPL		X	X	
DMIB	Directed Message Manager Interface Block	DFSDMIB		X	X	
CULE	Common Use List Element Block	DFSCULE		X	X	
CLLE	Common Latch List Element Block	DFSCLE		X	X	
LSMB	Logging Secondary Master Block	DFSLSMB		X	X	
SSIDX	Subsystem Status Index Entry	DFSSSIE	X		X	X
LCRE	Local Current Recovery Entry	DFSLCRE	X		X	X
RRE	Residual Recovery Element Block (LCRE)	DFSRRE		X		X
TIB	APPC Transaction Instance Block	DFSTIB		X		X
YTIB	OTMA Transaction Instance Block	DFSYTIB				
PCENTRY	Protected Conversation Task Table Entry	DFSRRSIB		X		X
RRE	Residual Recovery Element Block (PC)	DFSRRE		X		X
PST	Partition Specification Table	IPST	X		X	X
CNT	Communication Name Table	ICLI		X		X
SMB	Scheduler Message Block	IAPS		X		X
SQPST	Scheduler Queue Element	ISQPST		X		X
UOW	Unit of Work Value (QMGR)	DFSUOWE		X		X
UOWE	Unit of Work Table Entry	DFSUOWE		X		X
EPFXH	Extended Message Prefix Header	DFSEPFXH		X		X
MSGBP	Basic 01/03 Message Prefix	QLOGMSGP		X		X
BSGEP	Extended Message Prefix	QLOGMSGP		X		X
PDIR	Program Directory Block	DFSPDIR		X		X
PSB	Program Specification Block	DFSPSB		X		X
TLS	Transaction Level Statistics Area	DFSTLS		X		X
DACCT	DL/I Call Statistics Area	DFSDACCT		X		X
MSCEB	MSC Message Routing Exit Interface Block	DFSMSCEB		X		X
D2AFB	DB2 RRS Attach Facility Interface Block	DFSD2AF		X		X
OLRW	Online Reorganization Work Area Block	DFSOLRW		X		X
ISIT	Intersubsystem Interface Table	IPST		X		X

Table 268. /DIAGNOSE SNAP REGION() control blocks (continued)

Name	Block description	Macro	Primary	Optional	System	Application
DIRCA	Interregion Communication Area Block	IRC		X		X
EPST	Extended Partition Specification Table	DBFEPST		X		X
DSTA	DBCTL Statistics Area Block	DFSDSTA		X		X
RRE	Residual Recovery Element Block (PST)	DFSRRE		X		X
P62XT	LU 6.2 PST Extension Block	DFSP62XT		X		X
WLM	Workload Manager Work Area Block	DFSWLMWK		X		X
RSSMP	RAS System Security Message Prevention List	DFSRSSMP		X		X

**SHOW keyword parameters for the /DIAGNOSE SNAP REGION command**

**SHOW**

Specifies which blocks are returned. Using the SHOW() option, a single block, list of blocks, or group of blocks identified by the keyword can be returned. Example: /DIAG SNAP REG(1)

If the SHOW() option is omitted from the /DIAGNOSE SNAP REGION() command, only the primary control blocks for the dependent region are captured by default. The SNAP REGION() resource type supports the ALL, PRI, OPT, APP, and SYS keywords, and all block names listed in Table 268 on page 508 as parameters to the SHOW() option.

**ALL**

Captures all control blocks listed in Table 268 on page 508 that are available. Example: /DIAG SNAP REG(1) SHOW(ALL)

**PRI**

Captures the primary control blocks for the region. Primary control blocks are identified by the Primary column in Table 268 on page 508. Example: /DIAG SNAP REG(1) SHOW(PRI)

**OPT**

Captures all optional control blocks that are available. Optional control blocks are identified by the Optional column in Table 268 on page 508. Example: /DIAG SNAP REG(1) SHOW(OPT)

**APP**

Captures all application control blocks that are available. Control blocks that are part of the application group are identified by the Application column in Table 268 on page 508. Example: /DIAG SNAP REG(1) SHOW(APP)

**SYS**

Captures all system control blocks that are available. Control blocks that are part of the system group are identified in the System column in Table 268 on page 508. Example: /DIAG SNAP REG(1) SHOW(SYS)

**blockname**

Captures the specified block by block name. Block names are defined in the Name column in Table 268 on page 508. Example:  
/DIAG SNAP REG(1) SHOW(IDT)

**blockname,blockname**

Captures multiple blocks by block name. Example: /DIAG SNAP REG(1) SHOW(VTD,SAP)

**keyword,blockname**

Captures a group of blocks by keyword and individual blocks by block name. Example: /DIAG SNAP REG(1) SHOW(SYS,PDIR)

**Usage notes for the /DIAGNOSE SNAP REGION command**

DRAT (DRA Thread Control Block) is available only in a DBCTL environment. If the SHOW(ALL) or SHOW(SYS) option is specified in a non-DBCTL environment, the DRAT is ignored. If the SHOW(DRAT) option is specified in a non-DBCTL environment, a DFS110I message is issued.

DSTA (DBCTL Statistics Area Block) is available only in a DBCTL environment. If the SHOW(ALL) or SHOW(SYS) option is specified in a non-DBCTL environment, the DSTA is ignored. If the SHOW(DSTA) option is specified in a non-DBCTL environment, a DFS110I message is issued.

EPST (Extended Partition Specification Table) is available only in an IMS system where Fast Path is defined. If the SHOW(ALL) or SHOW(APP) option is specified in a non-Fast Path environment, the EPST is ignored. If the SHOW(EPST) option is specified in a non-Fast Path environment, a DFS154I message is issued.

TIB (APPC Transaction Instance Block) and YTIB (OTMA Transaction Instance Block) are mutually exclusive. Only one is present at any time. For SNAP REGION() SHOW () purposes, TIB and YTIB are treated as synonyms. Specifying either TIB or YTIB displays whichever of the two blocks is present.

The SNAP JOBNAME(*jobname*) format of the SNAP REGION() resource type cannot be used to identify a CCTL thread. All CCTL threads have the same job name and the CICS region, and therefore it is impossible to identify the correct thread by the job name.

If multiple regions are started with the same job name, only the first region will be found using the SNAP JOBNAME(*jobname*) format of the SNAP REGION() resource type.

VTD, ASCB, ASSB, DPDIR, IWALE, LESEP, DRAT, and IDT are not available for an OLR region. If the SHOW(ALL), SHOW(PRI), or SHOW(SYS) option is specified for an OLR region, these blocks are ignored. If any of these blocks are explicitly specified with the SHOW option for an OLR region, a DFS2859I message is issued.

**RM** Specifies the scope of the resource search for certain resource types. The RM() parameter is optional and applicable only to the LTERM(), NODE(), and USER() resource types.

Valid values for the RM() parameter are:

**YES** If YES is specified or the RM() parameter is omitted, a local search is performed for the requested resource. If a local copy of the resource is found, that copy is used. If a local copy of the resource

is not found, and a Resource Manager is available, a second search is performed on the RM for the resource. If a global copy of the resource is found, that copy is used.

**NO** If NO is specified, only a local search for the resource is performed.

**ONLY** If ONLY is specified, only a global search on the Resource Manager is performed. If RM(ONLY) is specified and a Resource Manager is not available, a DFS2859I message is issued with reason text RM(ONLY) INVALID, RM UNAVAILABLE, and the command is not processed.

### SHOW

Specifies which control blocks are captured by the SNAP function. The SHOW parameter is optional. Valid filtering values for the *keyword* and *blockname* parameters are listed in each section that describes a SNAP resource type. SHOW(PRI) is the default.

### STRUCTURE

Captures information about the DFSSQS control block storage for the specified shared queues structure. STRUCTURE(ALL) is the default.

**TRAN** Captures information about the transaction specified in the *tranname* parameter. This keyword captures the SMB data for a specified transaction.

**USER** Captures control block information about the user name specified in the *username* parameter. The *username* parameter specified must be alphanumeric, no longer than 8 characters, and identify a currently defined user name. Multiple *username* parameters can be specified with each parameter separated by a comma or a blank.

The control blocks that can be captured for user names are listed in the following table. Primary control blocks for a user name, which are always present and available for capture, are identified by the column labeled "Primary". Optional control blocks for a user name, which might or might not be present and available for capture depending on workload and other factors, are identified by the column labeled "Optional".

Table 269. /DIAGNOSE SNAP USER() control blocks

Name	Block description	Macro	Primary	Optional
SPQB	Subpool Queue Block	ICLI	X	
EPF	Event Control Block Prefix	IEPF		X
CULE	Common Use List Element Block	DFSCULE		X
DSPWRK1	Dispatcher Work Area	IDSPWRK		X
SAP	Save Area Prefix	ISAP		X
SAVEAREA	Save Area Set	REQUATE		X
PROLOG	Module Prolog Information	N/A		X
CLB	Communication Line Block	ICLI		X
CTB	Communication Terminal Block	ICLI		X
CTT	Communication Translate Table	ICLI		X
CRB	Communications Restart Block	ICLI		X
SPQBEXT	Subpool Queue Extension Block	ICLI		X
EMHB	Expedited Message Handler Block	DBFEMHB		X

Table 269. /DIAGNOSE SNAP USER() control blocks (continued)

Name	Block description	Macro	Primary	Optional
SMB	Scheduler Message Block	IAPS		X
TIB	APPC Transaction Instance Block	DFSTIB		X
YTIB	OTMA Transaction Instance Block	DFSYTIB		X
CNT	Communication Name Table	ICLI		X
ECNT	Extended Communication Name Table	DBFECNT		X
CRTCN	Command CART & 4-Byte Console ID	DFSMCSC		X
CCB	Conversational Control Block	ICLI		X
CIB	Communication Interface Block	ICLI		X
MSGBP	Basic 01/03 Message Prefix	QLOGMSGP		X
UOWE	Unit of Work Table Entry	DFSUOWE		X
INBUF	Input Line Buffer	N/A		X
OUTBUF	Output Line Buffer	N/A		X
RAQE	VTAM Receive Any IO Buffer (In)	BUFVTPRE		X
RAQERES	VTAM Receive Any IO Buffer (Out)	BUFVTPRE		X

**SHOW keyword parameters for the /DIAGNOSE SNAP USER command**

**SHOW**

Specifies which blocks are returned. If the SHOW keyword is omitted from the /DIAGNOSE SNAP USER command, only the primary control blocks for the user name are captured by default. The SNAP USER() resource type supports the ALL, OPT, and PRI keywords, and all block names that are listed in Table 269 on page 512 as parameters for the SHOW keyword. Multiple SHOW parameters can be specified with each parameter separated by a comma or a blank. A maximum of 32 SHOW parameters can be specified.

The effects of the SHOW option on the data that is captured by the /DIAGNOSE SNAP USER command are described in the following list:

**ALL**

Captures all control blocks listed in Table 269 on page 512 that are available. Example: /DIAGNOSE SNAP USER(IMSUS04) SHOW(ALL).

**OPT**

Captures all optional control blocks that are available. Optional control blocks are identified by the Optional column in Table 269 on page 512. Example: /DIAGNOSE SNAP USER(IMSUS04) SHOW(OPT).

**PRI**

Captures the primary control blocks for the user. Primary control blocks are identified by the Primary column in Table 269 on page 512. Example: /DIAG SNAP USER(IMSUS04) SHOW(PRI).

**blockname**

Captures the specified block by block name. Block names are defined in the Name column in Table 269 on page 512. Example: /DIAGNOSE SNAP USER(IMSUS04) SHOW(SPQB).

***blockname,blockname***

Captures multiple blocks by block name. Example: /DIAGNOSE SNAP USER(IMSUS04) SHOW(SPQB,EPF).

***keyword,blockname***

Captures a group of blocks by keyword and individual blocks by block name. Example: /DIAG SNAP USER(IMSUS04) SHOW(PRI,EPF).

***Usage notes for the /DIAGNOSE SNAP USER command***

The /DIAGNOSE SNAP USER command is available only in a DB/DC or DCCTL environment.

The following control blocks are available only in an IMS system where Fast Path is defined.

- ECNT (Extended Communication Name Table)
- EMHB (Expedited Message Handler Block)
- SMB (Scheduler Message Block)
- TIB (APPC Transaction Instance Block)
- YTIB (OTMA Transaction Instance Block)

If the SHOW(ALL) or SHOW(OPT) option is specified in a non-Fast Path environment, these blocks are ignored. If the SHOW(ECNT), SHOW(EMHB), SHOW(SMB), SHOW(TIB), or SHOW(YTIB) option is specified in a non-Fast Path environment, a DFS154I message is issued in response.

UOWE (Unit of Work Table Entry) block is available only in an IMS system where shared queues are defined. If the SHOW(ALL) or SHOW(APP) option is specified in a non-shared-queues environment, UOWE is ignored. If the SHOW(UOWE) option is specified in a non-shared-queues environment, a DFS154I message is issued in response.

TIB (APPC Transaction Instance Block) and YTIB (OTMA Transaction Instance Block) are mutually exclusive; only one is present at any given time. For SHOW() purposes, TIB and YTIB are treated as synonyms. Specifying either TIB or YTIB displays, if available, whichever of the two blocks is present.

The SHOW(PROLOG) option displays information about the modules that are associated with the save areas in the chain. The module entry point is obtained from the saved register 15 value of the previous save area. If SHOW(SAVEAREA) is combined with SHOW(PROLOG), the display consists of a save area followed by the related prolog of the module that uses the save area.

The SHOW(RECANY) option displays information about both RAQE (VTAM Receive Any IO Buffer (In)) and RAQERES (VTAM Receive Any IO Buffer (Out)) control blocks.

The SHOW(SA) option is a synonym for the SHOW(SAVEAREA) option.

**Usage notes**

One of the key uses for the /DIAGNOSE SNAP MODULE(*modname*) command is to return the entry point address of a module. With the /DIAGNOSE SNAP MODULE(*modname*) command, users can determine the entry point address of the target module without having to create a console dump to set an MVS SLIP trap.

Another important use for the `/DIAGNOSE SNAP MODULE(modname)` command is to extract the prolog information for a module. The prolog information for a module contains information about the current maintenance level of the module on your system. This can be useful to find out if there is any missing maintenance on your module. Establishing a common, or baseline, maintenance level can be useful in diagnosing difficult or complex problems.

## Examples

### *Example 1 for /DIAGNOSE command*

Entry ET:

```
/DIAGNOSE SNAP TRAN(tranname)
```

Response ET:

```
DFS058I DIAGNOSE COMMAND COMPLETED
```

**Explanation:** Diagnostic information about transaction *tranname* is captured and written to the IMS log.

### *Example 2 for /DIAGNOSE command*

Entry ET:

```
/DIAGNOSE SNAP BLOCK(ALL) OPTION(TRACE)
```

Response ET:

```
DFS058I DIAGNOSE COMMAND COMPLETED
```

**Explanation:** Diagnostic information about IMS control blocks is captured and written to the trace data sets.

### *Example 3 for /DIAGNOSE command*

Entry ET:

```
/DIAGNOSE SNAP BLOCK(CSCD)OPTION(OLDS)
```

Response ET:

```
DFS058I DIAGNOSE COMMAND COMPLETED
```

**Explanation:** Diagnostic information about APPC/OTMA SMQ SCD Extension control block is captured and written to the IMS log.

### *Example 4 for /DIAGNOSE command*

Entry ET:

```
/DIAGNOSE SNAP AREA(00010001)
```

Response ET:

```
/DIAGNOSE SNAP STORAGE DISPLAY
```

```
Resource: AREA(D0010001)
```

```
ALDS      DEDB Area Name List Entry      Loc: 09E0A590
-----
0000 C4F0F0F1 F0F0F0F1 09BF7E90 09B26A30 |D0010001..=.....|
```

DMAC	DEDB Area Control Block				Loc: 09B26A30
0000	F1F2F1F8	C4C5C4C2	D1F0F0F1	C4F0F0F1	1218DEDBJ001D001
0010	F0F0F0F1	0010213F	2217081F	00000000	0001.....
0020	00000000	000001EF	00008000	00000E5B	.....\$
0030	00000000	00000000	00000000	00000000	.....
0040	00000000	00000000	00000000	00000000	.....
0050	00000000	00000000	00000000	00000000	.....
0060	00000000	00000000	00000403	00000000	.....
0070	01C1000A	000F0019	00780002	00000200	.A.....
0080	000001F6	000003EA	00000005	00000000	..6.....
0090	00000000	00018000	00000000	01000101	.....
00A0	06088001	00000001	00000000	00000002	.....
00B0	00000000	00000000	40000000	00000000	.....
00C0	00000000	00000000	00000000	09B267A0	.....
00D0	00000000	00000000	00000000	40404040	.....
00E0	001CB600	000001C7	00000000	00000000	.....G.....
00F0	00000000	00080600	00000000	0A12F618	......6.
0100	09A5B0FC	00000000	00010000	00000000	.v.....
0110	00000000	00000000	00000000	00000000	.....
0120	09A5B0FC	00000100	00401800	00000000	.v.....
0130	00000000	00000000	00000000	00000000	.....
0140	00000000	00000000	00000000	00000000	.....
0150	00000000	00000000	00000000	00000000	.....
0160	00000000	00000000	00000000	00000000	.....
0170	00000000	00000000	00000000	00000000	.....
0180	00000000	0000FFA6	000001EF	00000000	.....w.....
0190	00000000	00000000	00000000	00000000	.....
01A0	00000000	00000000	00000000	00000000	.....
01B0	00000000	00000000	00000000	00000000	.....
01C0	00000000	00000000	00000000	00000000	.....
01D0	00000000	00000000	00000000	00000000	.....
01E0	00000000	00000000	00000000	00000000	.....
01F0	00000000	00000000	00000000	00000000	.....
0200	00000000	00000000	00000000	00000000	.....
0210	00000000	00000000	00000000	00000000	.....
0220	00000000	00000000	00000000	00000000	.....
0230	00000000	00000000	00000000	00000000	.....
0240	00000000	00000000	00000000	00000000	.....
0250	00000000	00000000	00000000	00000000	.....
0260	00000000	00000000	00000000	00000000	.....
0270	00000000	00000000	00000000	00000000	.....
0280	00000000	00000000	00000000	00000000	.....
0290	00000000	00000000	00000000	25262D2E	.....
02A0	34142B20	D5D6E5FF	00161718	00000000	...NOV.....
02B0	00010000	09A5C090	00000000	00000000	...v.....
02D0	00000000	00000000	00000000	00000000	.....
02E0	00000000	00000000	00000000	00000000	.....
02F0	00000000	00000000	00000000	00000000	.....
0300	00000000	00000000	00000000	00000000	.....
0310	00000000	00000000	00000000	00000000	.....
0320	00000000	00000000	00000000	00000000	.....
0330	00000000	00000000	00000000	00000000	.....
0340	00000000	00000000	00000000	00000000	.....
0350	00000000	00000000	00000000	00000000	.....
0360	09A5B0FC	00000000	00000000	00000000	.v.....
0370	00000000	00000000	00000000	00000000	.....
0380	00000000	00000000	00000000	00000000	.....
0390	00000000	00000000	00000000	00000000	.....
03A0	00000000	00000000	00000000	00000000	.....
03B0	00000000	00000000	00000000	00000000	.....
03C0	00000000	40404040	40404040	40404040	....
03D0	40404040	40404040	40404040	40404040	.....
03E0	40404040	00000000	00000000	00000000	.....
03F0	00000000	00000000	00000000	00000000	.....
0400	00000000	00000000	00000000	00000000	.....
0410	00000000	00000000	00000000	00000000	.....

```

0420 00000000 00000000 00000000 00000000 | .....
0430 00000000 00000000 00000000 00000000 | .....
0440 00000000 00000000 00000000 00000000 | .....
0450 2010213F 22193984 2034028D 00000000 | .....d.....
0460 00000000 00000000 00000000 00000000 | .....
0470 00000000 00000008 00000000 00000000 | .....
0480 00000000 00000000 C65DAC50 F91B8BE0 | .....F).&9...
0490 00000000 00000000 00000000 00000000 | .....
04A0 00000000 00000000 00000000 00000000 | .....
04B0 00000000 00000000 00000000 00000000 | .....
04C0 00000000 00000000 09B0D000 00000000 | .....
04D0 00000000 00000000 00000000 00000000 | .....
04E0 00000000 00000000 00000000 00000000 | .....
04F0 00000000 00000000 00000000 00000000 | .....
0500 00000000 00000000 | .....
*10213/152244*

```

**Explanation:** Diagnostic information about the primary control blocks for the area is captured, formatted, and displayed on the issuing LTERM.

*Example 5 for /DIAGNOSE command*

Entry ET:

```
/DIAGNOSE SNAP AREA(D0010001) SHOW(ALDS)
```

Response ET:

```
/DIAGNOSE SNAP STORAGE DISPLAY
```

```
Resource: AREA(D0010001)
```

```

ALDS      DEDB Area Name List Entry      Loc: 09E0A590
-----
0000 C4F0F0F1 F0F0F0F1 09BF7E90 09B26A30 |D0010001..=.....|
*10213/152259*

```

**Explanation:** Diagnostic information about the ALDS control block for the area is captured, formatted, and displayed on the issuing LTERM.

*Example 6 for /DIAGNOSE command*

Entry ET:

```
/DIAGNOSE SNAP AREA(D0010001,D0010002) SHOW(ALDS)
```

Response ET:

```
/DIAGNOSE SNAP STORAGE DISPLAY
```

```
Resource: AREA(D0010001)
```

```

ALDS      DEDB Area Name List Entry      Loc: 09E0A590
-----
0000 C4F0F0F1 F0F0F0F1 09BF7E90 09B26A30 |D0010001..=.....|

```

```
Resource: AREA(D0010002)
```

```

ALDS      DEDB Area Name List Entry      Loc: 09E0A5A0
-----
0000 C4F0F0F1 F0F0F0F2 09BF7E90 09B26F38 |D0010002..=...?.|
*10213/152311*

```

**Explanation:** Diagnostic information about the ALDS control block for areas D0010001 and D0010002 is captured, formatted, and displayed on the issuing LTERM.

*Example 7 for /DIAGNOSE command*

Entry ET:

```
/DIAGNOSE SNAP AREA(D0010001) SHOW(ALDS,ADSC)
```

Response ET:

```
/DIAGNOSE SNAP STORAGE DISPLAY
```

```
Resource: AREA(D0010001)
```

```
ALDS      DEDB Area Name List Entry      Loc: 09E0A590
-----
0000 C4F0F0F1 F0F0F0F1 09BF7E90 09B26A30 |D0010001..=.....|

ADSC      DEDB Area Data Set Control Block Loc: 09A5B0FC
-----
0000 C4F0F0F1 F0F0F0F1 C9D4E2E3 C5E2E3D3 |D0010001IMSTESTL|
0010 4BC4F0F0 F1F0F0F0 F1404040 40404040 |.D0010001|
0020 40404040 40404040 40404040 40404040 |
0030 40404040 00008003 08000000 10000000 |.....|
0040 09B26A30 09A5B040 00000000 F9000000 |.....v. ....9...|
0050 00000000 00000000 00000000 01CF1380 |.....|
0060 00000000 00000000 00000200 00000200 |.....|
0070 00000000 00000000 00000000 00000000 |.....|
0080 00000000 00000000 00000000 00000000 |.....|
0090 00000000 00000000 00000000 00000000 |.....|
00A0 00000000 00000000 00000000 00000000 |.....|
00B0 00000000 00000000 00000000 |.....|
*10213/152558*
```

**Explanation:** Diagnostic information about the ALDS and ADSC control blocks for the area is captured, formatted, and displayed on the issuing LTERM.

*Example 8 for /DIAGNOSE command*

Entry ET:

```
/DIAGNOSE SNAP MSNAME(LINK31V6) SHOW(PRI)
```

Response ET:

```
/DIAGNOSE SNAP STORAGE DISPLAY
```

```
Resource: MSNAME(LINK31V6)
```

```
LNB      Link Name Block      Loc: 0C67C800
-----
0000 00000000 00000000 00000000 00000000 |.....|
0010 00000000 00820084 00000000 D3C9D5D2 |.....b.d....LINK|
0020 F3F1E5F6 00100021 00053D30 00000000 |31V6.....|
0030 00000000 00000000 00000000 00000000 |.....|
0040 0C67C878 00000000 0004F150 00000000 |..H.....1&....|
0050 013F01B7 00000000 00000000 00000000 |.....|
0060 00000000 00000000 00000000 00000000 |.....|
*12216/072150*
```

**Explanation:** Diagnostic information about the primary control blocks for MSNAME LINK31V6 is captured, formatted, and displayed on the issuing LTERM.

**Related concepts:**

 /DIAGNOSE command SNAP function (Diagnosis)



---

## Chapter 26. /DISPLAY commands

Use the IMS /DISPLAY commands to display the processing status of IMS message queues and processing resources.

The /DISPLAY commands can provide helpful information for controlling IMS operations.

These commands can be issued to an IMSplex using the Batch SPOC utility.

The output from a /DISPLAY command consists of headings that describe the display, the information requested, and a time stamp showing Julian date and time, in one of the following two formats:

*yyddd/hhmmss*  
*yyyyddd/hhmmss*

You control the time stamp format by specifying the YEAR4= execution parameter.

The value shown in the QCNT column of the command output has different meanings for different commands, as described in individual /DISPLAY command topics.

If the command includes a generic parameter that does not match any existing resource, an error message results.

In an IMSplex, when status is copied to RM at signoff/logoff time, the status (and control blocks if the resource is dynamic) is deleted from the local system. An Automated Operator Interface (AOI) application program that issues a /DISPLAY command will not see some status information that is kept in RM if the IMS system on which the AOI program runs is not using RM services. When a /DISPLAY command is issued for NODE, USER, and LTERM resources from the command master, global status from RM might be displayed. The output of the display command will be dependent on the following guidelines:

### **Command master IMS system**

- Displays information from the local system
- Displays information from RM that is not owned, or is owned by a failed system

### **Other IMS systems**

Displays information from the local system

**Attention:** Using ALL, generic parameters, the /DISPLAY STATUS command, and the /DISPLAY CONVERSATION command without a specific resource name will result in extensive accesses to the Resource Manager for global information, and their use should be carefully considered.

Subsections:

- “Status and attributes for the /DISPLAY command” on page 523
- “/DISPLAY ACT command” on page 537
- “/DISPLAY AFFIN command” on page 554
- “/DISPLAY AOITKN command” on page 556

- “/DISPLAY APPC command” on page 557
- “/DISPLAY AREA command” on page 560
- “/DISPLAY ASMT command” on page 565
- “/DISPLAY CCTL command” on page 575
- “/DISPLAY CONV command” on page 578
- “/DISPLAY CPLOG command” on page 582
- “/DISPLAY CQS command” on page 583
- “/DISPLAY DB command” on page 583
- “/DISPLAY DBD command” on page 594
- “/DISPLAY DESC command” on page 595
- “/DISPLAY FDR command” on page 596
- “/DISPLAY FPV command” on page 597
- “/DISPLAY HSB command” on page 599
- “/DISPLAY HSSP command” on page 602
- “/DISPLAY LINE command” on page 603
- “/DISPLAY LINK command” on page 607
- “/DISPLAY LTERM command” on page 611
- “/DISPLAY LUNAME command” on page 614
- “/DISPLAY MASTER command” on page 620
- “/DISPLAY MODIFY command” on page 621
- “/DISPLAY MSNAME command” on page 635
- “/DISPLAY NODE command” on page 637
- “/DISPLAY OASN SUBSYS command” on page 648
- “/DISPLAY OLDS command” on page 649
- “/DISPLAY OTMA command” on page 652
- “/DISPLAY OVERFLOWQ command” on page 658
- “/DISPLAY PGM command” on page 661
- “/DISPLAY POOL command” on page 663
- “/DISPLAY PSB command” on page 682
- “/DISPLAY PTERM command” on page 684
- “/DISPLAY Q command” on page 686
- “/DISPLAY QCNT command” on page 689
- “/DISPLAY RECOVERY command” on page 694
- “/DISPLAY RTCODE command” on page 700
- “/DISPLAY SHUTDOWN STATUS command” on page 701
- “/DISPLAY STATUS command” on page 707
- “/DISPLAY STRUCTURE command” on page 714
- “/DISPLAY SUBSYS command” on page 716
- “/DISPLAY SYSID TRANSACTION command” on page 719
- “/DISPLAY TIMEOVER command” on page 720
- “/DISPLAY TMEMBER command” on page 721
- “/DISPLAY TRACE command” on page 734
- “/DISPLAY TRACKING STATUS command” on page 744
- “/DISPLAY TRAN command” on page 752
- “/DISPLAY UOR command” on page 757

- “/DISPLAY USER command” on page 760

**Related concepts:**

 [Displaying components and resources \(System Administration\)](#)

---

## Status and attributes for the /DISPLAY command

A list of the attributes and status shown in the /DISPLAY command are described in the following table.

*Table 270. Attributes and status in the /DISPLAY command*

Status or attribute	Meaning
	Indicates no status for the resource.
ACCEPT TRAFFIC	OTMA is ready to accept messages from this member.
ACT MODETBL	Indicates the mode table name actually used to initiate mode.
ACTIV	Indicates that this node is in an XRF session, on the active system.
ACTIVE, A	Indicates one of the following: <ul style="list-style-type: none"> <li>• Conversation in progress.</li> <li>• For /DISPLAY MODIFY, the routing code (RTCODE) named is active or that the library is active.</li> <li>• The IMS subsystem is the active subsystem.</li> <li>• The current XRF surveillance mechanism is active.</li> <li>• The OTMA member is in an active z/OS cross-system coupling facility (XCF) group.</li> <li>• The secondary master terminal is active.</li> </ul>
ACTIVE-DBCMD	Indicates an /DBD or /DBR command is in progress and waiting for the region to terminate before the /DBD or /DBR can complete.
ACTIVE-RRS	The APPC/OTMA shared queues enablement is active, and z/OS Resource Recovery Services (RRS) is used for communication between front-end and back-end systems for all synchronization levels.
ACTIVE-RRS/XCF	The APPC/OTMA shared queues enablement is active, and z/OS cross-system coupling facility (XCF) is used for communication between front-end and back-end systems if synchronization level is NONE or CONFIRM. For synchronization level SYNCPOINT, RRS is used for communication between front-end and back-end systems.
ACTIVE-XCF	The APPC/OTMA shared queues enablement is active, and XCF is used for communication between front-end and back-end if synchronization level is set to NONE or CONFIRM. Transactions of synchronization level SYNCPOINT are queued with affinity to the front-end system.
ACTV	Indicates, for a logical link, that link startup processing is complete and the line is available for message transfer.
ALLOC	Indicates that an ISC user is allocated. The node name to which the user is allocated or signed on follows in parentheses.
ALLOCATION IN PROGRESS	Indicates that a conversation is in the process of being allocated.

Table 270. Attributes and status in the /DISPLAY command (continued)

Status or attribute	Meaning
ALLOCF	Indicates that database allocation failed.
ALLOCS	Indicates that database allocation was successful.
AREA	The subset of a DEDB.
AUTOSR	Indicates a session initiation option of ASR.
AVAILABLE	Indicates one of the following: <ul style="list-style-type: none"> <li>• The OLDS can be reused.</li> <li>• A region is available to schedule an application.</li> <li>• A coupling facility structure is available for use by IMS.</li> </ul>
AWAITING RESPONSE	Indicates that an active terminal is waiting for a response.
AWAITING SNAPQ	Indicates that the alternate subsystem is waiting for /CHE SNAPQ from the active subsystem.
BACKOUT	Indicates that the OLDS is potentially required for backout or, in the case of a database, there are incomplete backouts that prevent the use of the database.
BACKUP	Indicates that the IMS subsystem is the alternate subsystem.
BAL	Identifies a transaction eligible for load balancing (for example, with parallel limits specified). The current maximum number of regions that can be simultaneously scheduled to process the transaction follows in parentheses.
BCKUP	Indicates that this node is in an XRF session, on the alternate system.
C1INOP, C2INOP, C3INOP, C4INOP	Indicates the inoperable node or terminal component, where C1, C2, C3, and C4 refer to different components as defined by system definition. (See the /COMPT and /RCOMPT commands for details on how to ready inoperable components.)
CANCEL	Indicates that an APPC connection was stopped by the /STOP APPC CANCEL command.
CHECK	Indicates that the OTMA client is using RACF for security verification.
CLSER	Indicates that an error occurred while closing the OLDS.
CLSDST	Indicates that a /CLSDST or /STOP command has been entered for a VTAM node but has not yet taken effect. This condition can occur because of an outstanding input or output operation that has not completed. The VTAM network operator can issue a z/OS VARY command to clear the condition.
COLD	For an MSC logical link, indicates that link startup processing is not complete.  For a SLUP or FINANCE terminal, indicates that the next session initiation is cold (message sequence numbers are initialized to 0.)
COMPINOP	Indicates an inoperable terminal component. For 3270 terminals, this is an indication that either the display or the printer is not ready.
CON	Indicates that a node is connected, or in session, with IMS.
CONN	Subsystem connection.

Table 270. Attributes and status in the /DISPLAY command (continued)

Status or attribute	Meaning
CONN, ACTIVE	Indicates that an application program has established communication with a subsystem.
CONNECTED	Indicates that IMS is connected to a coupling facility structure.
CONN IN PROGRESS	Subsystem connection in progress.
CONVACT, CONV-ACT	Indicates an active conversation on this terminal, node, or user.
CONVERSATION TERM/USER	Indicates for /DISPLAY MODIFY that the transaction to be changed or deleted or the transaction that references a program or database to be changed or deleted is in conversation. The terminal originating the conversational transaction and the conversation ID are also displayed.  The terminal displayed is: <ul style="list-style-type: none"> <li>• <b>nodename</b> for VTAM terminals</li> <li>• <b>nodename</b> and <b>username</b> if an ETO user is signed on to the node</li> <li>• <b>username</b> for ETO users who signed off while in conversation</li> <li>• <b>lin#-pte#</b> (line and pterm number) for non-VTAM terminals</li> <li>• <b>1- SC</b> for the system console</li> </ul>
CONVHLD, CONV-HLD	Indicates a held conversation on this terminal, node, or user.
COPY-PHASE	Indicates that the CREATE utility is currently in COPY-PHASE for an ADS.
CREATED	Indicates the XCF group has been created, but the OTMA member has not yet connected to it.
DB-STOPD	Indicates that a database used by this program is stopped.
/DBD ACTIVE	A /DBDUMP command is in progress for a database to be changed or deleted.
/DBR ACTIVE	A /DBRECOVERY command is in progress for a database to be changed or deleted.
DEACT (DEACTIVATED)	Indicates a physical terminal/node or line has been permanently deactivated. Restart of non-VTAM node requires /STOP DC and /START DC commands. Message DFS2473 in the system console log might contain information regarding the reason this status was set. DFS2473 can occur more than once in the system console log.
DEADQ	Indicates a user with dead letter queues or whose last access time was outside the limit set by the DLQT JCL parameter. The DEADQ status can be removed by signing on the user or entering the /DEQUEUE or /ASSIGN command.
DEALLOCATION IN PROGRESS	Indicates that a conversation is in the process of being deallocated.
DEQCT	Number of local message dequeues since the last IMS cold start or, in a shared-queue environment, since the last IMS restart. DEQCT is reset to zero when ENQCT reaches 32768. A DEQCT count can equal the sum of the DEQCTs of multiple destinations (logical terminals or transactions).

Table 270. Attributes and status in the /DISPLAY command (continued)

Status or attribute	Meaning
DISABLED	Indicates APPC/IMS is not known to APPC/z/OS.
DISCONNECTED	Indicates one of the following: <ul style="list-style-type: none"> <li>• A node is not in session.</li> <li>• The OTMA client has disconnected from the XCF group.</li> <li>• IMS is not connected to a coupling facility structure.</li> </ul>
DQF	Indicates a dequeue request for OTMA REPresynch command failed.
DYNAMIC	A transaction was built in a shared-queues environment to enable transaction messages to be enqueued, but the transaction cannot run on the current IMS subsystem.
EEQE	Extended error queue element, indicates that one or more error queue elements are associated with the database.
ELIGIBLE FOR SCHEDULING	Indicates that the transaction is eligible for scheduling and cannot be deleted by online change. Stop the transaction before attempting another online change commit.
ENABLED	Indicates that APPC is started.
ENQCT	Number of local message enqueues since the last IMS cold start, or in a shared-queue environment, since the last IMS restart. ENQCT is reset to the number of messages on queue (QCT) when the counter reaches 32768 (QCT = ENQCT - DEQCT). The ENQCT can exceed 32768 if the DEQCT remains zero. The ENQCT will be reset to zero if the counter exceeds 65535. An ENQCT count can equal the sum of the ENQCTs of multiple destinations (logical terminals or transactions).
ERE	Indicates, for a logical link, that link startup processing is not complete. When the link is started, emergency restart synchronization will be performed, because the previous link shutdown was either not normal or an IMS emergency restart was performed.
EXCL	Indicates a node, terminal, or user is in exclusive mode.
FAILED	Indicates one of the following: <ul style="list-style-type: none"> <li>• APPC failed to start.</li> <li>• OTMA failed to join the XCF group or the XCF group failed initialization. Check the DFSPBxxx PROCLIB member for the OTMA group name and member name definition. You might have defined an incorrect or duplicate XCF member name for IMS OTMA.</li> </ul>
FILLING	Indicates that a log gap is currently being filled.
FLOOD	Indicates that input from the tmember is suppressed because of a message flood condition. <p>Check if the flood was caused by stopped IMS programs or hung regions. Sometimes it could be caused by bad I/O response time.</p>
FORCE, FORCES	Indicates a session initiation option of FORCE.

Table 270. Attributes and status in the /DISPLAY command (continued)

Status or attribute	Meaning
FORCE-RRS	The APPC/OTMA shared queues enablement is active, and RRS is used for communication between front-end and back-end systems for all synchronization levels. All transactions are queued without affinity even if one IMS system has the APPC/OTMA shared queues enablement not active.
FORCE-RRS/XCF	The APPC/OTMA shared queues enablement is active, and XCF is used for communication between front-end and back-end systems if synchronization level is NONE or CONFIRM. For synchronization level SYNCPOINT, RRS is used for communication between front-end and back-end systems. All transactions are queued without affinity even if one IMS system has the APPC/OTMA shared queues enablement not active.
FORMAT-PHASE	Indicates that the CREATE utility is currently in FORMAT-PHASE for an ADS.
FULL	Indicates that the OTMA client is using RACF for security verification, including for dependent regions.
GLOBAL QUEUE COUNT INTERNAL ERROR, GLOBAL QUEUE COUNT STORAGE ERROR	Indicates that IMS is unable to access global queue counts or obtain storage to query the queue counts.
IC	Indicates that image copy is in progress for that area or database by an HSSP region.
IDENTIFICATION IN PROGRESS	Indicates that the IMS logger is in the process of identifying to the Transport Manager Subsystem (TMS) but has not yet completed.
IDENTIFIED	Indicates that an IMS logger has identified to the TMS.
IDLE	Indicates that no activity of any kind is in progress for a line, node, or logical link. This is a common condition for VTAM node channel-to-channel links, and processor storage-to-processor storage links.
IDLE-Cxx	When the Multiple Systems Coupling (MSC) TCP/IP link is in IDLE state, asynchronous output might still be in progress and the link is waiting for that response. xx represents the following values:  <b>X'10'</b> A response is owed for data. <b>X'16'</b> A response is owed for stop bracket initiation (SBI) or bracket initiation stopped (BIS).
INACTIVE	The APPC/OTMA shared queues enablement is inactive. If AOS=Y is specified, the global status might change to active if the IMS member that does not support the APPC and OTMA shared queues enablement leaves the shared queues group. A back-end system will still be able to process messages from a front-end system even with status INACTIVE.
INACTIVE, I	Indicates that the current XRF surveillance mechanism is inactive. For the /DISPLAY MODIFY command, indicates that the library is inactive. The RSR tracking subsystem is idle.
INOP	Indicates that a terminal or node is inoperable.

Table 270. Attributes and status in the /DISPLAY command (continued)

Status or attribute	Meaning
IN-OVERFLOW	Indicates that a coupling facility structure is in overflow mode.
INPUT IN PROGRESS	Indicates input in progress for an active terminal.
INQONLY	Indicates a /DBDUMP command was issued for the database.
IN SLOWDOWN	Indicates that the OTMA member is experiencing a slowdown because of excessive message traffic or resource constraints.  Check the XCF configuration in your environment.
INUSE, IN USE	Indicates that queuing is in process for this transaction. (either terminal input or program-to-program switch). Also indicates this is the OLDS currently being used.
INVALID SUBSYSTEM NAME = XXXX	Indicates that a subsystem name was not defined to IMS.
I/O PREVEN, I/O PREVENT, I/O PREV	Indicates a BMP program that contains GSAM cannot complete scheduling because I/O prevention has not completed. Indicates further I/O requests to data sets are inhibited.
I/O TOLERATION	Takeover process by which an alternate IMS subsystem ensures database integrity and enables new transaction processing as soon as possible.
JOIN	Indicates that the OTMA client is using RACF only to authorize the client connection at the time of the client bid request. No transaction, command, or dependent region security checking is performed for individual messages.
LOCK	Indicates a node, terminal, transaction, program, or database is locked.
LOOPTEST	Indicates a line or terminal in looptest mode.
LOST	Indicates that the VTAM LOSTERM EXIT has been scheduled for this node but has not yet been recognized by IMS. At the next interrupt for this node, IMS will interrogate the LOSTERM value.  All values, with one exception, result in an immediate CLSDST, or disconnection, from IMS. For the LOSTERM exception, IMS must wait for VTAM to notify IMS (by another LOSTERM) of completion of recovery operation. Indicates an IMS logger's connection to the TMS is gone as a result of TMS or VTAM failure.
MAX TPIPE	The total number of tpipes reaches the maximum number.  To resolve this flood condition for TPIPE limit, issue the IMS /CHECKPOINT command three times to clear the unused OTMA TPIPE, or restart IMS. Also, examine your TPIPE limit in the OTMA member descriptor in the DFSYDTx PROCLIB member.
MESSAGES WAITING	Indicates that there are system messages waiting to be sent, which prevents shutdown from completing.
MFSTEST, MFST	Indicates a terminal, node, or user in MFSTEST mode.

Table 270. Attributes and status in the /DISPLAY command (continued)

Status or attribute	Meaning						
MSG IN QUEUE	Indicates that there are messages in the queue for an active terminal.						
MSG CT	Number of messages on the queue for this destination (calculated by subtracting the DEQCT from ENQCT).						
N/A	Indicates that a link is not assigned or a user is signed on to a static terminal, in which case queues are not applicable.						
NEEDED	Indicates the OLDS that needs to be archived.						
NO ACTIVE CONVERSATION	Indicates that there is no active conversation between a logger and a log router.						
NODE-USR	Node name and user identifier.						
NOIN	Indicates a line or terminal stopped for input.						
NO INPUTTING LINES	Indicates no terminal activity.						
NO LOG ROUTER SG DEFINED	Indicates that there is no Log Router Service Group defined to DBRC.						
NONE	Indicates that the OTMA client is not using RACF for security verification.						
NOOUT	Indicates a line or terminal stopped for output.						
NO OUTPUTTING LINES	Indicates no terminal activity.						
NOQUEUE	Indicates a line or terminal that has had message queuing stopped for message switching.						
NOT CONN	No subsystem connection.						
NOT DEFINED	Indicates the OTMA member has left the XCF group (for example, because of a /STOP OTMA command).  Restart the OTMA client to join the XCF group.						
NOT IDENTIFIED	Indicates that an IMS logger is not identified to the TMS.						
NOTIDLE s	Indicates that a logical link is waiting for the completion of a synchronous event. The s represents the following subcodes:  <table border="0"> <tr> <td style="padding-right: 10px;"><b>A</b></td> <td>Indicates a status of NOTIDLE-POST, which means that an event has completed but the link has not been dispatched to process it.</td> </tr> <tr> <td><b>B</b></td> <td>Indicates a status of NOTIDLE-IWAIT, which means the link is waiting for completion of internal I/O.</td> </tr> <tr> <td><b>Cxx</b></td> <td>Indicates a status of NOTIDLE-TP WAIT, which means the link is waiting for completion of a TP access method request. The two characters xx indicate the value of the access method operation code. Blanks are displayed if the link is VTAM.  For MSC TCP/IP links, xx indicates the following:  <b>X'20'</b> A response is owed for a restart request.</td> </tr> </table>	<b>A</b>	Indicates a status of NOTIDLE-POST, which means that an event has completed but the link has not been dispatched to process it.	<b>B</b>	Indicates a status of NOTIDLE-IWAIT, which means the link is waiting for completion of internal I/O.	<b>Cxx</b>	Indicates a status of NOTIDLE-TP WAIT, which means the link is waiting for completion of a TP access method request. The two characters xx indicate the value of the access method operation code. Blanks are displayed if the link is VTAM.  For MSC TCP/IP links, xx indicates the following:  <b>X'20'</b> A response is owed for a restart request.
<b>A</b>	Indicates a status of NOTIDLE-POST, which means that an event has completed but the link has not been dispatched to process it.						
<b>B</b>	Indicates a status of NOTIDLE-IWAIT, which means the link is waiting for completion of internal I/O.						
<b>Cxx</b>	Indicates a status of NOTIDLE-TP WAIT, which means the link is waiting for completion of a TP access method request. The two characters xx indicate the value of the access method operation code. Blanks are displayed if the link is VTAM.  For MSC TCP/IP links, xx indicates the following:  <b>X'20'</b> A response is owed for a restart request.						
NOTINIT	Indicates a database or program for which directory initialization failed or indicates DMB or PSB pool space not large enough to accommodate blocks. Indicates a TRAN that could not be initialized.						

Table 270. Attributes and status in the /DISPLAY command (continued)

Status or attribute	Meaning
NOTOPEN, NOT-OPEN	Indicates a database, area, line, terminal, or physical link that is not in open status. This status is not applicable to MSDBs.
NOT USABLE	Indicates a log data set (OLDS) is unusable because of previous error.
NRE	Indicates, for a logical link, that link startup processing is not complete. When the link is started, normal restart synchronization will be performed, because the previous link shutdown or IMS restart was normal.
OFR	Indicates that the database or area is being brought up to the current tracking level with online forward recovery.
OLR	Indicates that the database has online reorganization in progress
OPEN	Indicates that the VTAM ACB is open. Indicates an area to be changed or deleted is open.
OPNDST	Indicates that an OPNDST is in process for this node.
OUTBOUND	Indicates that the alternate subsystem is APPC-enabled. Also shown on active subsystem when APPC processing cannot start.
OUTPUT IN PROGRESS	Indicates output in progress for an active terminal.
PAGE	Indicates an MFS paged message.
PERMNT ERR	Indicates that there is a permanent error, such as a read error, that prevents a log gap from being filled.
PLANNED TAKEOVER IN PROGRESS	Indicates all active subsystems being tracked have successfully shut down and the tracker is now in the process of shutting down.
PRE-OPEN FAILED	Pre-open for ADS failed.
PREV CLSER	Indicates that the previous OLDS could not be closed because of an I/O error; therefore, this OLDS is required by the Log Recovery utility to correct the condition.
PRI	Indicates that this node is the primary partner of an ISC session; or, for a link, indicates that it is the primary partner of an MSC-VTAM session.
PRIMARY MSTR	Indicates that the terminal is the primary master terminal.
PROFILE	Indicates that the OTMA client is using values in the Security Data section of the message prefix for a transaction.
PRST	Indicates that a terminal or user is in preset destination mode. The destination trancode or logical terminal name follows in parentheses.
PSB SCHEDULED	A program that references a database to be changed or deleted is scheduled.
PSTOPPED (PSTO, PSTOP)	Indicates a line, terminal, transaction, or logical link that has been pstopped. This status can indicate that a different, necessary resource is unavailable. For example, a transaction could show a status of PSTOP because the associated program is not initialized.
PUR, PURGING	Indicates a transaction, line, or terminal that is purging all of its output.

Table 270. Attributes and status in the /DISPLAY command (continued)

Status or attribute	Meaning
QCT	Number of messages on the queue for this destination (calculated by subtracting the DEQCT from ENQCT).
QERR, QERROR	Indicates that an I/O error occurred on the queue for this LTERM, or remote transaction (MSC).
QLOCK	Indicates that the LTERM is locked from sending any further output or from receiving input which could create additional output for the same LTERM until the state is reset by a specific request received on the session.
QSTP	Indicates that transaction queuing is stopped by online change because the transaction is affected by the online change. Online change might be changing or deleting the transaction, or changing or deleting a program, PSB, database, or DMB referenced by the transaction. Transaction queuing is stopped until the online change is committed or aborted.
QUEUING <i>mm</i>	Indicates that messages are queued to the transaction to be changed or deleted, and <i>mm</i> is the number of messages queued.
QUI	Indicates that a VTAM node has sent a VTAM Quiesce-End-of-Chain indicator to suspend IMS output.
QUIESCED	Indicates one of the following: <ul style="list-style-type: none"> <li>• The OTMA member is temporarily quiesced. Restart the member or OTMA.</li> <li>• The node is stopped.</li> <li>• The database is currently quiesced by a previous UPDATE DB START(QUIESCE) or UPDATE AREA START(QUIESCE) command.</li> </ul>
QUIESCING	Indicates that the database is undergoing quiesce by a previous UPDATE DB START(QUIESCE) or UPDATE AREA START(QUIESCE) command.
REBLD-INPROG	Indicates that a CQS structure rebuild is in progress for a structure.
RECALL	Indicates that database or area is in recall.
RECD	The number of messages that are received from the specified node.  For static terminals, this is the number of messages that are received since the last IMS cold start, warm start, or emergency restart. The count is not reset when a static terminal logs off or logs on.  For dynamic terminals, this is the number of messages that are received since the current user signed on to the dynamic terminal. The count is reset when a user signs off from the dynamic terminal, and following a cold start, warm start, or emergency restart.
RECOVERY	Requests the display of recovery values that pertain to the node or user. For the /DISPLAY DB command, indicates that DRF recovery is in progress for that database.
RECOVERY-NEEDED, RECOVERN	Indicates areas that need recovery.

Table 270. Attributes and status in the /DISPLAY command (continued)

Status or attribute	Meaning
RELREQ (NODE)	Indicates that the VTAM RELREQ exit routine has been driven but IMS is waiting for an operation in progress to complete before releasing the node.
REP	Indicates that IMS is waiting for an OTMA REPresynch command from the client.
REQ	Indicates that IMS is sending an OTMA REQresynch command to the client.
RESP	Indicates that the node, line, terminal, or user is in input response mode and the response reply message is available for output or in the process of being sent.
RESPINP, RESP-INP	Indicates the terminal, line, node, or user is in full-function input response mode and the response mode input is still in-doubt; for example, the response reply message is not available for output.
RESP-INP-FP	Indicates the terminal, line, node, or user is in Fast-Path input response mode and the response mode input is still in-doubt; for example, the response reply message is not available for output.
RESYNC	Indicates that the positive acknowledgment for an IMS recoverable output message was not received when the connection with the VTAM node was terminated. This message will be subject to resynchronization when the next connection for this node is attempted.
RNL	Randomizer is not loaded for a DEDB database.
RSF	Indicates that a reset request failed for an OTMA REPresynch command.
SCHEDULED	Indicates that a conversation, transaction, or program has been received and queued. For a conversation, this status will be displayed from the time an input message is entered until the output message is dequeued. With /DISPLAY MODIFY, indicates that the named resource (a transaction or program to be changed or deleted, or a program referencing a database to be changed or deleted) is scheduled. With /DISPLAY OLDS, indicates that an archive job had been generated.
SEC	Indicates that this node is the secondary partner of an ISC session; or, for a link, indicates that it is the secondary partner of an MSC-VTAM session.
SECOND MSTR	Indicates that the terminal is the secondary master terminal.
SENDING LOGS	Indicates that a logger has an active conversation with a log router and is sending logs to the tracking site.

Table 270. Attributes and status in the /DISPLAY command (continued)

Status or attribute	Meaning
SENT	<p>The number of messages that are sent from the specified node.</p> <p>For static terminals, this is the number of messages sent since the last IMS cold start, warm start, or emergency restart. The count is not reset when a static terminal logs off or logs on.</p> <p>For dynamic terminals, this is the number of messages sent since the current user signed on to the dynamic terminal. The count is reset when a user signs off from the dynamic terminal, and following a cold start, warm start, or emergency restart.</p>
SERVER	Indicates that the OTMA client is acting as the server.
SEVER+FLOOD	<p>Indicates that the global flood limit was specified in client descriptor or in the /START TMEMBER ALL INPUT command, and IMS has reached this OTMA global message flood limit.</p> <p>To resolve the OTMA global flood condition, examine each OTMA client and identify the pending message counts for each client. A client with too many unprocessed messages might have triggered the flood. Examine also any stopped programs and hung regions.</p>
SEVERE-ERROR	Indicates that ADS encountered a severe error.
SHUT	Indicates that normal processing has completed for the node and a VTAM shutdown-complete indicator was returned to IMS. The node can receive IMS output but cannot enter data while in this state.
SHUTDOWN-STRCHKPT	Indicates that CQS will take a structure checkpoint during normal shutdown.
SIGN	Indicates that a terminal or user is signed on to a node under enhanced security. The user signed on to the node is shown in parentheses following SIGN.
SIMLOGON	Indicates that a logon to IMS has been simulated.
SMQ BACKEND	Indicates that this OTMA member has been created at the shared queues back-end IMS to process transaction messages. However, this member has no XCF connection to this IMS.
SPND	Indicates that a transaction had messages on the suspend queue.
/STA ACTIVE	A /START DB command is in progress for a database to be changed or deleted.
STARTING	Shown after /DIS CONV is issued and before the status is set to SCHEDULED. Indicates that a conversation has been received but is not eligible for scheduling to an application program until an end-of-message indication is received. Also shown after /START APPC is issued and before the status is set to ENABLED or FAILED.
STATIC	Indicates that the node, LTERM, or user was defined during system definition.

Table 270. Attributes and status in the /DISPLAY command (continued)

Status or attribute	Meaning
STOP IN PROGRESS	Indicates a /STOP SUBSYS command is in progress.
STOPPED, STO, STOP	Indicates an area, line, LTERM, LU name, node, terminal, user, OLDS, subsystem, transaction, routing code (RTCODE), OTMA client, program, or database that is stopped. Indicates that a subsystem connection is stopped. Indicates a /STOP SERVGRP command was issued. Indicates that a transaction pipe is stopped.
STO-INPUT	Indicates one of the following: <ul style="list-style-type: none"> <li>The terminal is stopped with input messages enqueued to the LU name.</li> <li>The OTMA member has been stopped by the IMS command /STOP TMEMBER <i>member-name</i>. No new input from this member will be accepted.</li> </ul>
STO-OUTP	Indicates that the terminal is stopped with output message enqueued to the LU name.
STRCHKPT-INPROG	Indicates that a CQS structure checkpoint is in progress for a structure.
SUBSYSTEM XXXX NOT DEFINED BUT RECOVERY OUTSTANDING	Indicates that the subsystem was not defined but IMS has outstanding recovery elements.
SUSPENDED	Indicates that the transaction has been suspended. For /DISPLAY MODIFY, indicates that the transaction to be changed or deleted is on the suspend queue.
SUSPENDED LOGS	Indicates that a logger has an active conversation with a log router but has suspended sending logs to the tracking site because of resource shortage. The conversation is still intact.
SYNCHRONIZING, SYN	Indicates that the alternate subsystem is processing a /CHE SNAPQ command from the active subsystem. Indicates that a transaction pipe is being synchronized.
SYNC P2P	This status is displayed for the OTMA DFSYICAL member, which is created internally to process synchronous program switch requests.
SYNC P2P+FLOOD	OTMA is experiencing a message flood condition for synchronous program switch.  To resolve this message flood for synchronous program switch, the flood limit can be increased by issuing the /START TMEMBER DFSYICAL INPUT <i>higher_flood_limit</i> command.
SYS CONSOLE	Indicates that the terminal is the system console.
TAKEOVER IN PROGRESS	Indicates that the alternate subsystem is taking over workload from the active subsystem.
TAKEOVER REQUESTED	Indicates that the active subsystem has requested a takeover by the alternate subsystem.
TBR	Indicates that IMS is waiting for an OTMA TBResynch command from the client.
TERM IN PROGRESS	Indicates an internal termination of the subsystem.
TERMINATING	Indicates that the application program is being terminated.
TEST	Indicates a line, node, terminal, or user in test mode.

Table 270. Attributes and status in the /DISPLAY command (continued)

Status or attribute	Meaning
TKOTRA	Indicates that a node, line, link, or terminal in an XRF session is to be traced only during takeover, to help diagnose XRF terminal switch problems.
TMP	Indicates that a transaction pipe is temporary.
TRA	Indicates that the physical terminal, node, logical link, LU name, transaction, program, or transaction pipe is being traced.
TRACKING	Indicates that the XRF alternate subsystem is tracking the active subsystem or an RSR tracking subsystem is tracking one or more active subsystems.
UNAVAILABLE, UNAVAIL	Indicates an ADS that is unavailable. Indicates that a region is unavailable because an application is using it, even though the application is not currently scheduled. The region is not available to any other application for use. Indicates that there are no records of a log gap at the active site; these records may have been deleted. Indicates that a coupling facility structure is not available for use by IMS.
UNKNOWN	Indicates that the XCF member has an unknown status.  Restart OTMA or the OTMA client to join the XCF group.
UNPLANNED TAKEOVER IN PROGRESS	Indicates an /RTAKEOVER command was entered on the tracking subsystem and an unplanned takeover is in progress.
USTOP, USTOPPED	Indicates that scheduling of transactions has been stopped because of unavailable data.
UNSUPPORTED	The APPC/OTMA shared queues enablement is not supported for this member. This status applies only to the local status. The IMS system has been started with AOS=N or AOS=F and RRS=N. This IMS system will never be able to support the APPC/OTMA shared queues enablement.
VIR	Indicates that the DEDB area has the VSO option.

Table 270. Attributes and status in the /DISPLAY command (continued)

Status or attribute	Meaning
WAIT <i>cond</i>	<p data-bbox="768 254 1433 338">Indicates that the application program is waiting for the completion of an event. The <i>cond</i> represents the reason for waiting. The <i>cond</i> can be:</p> <p data-bbox="768 359 1433 443"><b>AOI</b> indicates that an AO application issued a GMSG call with the wait option specified, but there are no messages for the application to retrieve.</p> <p data-bbox="768 464 1433 600"><b>BLOCKMOVER or BLKMVR</b> indicates that an application control block cannot be loaded because the ACB block mover is busy. The application might be waiting in the block mover because of an IRLM failure.</p> <p data-bbox="768 621 1433 705"><b>CMD/PENDING</b> indicates that a /DBD command or a /DBR command is in progress.</p> <p data-bbox="768 726 1433 779"><b>INPUT</b> indicates that the application program is in wait-for-input (WFI) mode.</p> <p data-bbox="768 800 1433 1020"><b>INTENT or INTENT/POOL</b> indicates one of two conditions: <ol data-bbox="862 863 1433 1020" style="list-style-type: none"> <li data-bbox="862 863 1433 947">1. The application program's intent for a database conflicts with the use of the database by a scheduled program.</li> <li data-bbox="862 957 1433 1020">2. A temporary shortage of DMB, PSB, or PSB work area pool space exists.</li> </ol> </p> <p data-bbox="768 1041 1433 1146"><b>I/O PREVEN</b> indicates that a BMP region which accesses a GSAM database cannot schedule until I/O prevention has completed.</p> <p data-bbox="768 1167 1433 1283"><b>MESSAGE</b> indicates that the application program is in a pseudo wait-for-input mode. The application is scheduled and is waiting for a message.</p> <p data-bbox="768 1304 1433 1377"><b>POOLSPACE or POOLSP</b> indicates a temporary shortage of DMB, PSB, or PSB work area pool space exists.</p> <p data-bbox="768 1398 1433 1482"><b>SWITCHOVER or SWITCH</b> indicates that the alternate system is tracking the active system.</p> <p data-bbox="768 1503 1433 1587"><b>SYNCPOINT</b> indicates that the application in the region is now in sync point.</p> <p data-bbox="768 1608 1433 1692"><b>WAIT-EPCB POOL</b> indicates that there is a temporary shortage of EPCB pool space.</p> <p data-bbox="768 1713 1433 1797"><b>WAIT-RRS/OTMA PC</b> A program has a protected conversation with an OTMA client.</p>
WAIT BID	<p data-bbox="768 1808 1433 1871">Indicates that the OTMA client is processing a Client-Bid request.</p> <p data-bbox="768 1892 1433 1923">Start the OTMA client to join the XCF group.</p>

Table 270. Attributes and status in the /DISPLAY command (continued)

Status or attribute	Meaning
WAITING	Indicates that the MPP region is waiting for work. Indicates a log gap is not being filled yet.
WAIT RESPONSE	Indicates that the OTMA member is processing a message and is waiting for a response from the server.  This status code might indicate a potential IMS OTMA error. Contact IBM Software Support.
WRERR	Indicates a write I/O error on the OLDS or a database.

**Related tasks:**

 Setting DEADQ status time with the DLQT parameter (Communications and Connections)

## /DISPLAY ACT command

Use the /DISPLAY ACT command to display region and DC information associated with an IMS system. The region is scheduled to an application program and the IMS resources are assigned.

Subsections:

- “Environment”
- “Syntax”
- “Keywords”
- “Usage notes” on page 538
- “Output fields” on page 538
- “Examples” on page 546

### Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

Table 271. Valid environments for the /DISPLAY ACT command and keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
/DISPLAY	X	X	X
ACT	X	X	X
DC	X		X
REGION	X	X	X

### Syntax



### Keywords

The following keywords are valid for the /DISPLAY ACT command:

- DC** Displays only the DC subset of the output. The DC portion of the display can include:
- The status of active VTAM ACBs. If MNPS is used for XRF, then both the APPLID ACB and the MNPS ACB are displayed. If MNPS is not active, only the APPLID ACB displays.
  - User variable (USERVAR) name if using an XRF. If MNPS is active, USERVAR is invalid and is not displayed. Instead, the MNPS ACB name is displayed.
  - VTAM generic resource name (GRSNAME) and its current status.
  - The APPC IMS LU name, the APPC connection status, the APPC timeout value, and the APPC/IMS flood control limit for active APPC conversations.
  - The APPC/OTMA SHARED QUEUE status, the number of RRS TCBs attached and the number of AWEs (work-elements) queued to the RRS TCBs.
  - OTMA status.
  - The TCP/IP generic resources GENIMSID value and status.
  - The number of active LINES receiving input or sending output.
  - The number of active NODES receiving input or sending output.
  - The number of active LINKS (half sessions) receiving input or sending output.

**REGION**

Displays only the REGION subset of the output. The display consists of active regions.

**Usage notes**

When you issue the /DISPLAY ACT command from a specific environment, such as DCCTL, only the information that is valid for that environment is displayed.

ACTIVE-XCF, ACTIVE-RRS, and ACTIVE-RRS/XCF have similar values for local and global status.

**FORCE-RRS/XCF**

The global status is ACTIVE-XCF because FORCE is a local status.

**ACTIVE-RRS**

Both the local and global status is changed to INACTIVE if RRR becomes unavailable.

**ACTIVE-RRS/XCF**

Both the local and global status is changed to ACTIVE-XCF if RRS becomes unavailable.

IMS creates a task to process the IMS catalog updates for a database definition change through the DDL or another process. The task is started when the INIT DFSDMU is issued by the application program changing the IMS catalog. The task is then shown among the other IMS regions with the /DISPLAY ACT command.

**Output fields**

If /DISPLAY ACT command is issued, the output for both DC and REGION is displayed.

The following output fields are displayed for the /DISPLAY ACT DC command, and also for the /DISPLAY ACT command (no DC or REGION keyword specified):

#### **OTMA GROUP**

Status of the IMS Open Transaction Manager Access (OTMA) group. The status can be either ACT or NOTACTIVE.

Only one OTMA group can be active at one time.

#### **VTAM ACB**

The status of the VTAM ACB, which can be OPEN, CLOSE PENDING, or CLOSED. The status of LOGONS can be ENABLED or DISABLED. The base IMS LU name and the current APPC connection status are shown if IMS is running with an APPC/MVS-capable operating system.

The following are displayed for the VTAM ACB:

##### **APPLID=**

The application identification name for IMS. If MNPS is used with XRF, then it is the APPLID ACB.

##### **USERVAR=**

The user name for IMS. Only applicable for XRF-capable or RSR-capable systems.

##### **GRSNAME=**

The VTAM generic resource name. Displayed as blanks if no generic resource name exists.

**Restriction:** IMS displays either the USERVAR or the GRSNAME. For non-XRF or non-RSR subsystems (no USERVAR specified), IMS displays the GRSNAME, which can be blank.

If the IMS system is XRF, the following fields are appended to the date/time stamp:

##### **RSEname**

Recoverable service element name

##### **System**

ACT or BACKUP

##### **MNPS=**

The MNPS ACB name if MNPS is used for XRF.

##### **QUEUED RRSWKS**

The number of RRS requests (work elements) currently queued to the RRS TCBS. The work elements might be queued to the RRS TCBS for work from the APPC or OTMA shared message queue function.

##### **STATUS**

Status can be one of the following:

##### **AWAITING SNAPQ**

Alternate system awaiting SNAPQ checkpoint from active

##### **SYNCHRONIZING**

Alternate system processing SNAPQ checkpoint

##### **TRACKING**

Alternate system is in process of tracking active system

**TAKEOVER REQUESTED**

Alternate system requests a takeover

**TAKEOVER IN PROGRESS**

Alternate system in process of taking over workload from active

**I/O TOLERATION**

New active system in I/O toleration phase of processing prior to becoming a true active system

**TCPIP\_GENIMSID**

The generic IMS ID that is used to participate in an MSC TCP/IP generic resource group. The value shown is the value specified on the GENIMSID parameter of the IMS DFSDCxxx member of the IMS.PROCLIB data set.

TCP/IP generic resources enable remote MSC systems to connect to an IMSplex without having to specify a specific IMS system.

If TCPIP\_GENIMSID displays a value, then the following STATUS field displays ACTIVE. If no value is shown, then the STATUS field displays DISABLED.

**STATUS**

Displays ACTIVE or DISABLED depending on whether a value is displayed in the TCPIP\_GENIMSID field.

The following output fields are displayed for the /DISPLAY ACT REGION command, and also for the /DISPLAY ACT command (no DC or REGION keyword specified):

**END TIME**

The time at which IMS will end this synchronous program switch request.

**REGID**

Region identifier. For DBCTL, the thread number is displayed.

**JOBNAME**

The name of the job processing in the region. If no regions of that type exist, the job name is set to MSGRGN, FPRGN, BATCHRGN, or DBTRGN, depending on the region type. DBTRGN is only displayed for systems that support DBT threads.

DBTRGN, BATCHRGN, or FPRGN is displayed for DBCTL.

**TYPE** Type of application program processing in the region, or type of region. BMH, BMP, DBRC, DBT, DLS, and FPU are displayed for DBCTL.

The following region/program types can be displayed:

**BMH** HSSP processing

**BMP** Batch message processing

**BMPE** Batch message processing for an external subsystem thread

**DBRC** DBRC address space

**DBT** DBCTL thread

**DLS** DL/I address space

**FP** Fast Path

**FPE** Fast Path processing for an external subsystem

**FPM** Fast Path message-driven program

**FPME** Fast Path message-driven program for an external subsystem thread

**FPU** Fast Path utility program

**FPUE** Fast Path utility program for an external subsystem thread

**TP** Teleprocessing

**TPE** Message-driven program for an external subsystem thread

**TPI** Transaction Program Instance. A CPI Communications driven transaction is running in the region

**TRAN** The name of the target transaction in the request data of a DL/I ICAL request that initiated a synchronous program switch.

**TRAN/STEP**

Transaction code being processed by region, or NONE, if there are no regions of that type.

**PROGRAM**

Name of the program processing in the region.

DFSCPIC is displayed for CPI Communications driven transaction programs that have not issued a DL/I APSB call to allocate a PSB.

For DBCTL, SB name is displayed.

**STATUS**

Status of the region, which can be one of the following:

**ACTIVE-RRS**

The z/OS Resource Recovery Services (RRS) enablement is active. The synchronous transaction inputs from APPC/OTMA with sync levels of NONE, CONFIRM, and SYNCPT with the RRS indicator are queued onto the shared queue. This status is equivalent to the existing status of ACTIVE and is for a setting combination of AOS=Y and RRS=Y.

**ACTIVE-XCF**

The z/OS cross-system coupling facility (XCF) enablement is active. The synchronous transaction inputs from APPC/OTMA with sync levels of NONE and CONFIRM with the XCF indicator are queued onto the shared queue. This status is for either AOS=X or a setting combination of AOS=B and RRS=N.

**ACTIVE-RRS/XCF**

The RRS enablement and the XCF enablement are both active. The synchronous transaction inputs from APPC/OTMA with sync level of SYNCPT with the RRS indicator are queued onto the shared queue. The synchronous transaction inputs with sync level of NONE and CONFIRM with the XCF indicator are queued onto the shared queue. This status is for a setting combination of AOS=B and RRS=Y.

**ACTIVE-DBCMD**

A /DBD, /DBR, UPD DB STOP(ACCESS), or UPDATE DB STOP(UPDATES) command is in progress and waiting for the region to terminate before the command can complete.

**AVAILABLE**

The active threads are available. The region is available to schedule an application.

**FORCE-RRS**

The synchronous transaction inputs with sync levels of NONE, CONFIRM, and SYNCPT with the RRS indicator are queued onto the shared queue (without any affinity). This status is equivalent to the existing status of FORCE where AOS=F is specified.

**FORCE-RRS/XCF**

The synchronous transaction inputs with sync level of SYNCPT with the RRS indicator are queued onto the shared queue (without any affinity).

The XCF enablement is also active. The synchronous transaction inputs with sync levels of NONE and CONFIRM with the XCF indicator are queued onto the shared queue.

This status is equivalent to the combination of the existing status of FORCE where AOS=F is specified and the new status ACTIVE-XCF.

The status is for a setting combination of AOS=S. It is equivalent to the combination of the existing status of FORCE where AOS=F is specified and the new status ACTIVE-XCF.

**OLR-RUNNING**

The region is an IMS system-created BMP region to process the HALDB online reorganization (HALDB OLR) for the partition name. For those regions that have OLR running, the region type is BMP, the job name is the DBD name, the transaction name is the partition name, and the program name is the partition name with a suffix of 0.

**OLR-ALTER-INPROG**

The region is an IMS system-created BMP region to process the OLR-ALTER for the HALDB database. For those regions that have OLR-ALTER in progress, the region type is BMP, the job name is the DBD name, the transaction name is the partition name, and the program name is the partition name with a suffix of 0.

**SCHEDULING**

The application program is being scheduled.

**TERMINATING**

A dependent region or application program is being terminated.

**TERM-ABDUMP IN-P**

A /STOP REGION ABDUMP command has been issued, an SRB has been scheduled for the region, and abnormal termination is in progress.

**TERM-ABDUMP PEND**

A /STOP REGION ABDUMP command has been issued, but an abnormal termination SRB has not been scheduled because the region is in an ineligible state.

**TERM-ABDUMP SCHD**

A /STOP REGION ABDUMP command has been issued and an abnormal termination SRB has been scheduled for the region.

**TERM-BCKOUT IN-P**

Abnormal termination is in progress and backout processing has been initiated.

**TERM-THREAD TERM**

Region termination is in progress. For abnormal termination, either a /STOP REGION ABDUMP command was issued or the region encountered an error and terminated independently. For normal termination, either a /STOP REGION command was issued or the application program completed and terminated independently.

**TERM-WAIT SYNCPT**

Region termination is in progress and the application in the region is in the sync point.

When a dependent region is found in this state, a continuation line is inserted into the display, which shows the recovery unit ID (URID) and the originating IMS system ID (ORIGIN) that are associated with the transaction processing in the dependent region.

**TERM-WAIT RRS**

Region termination is in progress and the application in the region is in the sync point. This status is displayed when an application has a protected conversation with an OTMA client that is processing a sync point. The sync point can continue after the OTMA client issues either an SRRCMIT or SRRBACK call. This status is also displayed when the application program is part of a cascaded family and is processing a sync point. RRS cascaded transaction support is used by APPC/OTMA SMQ enablement to synchronize the back-end and front-end systems.

When a dependent region is found in this state, a continuation line is inserted into the display, which shows the recovery unit ID (URID) and the originating IMS system ID (ORIGIN) that are associated with the transaction processing in the dependent region.

**UNAVAILABLE**

An active DBT thread is unavailable. An application is using the region, even though the application is not currently scheduled. This region is therefore not available to any other application.

**WAIT-AOI**

An AO application issued a GMSG call with the WAITAOI subfunction specified, but there are no messages for the AO application to retrieve.

**WAIT-BLOCKMOVER**

An application control block cannot be loaded because the ACB block mover is busy. The application might be waiting in the block mover because of an IRLM failure.

**WAIT-CALLOUT**

The application program in the region is waiting for a response to a synchronous callout or synchronous program switch message.

When a dependent region is found in this state and is waiting for a synchronous callout message, a continuation line is inserted into the display, which shows the target member (TMEM) and the transaction pipe (TPIPE) that are associated with the transaction processing in the dependent region.

When the dependent region is waiting for a synchronous program switch message, the continuation line displays the target transaction code and when the synchronous program switch request will time out. The time value is in the format YYDDD/HHMMSS.

#### **WAIT-CMD/PENDING**

A /DBDUMP, /DBRECOVERY, or /START command is in progress.

The status of the MPP region continues to show WAIT-CMD/PENDING until the command completes and the region is rescheduled. To clear a WAIT-CMD/PENDING status without rescheduling the MPP region, issue a /STA DB command against one of the databases in the intent list for the PSB used by the MPP region that failed to initially schedule. Issuing this command drains the scheduler subqueue (SUBQ) 4 and posts all PSTs there, including any suspended MPP regions.

#### **WAIT-EPCB POOL**

A temporary shortage of EPCB pool space exists.

#### **WAITING**

The MPP region is waiting for work.

#### **WAIT-INPUT**

The application program is in WAIT-FOR-INPUT (WFI) mode.

#### **WAIT-INTENT**

The application program's intent for a database conflicts with the use of the database by a scheduled program.

#### **WAIT-INTENT/POOL**

Indicates either the application program's intent for a database conflicts with the use of the database by a scheduled program, or a temporary shortage of DMB, PSB, or PSB work area pool space exists.

#### **WAIT-INTENT SCHD**

The IMS transaction scheduler detected an application scheduling intent failure. (For example, Load Balancing.)

#### **WAIT-I/O PREVEN**

A BMP region that accesses a GSAM database cannot schedule until I/O prevention has completed.

#### **WAIT-MESSAGE**

The application program is in a pseudo WAIT-FOR-INPUT (WFI) mode. The application is scheduled and is waiting for a message.

#### **WAIT-POOLSPACE**

A temporary shortage of DMB, PSB, or PSB work area pool space exists.

#### **WAIT-QSC**

The application program (BMP, MPP, IFP, or DBCTL) is waiting for the quiesced state of a database that it tries to access to be released.

#### **WAIT-QSN**

The dependent region was placed in a wait state by the Queue Space Notification Exit (DFSQSPC0). If you are using a queue

|  
|  
|

|  
|  
management product with IMS, refer to the product's documentation for further information on resolving this status.

#### **WAIT-RRS PC**

The application program has a protected conversation with an OTMA client that is processing a sync point. The sync point can continue after the OTMA client issues either an SRRCMIT or SRRBACK call. Or, the application program is part of a cascaded family and is processing a sync point. APPC/OTMA SMQ Enablement uses RRS cascaded transaction support to synchronize the back-end and the front-end system.

When a dependent region is found in this state, a continuation line is inserted into the display, which shows the recovery unit ID (URID) and the originating IMS system ID (ORIGIN) that are associated with the transaction processing in the dependent region.

#### **WAIT-SWITCHOVER**

The alternate system is tracking the active system.

#### **WAIT-SYNCPOINT**

The application in the region is now in SYNC POINT.

When a dependent region is found in this state, a continuation line is inserted into the display, which shows the recovery unit ID (URID) and the originating IMS system ID (ORIGIN) that are associated with the transaction processing in the dependent region.

#### **WAIT-XCF**

The application program has a standard conversation with an APPC or OTMA client that is processing a sync point. The sync point can continue after the client issues either a positive acknowledgment (ACK) or a negative acknowledgment (NAK). The APPC/OTMA SMQ enablement uses XCF to communicate between the back-end and the front-end IMS systems.

A standard conversation is indicating a synchronous SQ transaction with sync level of NONE or CONFIRM.

#### **TERM-WAIT XCF**

Region termination is in progress and the application in the region is in the sync point. This status is displayed when an application program has a standard conversation with an APPC or OTMA client that is processing a sync point. The sync point can continue after the client issues either an ACK or NAK. The APPC/OTMA SMQ enablement uses XCF to communicate between the back-end and front-end IMS systems.

When a dependent region is found in this state, a continuation line is inserted into the display. This line shows either the target member (TMEM) and the transaction pipe (TPIPE) for an OTMA client, or the network ID (NETWORKID) and the logic unit name (LUNAME) for an APPC client, and the originating IMS system ID (ORIGIN) that is associated with the transaction processing in the dependent region.

#### **TMEM**

A 1- to 16-alphanumeric character that displays the OTMA client target member.

**TPIPE** A 1- to 8-alphanumeric character that displays the OTMA output message destination.

**LUNAME**

A 1- to 8-alphanumeric character NETWORK ID concatenated with a 1- to 8-alphanumeric character LU name.

**CLASS**

One of the classes associated with the region. The region can have from one to four classes, whose values range from 1 to 999.

**Examples**

The examples of the /DISPLAY ACTIVE command are organized into the following three sections:

- "Examples of /DISPLAY ACTIVE"
- "Examples of /DISPLAY ACTIVE DC" on page 550
- "Examples of /DISPLAY ACTIVE REG" on page 552

The following are examples of the /DISPLAY ACT command.

**Examples of /DISPLAY ACTIVE**

The following series of examples show the output of the /DISPLAY ACTIVE command for various configuration scenarios.

*Example 1 for /DISPLAY ACT command*

Entry ET:

/DISPLAY ACT

Response ET:

REGID	JOBNAME	TYPE	TRAN/STEP	PROGRAM	STATUS	CLASS
	MSGRGN	TP	NONE			
	BATCHREG	BMP	NONE			
	FPRGN	FP	NONE			
	DBTRGN	DBT	NONE			
1	IMSMPP0	TPE	DSN8PS	DSN8IP13		1, 3
	BATCHREG	BMP	NONE			
	FPRGN	FP	NONE			
	DBR3CTA3	DBRC				
	DLI3CTA3	DLS				
VTAM ACB OPEN				-LOGONS ENABLED		
APPLID=APPL8		GRSNAME=		STATUS=DISABLED		
TCP/IP_GENIMSID=		STATUS=DISABLED				
LINE	ACTIVE-IN	-	1	ACTIV-OUT	-	0
NODE	ACTIVE-IN	-	0	ACTIV-OUT	-	4
LINK	ACTIVE-IN	-	0	ACTIV-OUT	-	0
*89041/163619*						

*Example 2 for /DISPLAY ACT command*

Entry ET:

/DISPLAY A

Response ET:

REGID	JOBNAME	TYPE	TRAN/STEP	PROGRAM	STATUS	CLASS
1	I3YMPP	TP			WAITING	4, 1, 2, 3
	BATCHREG	BMP	NONE			

```

          FPRGN      FP      NONE
          DBRC       DBRC
VTAM ACB CLOSED
APPLID=APPL8      GRSNAME=          STATUS=DISABLED
TCPIP_GENIMSID=   STATUS=DISABLED
LINE ACTIVE-IN - 1 ACTIV-OUT - 0
NODE ACTIVE-IN - 0 ACTIV-OUT - 0
*11041/100117*

```

*Example 3 for /DISPLAY ACT command*

Entry ET:

```
/DISPLAY A
```

Response ET:

```

REGID JOBNAME  TYPE  TRAN/STEP PROGRAM  STATUS          CLASS
   2 I431MPP   TP   SKS7      DFSDDL7          1, 3, 6, 7
   3 IMS1BMP   BMP  REGION    SIS04P01
   1 I431IFP   FPM  TXCDRN07  DDLTRN07
      R3       DBRC
VTAM ACB OPEN          -LOGONS ENABLED
APPLID=APPL8      GRSNAME=          STATUS=DISABLED
TCPIP_GENIMSID=   STATUS=DISABLED
LINE ACTIVE-IN - 2 ACTIV-OUT - 0
NODE ACTIVE-IN - 0 ACTIV-OUT - 0
LINK ACTIVE-IN - 0 ACTIV-OUT - 0
*11041/144425*  DFSRSENM  ACTIVE

```

Explanation: The system is XRF capable so the date time stamp includes the RSE name and system indicator.

*Example 4 for /DISPLAY ACT command*

Entry ET:

```
/DISPLAY A
```

Response ET:

```

REGID JOBNAME  TYPE  TRAN/STEP PROGRAM  STATUS          CLASS
   1 I431MPP   TP   TXCDRN18  DDLTRN18  WAIT-INPUT      1, 3, 6, 7
      BATCHREG  BMP  NONE
   2 I431IFP   FPM  NO MSG.   DDLTRN07
      R3       DBRC
VTAM ACB OPEN          -LOGONS ENABLED
APPLID=APPL8      GRSNAME=          STATUS=DISABLED
TCPIP_GENIMSID=   STATUS=DISABLED
LINE ACTIVE-IN - 1 ACTIV-OUT - 0
NODE ACTIVE-IN - 0 ACTIV-OUT - 0
LINE ACTIVE-IN - 0 ACTIV-OUT - 0
*11041/132348*  DFSRSENM  ACTIVE

```

Explanation: Transaction TXCDRN18 is waiting for an input message. Program DDLTRN07 currently has no messages to process. Region 1 is in WAIT-FOR-INPUT (WFI) mode.

*Example 5 for /DISPLAY ACT command*

Entry ET:

```
/DISPLAY A
```

Response ET:

```

REGID JOBNAME  TYPE  TRAN/STEP  PROGRAM  STATUS          CLASS
   2 I431MPP   TP    SKS7      DFSDDL7  WAIT-MESSAGE    1, 3, 6, 7
   3 IMS1BMP   BMP   REGION    SIS04P01
   1 I431IFP   FPM   TXCDRN07  DDLTRN07
      R3      DBRC
VTAM ACB OPEN                -LOGONS ENABLED
APPLID=APPL8      GRSNAME=          STATUS=DISABLED
TCPIP_GENIMSID=   STATUS=DISABLED
LINE ACTIVE-IN - 2 ACTIV-OUT - 0
NODE ACTIVE-IN - 0 ACTIV-OUT - 0
LINK ACTIVE-IN - 0 ACTIV-OUT - 0
*11041/144425* DFSRSENM ACTIVE

```

Explanation: Transaction SKS7 is waiting for an input message. Program DFSDDL7 currently has no messages to process. Region 2 is in Pseudo Wait-For-Input (PWFI) mode.

*Example 6 for /DISPLAY ACT command*

Entry ET:

```
/DISPLAY A
```

Response ET:

```

** SUBSYSTEM NAME **
  SYS3
REGID JOBNAME TYPE  TRAN/STEP  PROGRAM  STATUS          CLASS
   2 I431MPP   TP    SKS7      DFSDDL7  WAIT-MESSAGE    1,3,6,7
   3 IMS1BMP   BMP   REGION    SIS04P01
   1 I431IFP   FPM   TXCDRN07  DDLTRN07
      R3      DBRC
VTAM ACB OPEN                -LOGONS ENABLED
APPLID=APPL8      GRSNAME=          STATUS=DISABLED
TCPIP_GENIMSID=   STATUS=DISABLED
LINE ACTIVE-IN- 2 ACTIV-OUT- 0
NODE ACTIVE-IN- 0 ACTIV-OUT- 0
LINK ACTIVE-IN- 0 ACTIV-OUT- 0
*11041/144425* DFSRSENM ACTIVE

```

Explanation: Transaction SKS7 is waiting for an input message. Program DFSDDL7 currently has no messages to process. Region 2 is in Pseudo Wait-For-Input (WPFI) mode.

*Example 7 for /DISPLAY ACT command*

Entry ET:

```
/DISPLAY ACTIVE
```

Response ET:

```

REGID JOBNAME  TYPE  TRAN/STEP  PROGRAM  STATUS          CLASS
   1 IMSMPPA   TPI   TPN1      DFSCPIC  WAIT-BLOCKMOVER 20, 22, 24, 26
   2 IMSMPPB   TPI   TPN2      DFSCPIC  21, 23, 25, 22
   3 IMSMPPC   TP    TPN3      SAAPSB1  28, 29, 24, 23
   4 IMSMPPD   TPI   TPN4      DFSCPIC  WAIT-SYNCPOINT  14, 15, 16, 17
   4 IMSMPPE   TP    TRAN1     LU2PGMA  WAIT-INPUT       2, 4, 6, 8
   5 IMSMPPF   TP    TRAN2     LU2PGMB  1, 3, 5, 7
VTAM ACB OPEN                / APPC LUNAME=IMSLUNME ENABLED -LOGONS ENABLED
APPLID=APPL8      GRSNAME=          STATUS=DISABLED
TCPIP_GENIMSID=   STATUS=DISABLED
LINE ACTIVE-IN - 2 ACTIV-OUT - 0
NODE ACTIVE-IN - 0 ACTIV-OUT - 0
LINK ACTIVE-IN - 0 ACTIV-OUT - 0
*90332/114253*

```

*Example 8 for /DISPLAY ACT command*

Entry ET:

/DISPLAY ACTIVE

Response ET:

```
REGID JOBNAME  TYPE  TRAN/STEP PROGRAM  STATUS      CLASS
   1 IMSPPA    TP      REGION    SIS04P01  WAITING     4, 1, 2
   2 IMS1BMP   BMP     REGION    SIS04P01  WAIT-AOI
   4 IMS2BMP   BMP     REGION    SIS04P02  WAIT-AOI
   3 I510MPP   TP      REGION              WAIT-EPCB POOL  8,  8,  8,  8
   2 IMS3BMP   BMP     REGION              WAIT-EPCB POOL

      FPRGN    FP      NONE
      DBRC     DBRC

VTAM ACB CLOSED
APPLID=APPL8      GRSNAME=          STATUS=DISABLED
TCPIP_GENIMSID=  STATUS=DISABLED
LINE ACTIVE-IN - 1 ACTIV-OUT - 0
NODE ACTIVE-IN - 0 ACTIV-OUT - 0
*11280/095345*
```

*Example 9 for /DISPLAY ACTIVE command*

Entry ET:

/DISPLAY ACTIVE

Response ET:

```
REGID JOBNAME  TYPE  TRAN/STEP PROGRAM  STATUS      CLASS
   1 I431MPP   TP      REGION              TERMINATING 1, 2, 3, 4
      BATCHREG BMP NONE
   2 I431IFP   FPM    NO MSG.  DDLTRN07
      R3 DBRC

VTAM ACB OPEN  -LOGONS ENABLED
APPLID=APPL8      GRSNAME=          STATUS=DISABLED
TCPIP_GENIMSID=  STATUS=DISABLED
LINE ACTIVE-IN - 1 ACTIV-OUT - 0
NODE ACTIVE-IN - 0 ACTIV-OUT - 0
LINE ACTIVE-IN - 0 ACTIV-OUT - 0
*11041/132348* DFSRSENM ACTIVE
```

Explanation: MPP region I431MPP is in the process of terminating because of a /STOP REGION command. Region 1 is in TERMINATING mode.

*Example 10 for /DISPLAY ACT command*

Entry ET:

/DISPLAY ACTIVE

Response ET:

```
|      REGID JOBNAME  TYPE  TRAN/STEP PROGRAM  STATUS      CLASS
|      MSGRGN    TP      NONE
|      BATCHREG   BMP     NONE
|      FPRGN     FP      NONE
|      DBTRGN    DBT     NONE
|      DBRSCHA5  DBRC
|      DLISCHA5  DLS
|      VTAM ACB OPEN  -LOGONS DISABLED
|      IMSLU=N/A.N/A  APPC STATUS=DISABLED  TIMEOUT= 0  MAXC= 5000
|      APPLID=APPL8  USERVAR=IMS1
```

```

|          TCPIP_GENIMSID=          STATUS=DISABLED
|          LINE ACTIVE-IN -      1 ACTIV-OUT -      0
|          NODE ACTIVE-IN -      0 ACTIV-OUT -      0
|          LINK ACTIVE-IN -      0 ACTIV-OUT -      0
|          *11080/183050* DFSRSENM ACTIVE

```

Explanation: The IMS subsystem is both XRF-capable and RSR-capable.

## Examples of /DISPLAY ACTIVE DC

The following series of examples show the output of the /DISPLAY ACTIVE DC command for various configuration scenarios.

### *Example 1 for /DISPLAY ACTIVE DC command*

Entry ET:

```
/DISPLAY ACTIVE DC
```

Response ET:

```

VTAM STATUS AND ACTIVE DC COUNTS
VTAM ACB OPEN                      -LOGONS ENABLED
IMSLU=SYS1.IMSLUNME                STATUS=ENABLED
APPLID=APPL8                        USERVAR=IMS1
TCPIP_GENIMSID=                     STATUS=DISABLED
LINE ACTIVE-IN -      2 ACTIV-OUT -      0
NODE ACTIVE-IN -      0 ACTIV-OUT -      0
LINK ACTIVE-IN -      0 ACTIV-OUT -      0
*11232/114253*

```

### *Example 2 for /DISPLAY ACTIVE DC command*

Entry ET:

```
/DISPLAY ACTIVE DC
```

Response ET:

```

|          VTAM STATUS AND ACTIVE DC COUNTS
|          VTAM ACB CLOSED                      -APPLID ACB
|          VTAM ACB OPEN                      -LOGONS ENABLED -MNPS ACB
|          IMSLU=N/A.N/A                      APPC STATUS=DISABLED TIMEOUT=  0  MAXC= 5000
|          OTMA GROUP=N/A                      STATUS=NOTACTIVE
|          APPC/OTMA SHARED QUEUE STATUS - LOCAL=INACTIVE GLOBAL=INACTIVE
|          APPC/OTMA RRS MAX TCBS - 2 ATTACHED TCBS - QUEUED RRSWKS -  0  N
|          APPLID=APPL8                      GRSNAME=                      STATUS=DISABLED MNPS=APPL1
|          TCPIP_GENIMSID=                     STATUS=DISABLED
|          LINE ACTIVE-IN -      1 ACTIV-OUT -      0
|          NODE ACTIVE-IN -      0 ACTIV-OUT -      0
|          LINK ACTIVE-IN -      0 ACTIV-OUT -      0
|          *11069/143346* DFSRSENM ACTIVE

```

Explanation: An XRF system is using MNPS. The MNPS ACB is APPL1, which is open and is accepting logons. The APPLID ACB is APPL8, but it is closed.

### *Example 3 for /DISPLAY ACTIVE DC command*

Entry ET:

```
/DISPLAY ACTIVE DC
```

Response ET:

```

DFS000I  VTAM STATUS AND ACTIVE DC COUNTS
DFS000I  VTAM ACB OPEN          -LOGONS DISABLED
DFS000I  IMSLU=N/A.N/A          APPC STATUS=DISABLED TIMEOUT=  0  MAXC= 5000
DFS000I  OTMA GROUP=N/A        STATUS=NOTACTIVE
DFS000I  APPC/OTMA SHARED QUEUE STATUS - LOCAL=ACTIVE-XCF GLOBAL=ACTIVE-XCF
DFS000I  APPC/OTMA SHARED QUEUES LOGGING=Y
DFS000I  APPC/OTMA RRS MAX TCBS - 40 ATTACHED TCBS -  1 QUEUED  RRSWKS-  0
DFS000I  APPLID=APPL8  GRSNAME=          STATUS=DISABLED
DFS000I  TCPIP_GENIMSID=      STATUS=DISABLED
DFS000I  LINE ACTIVE-IN  -  1 ACTIV-OUT -  0
DFS000I  NODE ACTIVE-IN  -  0 ACTIV-OUT -  0
DFS000I  LINK ACTIVE-IN  -  0 ACTIV-OUT -  0
DFS000I  *09039/142037*      IMS1

```

**Example 4 for /DISPLAY ACTIVE DC command showing TCP/IP generic resources**

Two examples of the /DISPLAY ACT DC command are shown here. The first example is issued on an IMS system within a TCP/IP generic resource group. The second example is issued on the remote partner IMS system that is connects to the TCP/IP generic resource group.

Entry ET in the TCP/IP generic resource group:

```
/DISPLAY ACTIVE DC
```

Response ET in the TCP/IP generic resource group:

```

VTAM STATUS AND ACTIVE DC COUNTS
VTAM ACB OPEN          -LOGONS ENABLED
IMSLU=N/A.N/A          APPC STATUS=DISABLED TIMEOUT=  0  MAXC= 5000
OTMA GROUP=N/A        STATUS=NOTACTIVE
APPC/OTMA SHARED QUEUE STATUS - LOCAL=INACTIVE GLOBAL=INACTIVE
APPC/OTMA SHARED QUEUES LOGGING=N
APPC/OTMA RRS MAX TCBS - 40 ATTACHED TCBS -  1 QUEUED  RRSWKS-  0
APPLID=L6APPL3  GRSNAME=          STATUS=DISABLED
TCPIP_GENIMSID=IMS    STATUS=ACTIVE
LINE ACTIVE-IN  -  1 ACTIV-OUT -  0
NODE ACTIVE-IN  -  0 ACTIV-OUT -  0
LINK ACTIVE-IN  -  0 ACTIV-OUT -  0
*11130/230804*

```

Entry ET in the partner of a TCP/IP generic resource group:

```
/DISPLAY ACTIVE DC
```

Response ET in the partner of a TCP/IP generic resource group:

```

VTAM STATUS AND ACTIVE DC COUNTS
VTAM ACB OPEN          -LOGONS ENABLED
IMSLU=N/A.N/A          APPC STATUS=DISABLED TIMEOUT=  0  MAXC= 5000
OTMA GROUP=N/A        STATUS=NOTACTIVE APPC/OTMA SHARED QUEUE STATUS - LOCAL=INACTIVE
GLOBAL=INACTIVE
APPC/OTMA SHARED QUEUES LOGGING=N
APPC/OTMA RRS MAX TCBS - 40 ATTACHED TCBS -  1 QUEUED  RRSWKS-  0
APPLID=L6APPL1  GRSNAME=          STATUS=DISABLED
TCPIP_GENIMSID=      STATUS=DISABLED
LINE ACTIVE-IN  -  1 ACTIV-OUT -  0
NODE ACTIVE-IN  -  0 ACTIV-OUT -  0
LINK ACTIVE-IN  -  0 ACTIV-OUT -  0
*11130/231519*

```

**Example 5 for /DISPLAY ACTIVE DC command showing TCP/IP generic resources with XRF**

The following example shows the output of the /DISPLAY ACT DC command for a system that is participating in both an XRF complex and in an MSC TCP/IP generic resource group.

In the example, the use of XRF is indicated by the presence of the USERVAR field and the DFSRSENM field. The use of TCP/IP generic resources is indicated by a non-blank value on the TCPIP\_GENIMSID field and a status of ACTIVE.

Entry ET in the TCP/IP generic resource group:

/DISPLAY ACTIVE DC

Response ET in the TCP/IP generic resource group:

```

DFS000I      VTAM STATUS AND ACTIVE DC COUNTS
DFS000I      VTAM ACB OPEN          -LOGONS ENABLED
DFS000I      IMSLU=N/A.N/A          APPC STATUS=DISABLED TIMEOUT=  0  MAXC= 5000
DFS000I      OTMA GROUP=N/A          STATUS=NOTACTIVE
DFS000I      APPC/OTMA SHARED QUEUE STATUS - LOCAL=UNSUPPORTED  GLOBAL=INACTIVE
DFS000I      APPC/OTMA SHARED QUEUES LOGGING=N
DFS000I      APPC/OTMA RRS MAX TCBS - 40 ATTACHED TCBS -  2  QUEUED RRSWKS-  0
DFS000I      APPLID=APPL8          USERVAR=IMS1
DFS000I      TCPIP_GENIMSID=IMS      STATUS=ACTIVE
DFS000I      LINE ACTIVE-IN -  1  ACTIV-OUT -  0
DFS000I      NODE ACTIVE-IN -  0  ACTIV-OUT -  0
DFS000I      LINK ACTIVE-IN -  0  ACTIV-OUT -  0
DFS000I      *11201/163416* DFSRSENM ACTIVE

```

## Examples of /DISPLAY ACTIVE REG

### Example 1 for /DISPLAY ACTIVE REG command

Entry ET:

/DISPLAY ACTIVE REG

Response ET:

REGID	JOBNAME	TYPE	TRAN/STEP	PROGRAM	STATUS	CLASS	IMS1
2	MPPI3	TP	SHF1	PMVAPZ12	WAIT-QSC	1, 2, 3, 4	
1	MPPI2	TP	NQF1	PMVAPZ12	WAIT-QSC	1, 2, 3, 4	
	JMPRGN	JMP	NONE	IMS1			
3	BMPI4	BMP		PMVAPZ12	WAIT-QSC	IMS1	
	JBPRGN	JBP	NONE	IMS1			
	FPRGN	FP	NONE	IMS1			
	DBTRGN	DBT	NONE	IMS1			
	DBRISAK	DBRC				IMS1	
	DLIISAK	DLS	IMS1				
*08030/132621*			IMS1				

Explanation: The status of MPPI2, MPPI3, and BMPI4 transactions are shown as WAIT-QSC, which indicates that the database that these transactions are trying to access are in quiesce state and that the transactions are waiting for the UPDATE STOP(QUIESCE) command to be issued to release the quiesce.

### Example 2 for /DISPLAY ACTIVE REG command

Entry ET:

/DISPLAY ACTIVE REG

Response ET:

REGID	JOBNAME	TYPE	TRAN/STEP	PROGRAM	STATUS	CLASS
1	MPP1A	TP	APOL11	APOL1	WAIT-CALLOUT	1
		TMEM:	HWS1		TPIPE:	TPIPE1

```

      JMPRGN  JMP  NONE
      JBPRGN  JBP  NONE
      BATCHREG BMP  NONE
      FPRGN   FP   NONE
      DBTRGN  DBT  NONE
      DBRZCSAJ DBRC
      DLIZCSAJ DLS
*08235/173441*

```

Explanation: Message processing program APOL1 is processing transaction APOL11, which is waiting for a response to a synchronous callout request (WAIT-CALLOUT) from transaction pipe TPIPE1.

*Example 3 for /DISPLAY ACTIVE REG command*

Entry ET:

```
/DISPLAY ACTIVE REG
```

Response ET:

```

REGID JOBNAME  TYPE  TRAN/STEP  PROGRAM  STATUS CLASS  IMS1
      JMPRGN   JMP   NONE   IMS1
      1 IMSMPPA  TPI   APOL11 IMS1  APOL1 WAIT-RRS/PC 1,2,3,4
URID:  C2D6B6917DE82000000000001010000 ORIGIN: IMS1
      2 IMSMPPB  TPI   APOL12 IMS1  APOL1 TERM-WAIT RRS 1,2,3,4
URID:  C2D6B6917DE83000000000001010000 ORIGIN: IMS1
      3 IMSMPPB  TPI   APOL13 IMS1  APOL1 WAIT-XCF 1,2,3,4
TMEM:  HWS1                    TPIPE: CLIENT01 ORIGIN: IMS1
      4 IMSMPPB  TPI   APOL14 IMS1  APOL1 TERM-WAIT XCF 1,2,3,4
LUNAME: IMSNETWK.LU62IMS1 ORIGIN: IMS1
      5 IFP2A    FPM   TXCDRN01 DDLTRN01 WAIT-RRS/PC
URID:  D02C7FD77E31600000000001030000 ORIGIN: IMS1
      JBPRGN   JBP   NONE   IMS1
      BATCHREG BMP   NONE   IMS1
      FPRGN    FP    NONE   IMS1
      DBTRGN   DBT   NONE   IMS1
      DBRICTAB DBRC                   IMS1
      DLISDEP  DLS                   IMS1
*16019/150537*

```

*Example 4 for /DISPLAY ACTIVE REG command*

Entry ET:

```
/DIS ACTIVE REG
```

Response ET:

```

REGID JOBNAME  TYPE  TRAN/STEP  PROGRAM  STATUS      CLASS
      1 MPP1A    TP   APOL11    APOL1    WAIT-CALLOUT 1
                                TRAN:SKS1    END TIME: 2010.172 12:45:04
      JMPRGN   JMP   NONE
      JBPRGN   JBP   NONE
      BATCHREG BMP   NONE
      FPRGN    FP    NONE
      DBTRGN   DBT   NONE
      DBRZCSAJ DBRC
      DLIZCSAJ DLS
*10172/120101*
*
```

The APOL11 transaction is in the WAIT-CALLOUT state while it waits for a response to a synchronous program switch request. SKS1 is the transaction name

that was requested in the ICAL call. The timeout value of the ICAL call was used to calculate the value of the END TIME field.

**Related reference:**

Chapter 2, "Command keywords and their synonyms," on page 63

## /DISPLAY AFFIN command

Use the /DISPLAY AFFIN command to display the current link or node affinities that an IMS subsystem has within a TCP/IP or VTAM generic resource group.

Subsections:

- "Environment"
- "Syntax"
- "Keywords"
- "Usage notes" on page 555
- "Equivalent IMS type-2 commands" on page 555
- "Examples" on page 555

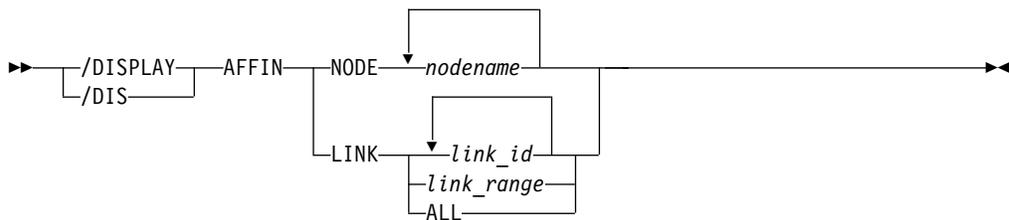
### Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

Table 272. Valid environments for the /DISPLAY AFFIN command and keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
/DISPLAY	X	X	X
AFFIN	X		X
NODE	X		X
LINK	X		X

### Syntax



### Keywords

The following keywords are valid for the /DISPLAY AFFIN command:

**NODE**

Specify one or multiple nodes. You cannot specify generic names such as *nodename\** or ALL.

**LINK**

Specify one or multiple links, a range of links, or ALL. You must specify *link\_range* in the form of *link\_a-link\_b*; for example, 1-3.

## Usage notes

This command is valid only when you use TCP/IP or VTAM generic resources.

## Equivalent IMS type-2 commands

The following table shows variations of the /DISPLAY AFFIN command and the IMS type-2 commands that perform similar functions.

Table 273. Type-2 equivalents for the /DISPLAY AFFIN command

Task	/DISPLAY AFFIN command	Similar IMS type-2 command
Displays TCP/IP generic resource affinities.	/DISPLAY AFFIN LINK <i>links</i>	QUERY MSLINK NAME(linkname   *) SHOW(AFFIN)
Displays VTAM generic resource affinities.	/DISPLAY AFFIN NODE <i>node</i>	QUERY NODE SHOW(AFFIN)

## Examples

The following are examples of the /DISPLAY AFFIN command.

### Example 1 for /DISPLAY AFFIN command

Entry ET:

```
/DISPLAY AFFIN NODE NDSL2A1
```

Response ET:

```
NODE      APPLID
NDSL2A1  APPL9
*97098/162106*
```

Explanation: The node has an affinity for APPL ID APPL9.

### Example 2 for /DISPLAY AFFIN command

Entry ET:

```
/DISPLAY AFFIN NODE LU37722
```

Response ET:

```
NODE      APPLID
LU37722  N/A
*97098/162156*
```

Explanation: The node has no affinities.

### Example 3 for /DISPLAY AFFIN command

Entry ET:

```
/DISPLAY AFFIN LINK 22
```

Response ET:

```
DFS000I      LINK LINKNAME  NODE      APPLID
DFS000I      22 LNK12T01      IMS1
```

Explanation: Link 22 has affinity with the current IMS system under the TCP/IP generic resource ID of IMS1.

*Example 4 for /DISPLAY AFFIN command*

Entry ET:

```
/DISPLAY AFFIN LINK ALL
```

Response ET:

DFS000I	LINK	LINKNAME	NODE	APPLID
DFS000I	1	LNK32V01	L6APPL1	N/A
DFS000I	2	LNK32V02	L6APPL1	N/A
DFS000I	3	LNK32V03	L6APPL1	N/A
DFS000I	4	LNK31V01	L6APPL3	APPL7
DFS000I	5	LNK31V05	L6APPL3	APPL7
DFS000I	6	LNK32V05	L6APPL1	N/A
DFS000I	7	LNK31T01		N/A
DFS000I	22	LNK32T01		IMS1
DFS000I	23	LNK33T01		IMS1

Explanation: Links 4 and 5 to IMS node L6APPL3 have affinity for APPL7 in the IMSplex. Links 1, 2, 3, and 6 are VTAM links to IMS node L6APPL1 and have no affinity. Link 7 is not a VTAM link.

Links 22 and 23 are MSC TCP/IP links that have affinity to the current IMS system within the IMS1 TCP/IP generic resource group.

**Related reference:**

 [QUERY NODE command \(Commands\)](#)

---

## **/DISPLAY AOITKN command**

Use the /DISPLAY AOITKN command to display all Automated Operator Interface (AOI) tokens in the system.

Subsections:

- “Environment”
- “Syntax”
- “Usage notes” on page 557
- “Output fields” on page 557
- “Examples” on page 557

### **Environment**

The /DISPLAY AOITKN command is valid in DB/DC, DBCTL, and DCCTL environments.

### **Syntax**

```
▶▶ /DISPLAY AOITKN ▶▶  
   └─ /DIS ─┘
```

## Usage notes

If QCT=0 and W-REGID=NONE, then the AOI token is deleted at the next checkpoint.

## Output fields

The following output fields are displayed for each AOI token:

### AOITKN

AOI token name.

### ENQCT

Total number of messages enqueued. In a shared-queues environment, only shows messages enqueued for the local subsystem.

**QCT** Number of messages currently queued to the token and not yet retrieved by the AO application. In a shared-queues environment, only shows messages enqueued for the local subsystem.

**SEGS** Number of segments in messages queued (QCT) to the AOI token.

### W-REGID

The region IDs of AO applications that issued a GMSG call with the WAITAOI subfunction specified. This AO application is in a wait state, since there are currently no messages for it to retrieve.

## Examples

Entry ET:

```
/DISPLAY AOITKN
```

Response ET:

AOITOKEN	ENQCT	QCT	SEGS	W-REGID
AOITOKN1	4	2	12	NONE
AOITOKN2	0	0	0	2,4
AOITOKN3	1	0	0	NONE
AOITOKN4	1000000000	1000000000	2147483647	NONE

\*92280/095345\*

---

## /DISPLAY APPC command

Use the /DISPLAY APPC command to display inquiries about LU 6.2 related activities, including the current outbound LU if it is different from the base LU.

Subsections:

- "Environment"
- "Syntax" on page 558
- "Output fields" on page 558
- "Examples" on page 559

## Environment

The /DISPLAY APPC command is valid in DB/DC and DCCTL environments.

## Syntax

► `/DISPLAY` `APPC` 

## Output fields

The following output fields are displayed for the `/DISPLAY APPC` command:

### IMSLU

The IMSLU is a base IMS LU name. This LU is always a network-qualified LU name. It is displayed as N/A.N/A if IMS is not connected to APPC/MVS. For example, if status is `DISABLED`, `FAILED`, or `STARTING`, N/A.N/A applies. For the `CANCEL` status, the field can contain either the LU name or N/A.N/A.

### #APPC-CONV

LU 6.2 inbound and outbound conversation count.

### SECURITY

RACF security level, which is one of the following:

- `CHECK`
- `FULL`
- `NONE`
- `PROFILE`

### STATUS

The current APPC connection status. The possible values for APPC connection status are:

#### **CANCEL**

Shown after the `/STOP APPC CANCEL` command.

#### **DISABLED**

Shown when APPC/IMS is not identified or connected to APPC/MVS.

#### **ENABLED**

Shown after successful `/START APPC` command completion.

#### **FAILED**

Shown after unsuccessful `/START APPC` command.

#### **OUTBOUND**

Shown on the XRF alternate system as it tracks the active. It is also shown on the active if IMS is unable to initiate normal APPC processing. For example, the `/START APPC` command can be reissued to attempt APPC enablement.

#### **PURGING**

Shown after the `/PURGE APPC` command.

#### **STARTING**

Shown after the `/START APPC` is issued and before the status is set to `ENABLED` or `FAILED`.

#### **STOPPED**

Shown after the `/STOP APPC` command.

**DESIRED**

Desired APPC connection status. This status is changed by the /PURGE, /START and /STOP APPC (CANCEL) commands. The possible values for desired APPC connection status are:

**CANCEL**

Shown after /STOP APPC CANCEL

**DISABLED**

Shown when APPC/IMS is not identified or connected to the APPC/MVS

**ENABLED**

Shown after /START APPC

**OUTBOUND**

Shown on the XRF alternate system as it tracks the active

**PURGING**

Shown after /PURGE APPC

**STOPPED**

Shown after /STOP APPC

**GRNAME**

The IMS/APPc generic LU name (if VTAM Generic Resources is activated).

**TYPE** BASE for BASE LU or OUTB for OUTBOUND LU.

**Examples**

Entry ET:

```
/DISPLAY APPC
```

Response ET:

```
IMSLU #APPC-CONV SECURITY STATUS DESIRED
IMSLUNME 0 FULL ENABLED ENABLED
*90347/114253*
```

Entry ET:

```
/DISPLAY APPC
```

Response ET:

```
IMSLU #APPC-CONV SECURITY STATUS DESIRED
IMSLUNME 0 FULL DISABLED ENABLED
*90347/114253*
```

Entry ET:

```
/DISPLAY APPC
```

Response ET:

```
IMSLU #APPC-CONV SECURITY STATUS DESIRED
SYS1.IMSLUNME 0 FULL STOPPED STOPPED
*92232/114253*
```

---

## /DISPLAY AREA command

Use the /DISPLAY AREA command to display data sets, status conditions, control intervals, and databases associated with an area.

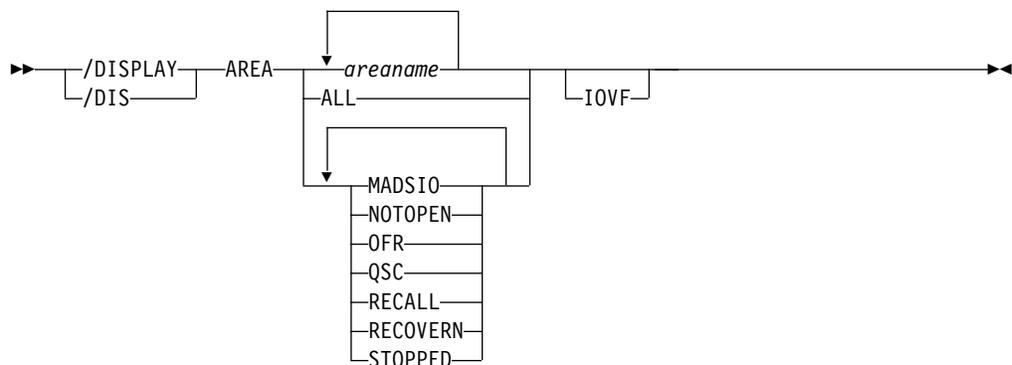
Subsections:

- "Environment"
- "Syntax"
- "Keywords and reserved parameters"
- "Usage notes" on page 561
- "Equivalent IMS type-2 commands" on page 562
- "Output fields" on page 562
- "Examples" on page 563

### Environment

The /DISPLAY AREA command is valid in DB/DC and DBCTL environments.

### Syntax



### Keywords and reserved parameters

The following keywords can be specified for the /DISPLAY AREA command:

#### IOVF

Displays the independent overflow area statistics.

If the /DISPLAY AREA IOVF command is entered on an RSR tracking subsystem, the IOVF statistics might be incorrect.

When the ALL parameter is used with the IOVF keyword (for example, /DIS AREA ALL IOVF), performance can be affected depending on the number and size of areas involved. If large areas, large numbers of areas, or both are involved, the control region can appear to be stopped while processing this command.

Areas are selected for display based on the attributes specified. For example, /DISPLAY AREA STOPPED only displays the areas that are currently stopped. The attribute display format is the same as the standard AREA display. Attributes generally correspond to the conditions displayed for areas. Any exceptions are flagged in the following list.

The attributes are reserved parameters for the /DISPLAY AREA command and cannot be used to name areas. The following list describes the attributes that can be used with the AREA keyword:

**MADSI0**

Displays the areas that are currently in a long busy condition or are in long busy recovery mode.

**NOTOPEN**

Displays the areas that have yet to be opened.

**OFR**

Displays the areas that are being brought up to the current tracking level with online forward recovery.

**QSC**

Displays the Fast Path DEDB areas that are currently quiesced by a previous UPDATE DB START(QUIESCE) or UPDATE AREA START(QUIESCE) command.

**RECALL**

Displays the areas that are in recall.

**RECOVERN**

Displays the areas that need recovery. This parameter corresponds to the area condition RECOVERY-NEEDED.

**STOPPED**

Displays the areas that are currently stopped.

## Usage notes

If an IMS Fast Path utility is active when the /DISPLAY AREA command is issued, the following output line is displayed:

```
UTILITY XXXX  PVTTOT#  YY  PVTAVL#  ZZ  IMSID:SSSS  TTTT
```

or

```
UTILITY: XXXX                               IMSID:SSSS
```

The output values represent:

- XXXX - the active utility name or VSO function
- YY - the total number of private buffers available to the utility
- ZZ - the number of unused private buffers available to the utility
- SSSS - the IMSID of the IMS system on which the utility is active
- TTTT - the IMSID of the IMS system on which the display was issued

The following IMS Fast Path utilities, VSO functions, and IBM DBTOOL utility can be displayed:

- DA00 (DBFUDA00, DEDB Alter utility)
- HSSP (High Speed Sequential Processing utility)
- HSRE (DBFUHDR0, High Speed Reorganization utility)
- MDL0 (DBFUMDL0, SDEP Delete utility)
- MSC0 (DBFUMSC0, SDEP Scan utility)
- MRI0 (DBFUMRI0, DEDB Create utility)
- MMH0 (DBFUMMH0, DEDB Compare utility)
- PRLD (VSO Preload is active)

- TOOL (an IBM DBTOOL Online utility)

## Equivalent IMS type-2 commands

The following table shows variations of the /DISPLAY AREA command and the IMS type-2 commands that perform similar functions.

Table 274. Type-2 equivalents for the /DISPLAY AREA command

Task	/DISPLAY AREA command	Similar IMS type-2 command
Displays information about an area.	/DISPLAY AREA	QUERY AREA

## Output fields

The following output fields are displayed for the /DISPLAY AREA command:

### AREANAME

Name of the Fast Path DEDB area.

### DDNAME

Names of the data sets that contain that area.

### CONDITIONS

Status conditions associated with the area or ADS.

Status conditions associated with the area include:

**IC** HSSP image copy is currently active on the area.

### MADSIO

The areas is currently in a long busy condition or long busy recovery mode.

### NOTOPEN

The area has yet to be opened. This will be done when first call is issued to the area.

### RECALL

Area is in recall.

### RECOVERY-NEEDED

The area has been marked RECOVERY-NEEDED. The Database Recovery utility needs to be run to restore this area.

### STOPPED

The area is in STOPPED status. All access to it is prohibited. This can be changed with the /START command.

**VIR** The area is a VSO area.

**PREO** The area is defined to be preopened during IMS initialization.

**PREL** The area is defined to be preloaded during IMS initialization.

### QUIESCED

The Fast Path DEDB area named on the command is currently quiesced by a previous UPDATE DB START(QUIESCE) or UPDATE AREA START(QUIESCE) command.

### QUIESCING

The Fast Path DEDB area named on the command is currently undergoing quiesce by a previous UPDATE DB START(QUIESCE) or UPDATE AREA START(QUIESCE) command.

Status conditions associated with the ADS include the following:

**COPY-PHASE**

The CREATE utility is currently in the COPY phase on this ADS. The CREATE utility must complete before anything can be done with this ADS.

**FORMAT-PHASE**

The CREATE utility is currently in the FORMAT phase on this ADS. The CREATE utility must complete before anything can be done with this ADS.

**PRE-OPEN FAILED**

XRF PREOPEN was not successful for this ADS.

**RECALL**

Area is in Recall.

**SEVERE-ERROR**

This ADS has had a severe I/O error (write error to second CI, in which Fast Path keeps its control information).

**UNAVAILABLE**

This ADS has been marked unavailable, probably because of I/O errors. (EQE REMAIN = 0).

**EQECT**

Specifies the count of I/O errors for the area.

**TOTAL SEQ DEPENDENT**

Total control intervals defined for sequential dependent space.

**UNUSED SEQ DEPENDENT**

Unused control intervals for sequential dependent space.

**TOTAL DIRECT ADDRESSABLE**

Total control intervals defined for direct addressable space.

**UNUSED DIRECT ADDRESSABLE**

Unused control intervals for direct addressable space.

**DBNAME**

Database name.

**EEQECT**

Count of write error EEQE for this area.

**IOVF** Independent overflow area statistics.

These statistics are only displayed if IOVF is specified on the /DIS AREA command (/DIS AREA ALL IOVF or /DIS AREA areaname IOVF). If the IOVF parameter is not specified on the /DIS AREA command, N/A appears in the display output.

**Examples**

The following are examples of the /DISPLAY AREA command.

*Example 1 for /DISPLAY AREA command*

Entry ET:

```
/DIS AREA db21ar1 db21ar3 db21ar6 IOVF
```

Response ET:

AREANAME	EQECT	TOTAL UNUSED	TOTAL UNUSED	DBNAME	EEQECT	CONDITIONS
DDNAME	REMAIN	SEQ DEPENDENT	DIR ADDRESSABLE			
DB21AR1	N/A	13	11	74	74-CI DEDBJN21 999	VIR, PREO, PREL
DB21AR1	10	N/A	N/A	N/A	N/A	N/A
DB21AR1B	10	N/A	N/A	N/A	N/A	N/A
DB21AR1C	10	N/A	N/A	N/A	N/A	N/A
DB21AR3	N/A	13	11	74	56-CI DEDBJN21	PREO
DB21AR3	10	N/A	N/A	N/A	N/A	N/A
DB21AR3B	10	N/A	N/A	N/A	N/A	N/A
DB21AR6	N/A	17	15	74	56-CI DEDBJN21	
DB21AR6	10	N/A	N/A	N/A	N/A	N/A

\*93076/173254\*

Explanation: DIR ADDRESSABLE means Independent Overflow part.

*Example 2 for /DISPLAY AREA command*

Entry ET:

/DIS AREA ALL

Response ET:

AREANAME	EQECT	TOTAL UNUSED	TOTAL UNUSED	DBNAME	EEQECT	CONDITIONS	
DDNAME	REMAIN	SEQ DEPENDENT	DIR ADDRESSABLE				
DB21AR0	N/A	17	15	74	N/A	DEDBJN21 999	VIR, PREO, PREL
DB21AR0	10	N/A	N/A	N/A	N/A	N/A	
DB21AR1	N/A	13	11	74	N/A	DEDBJN21	VIR, PREO
DB21AR1	10	N/A	N/A	N/A	N/A	N/A	
DB21AR1B	10	N/A	N/A	N/A	N/A	N/A	
DB21AR1C	10	N/A	N/A	N/A	N/A	N/A	
DB21AR10	N/A	17	15	74	N/A	DEDBJN21	VIR, PREO, PREL
DB21AR10	10	N/A	N/A	N/A	N/A	N/A	
DB21AR11	N/A	133	131	74	N/A	DEDBJN21	VIR
DB21AR11	10	N/A	N/A	N/A	N/A	N/A	
DB21AR2	N/A	19	17	74	0-UW	DEDBJN21	VIR, PREO, PREL
UTILITY:	HSRE	PVTTOT#	45	PVTAVL#	15		
DB21AR2	10	N/A	N/A	N/A	N/A	N/A	
DB21AR3	N/A	13	11	74	N/A	DEDBJN21	PREO
DB21AR3	10	N/A	N/A	N/A	N/A	N/A	
DB21AR3B	10	N/A	N/A	N/A	N/A	N/A	
DB21AR4	N/A	N/A	N/A	N/A	N/A	DEDBJN21	STOPPED, NOTOPEN
DB21AR5	N/A	N/A	N/A	N/A	N/A	DEDBJN21	STOPPED, NOTOPEN
DB21AR6	N/A	N/A	N/A	N/A	N/A	DEDBJN21	STOPPED, NOTOPEN
DB21AR7	N/A	N/A	N/A	N/A	N/A	DEDBJN21	NOTOPEN
DB21AR8	N/A	N/A	N/A	N/A	N/A	DEDBJN21	NOTOPEN
DB21AR9	N/A	N/A	N/A	N/A	N/A	DEDBJN21	NOTOPEN

\*93076/173254\*

Explanation: DIR ADDRESSABLE means Independent Overflow part.

*Example 3 for /DISPLAY AREA command*

Entry ET:

/DIS AREA db11ar4

Response ET:

AREANAME	EQECT	TOTAL UNUSED	TOTAL UNUSED	DBNAME	EQECT	CONDITIONS
DDNAME	REMAIN	SEQ	DEPENDENT DIR	ADDRESSABLE		
DB21AR4	N/A	N/A	N/A	N/A	N/A	DEDBJN21 999 STOPPED, NOTOPEN, RECOVERY
*97136/173254*						

Explanation: RECOVERY means this Fast Path area is undergoing recovery with the Online Recovery Service.

**Example 4 for /DISPLAY AREA command**

Entry ET:

/DIS AREA DB21AR0

Response ET:

DFS000I	AREANAME	EQECT	TOTAL UNUSED	TOTAL UNUSED	DBNAME	EEQECT	CONDITIONS	SYS3
DFS000I	DDNAME	REMAIN	SEQ	DEPENDENT DIR	ADDRESSABLE			SYS3
DFS000I	DB21AR0	N/A	29	28	74	0-UW	DEDBJN21 0 VIR, PREO, PREL	SYS3
DFS000I	UTILITY:	HSSP	PVTTOT#	45	PVTAVL#	30	IMSID: IMS2	SYS3
DFS000I	DB21AR01	10	N/A	N/A	N/A	N/A	N/A	SYS3
DFS000I	DB21AR02	10	N/A	N/A	N/A	N/A	N/A	SYS3
DFS000I	DB21AR03	10	N/A	N/A	N/A	N/A	N/A	SYS3

Explanation: The IMS Fast Path utility, HSSP, is active on IMSID=IMS2.

**Example 5 for /DISPLAY AREA command**

Entry ET:

/DIS AREA QSC

Response ET:

AREANAME	EQECT	TOTAL UNUSED	TOTAL UNUSED	DBNAME	EEQECT	CONDITIONS
DDNAME	REMAIN	SEQ	DEPENDENT DIR	ADDRESSABLE		
AXYZ01	N/A	N/A	N/A	N/A	N/A	FPDBXYZ 0 QUIESCED

Explanation: This is an example of displaying databases that have the status of quiesced.

**Related concepts:**

 [DEDB area design guidelines \(Database Administration\)](#)

**Related reference:**

 [QUERY AREA command \(Commands\)](#)

## **/DISPLAY ASMT command**

Use the /DISPLAY ASMT command to display resource assignments.

Subsections:

- “Environment” on page 566
- “Syntax” on page 566
- “Keywords” on page 567
- “Equivalent IMS type-2 commands” on page 568
- “Examples” on page 569

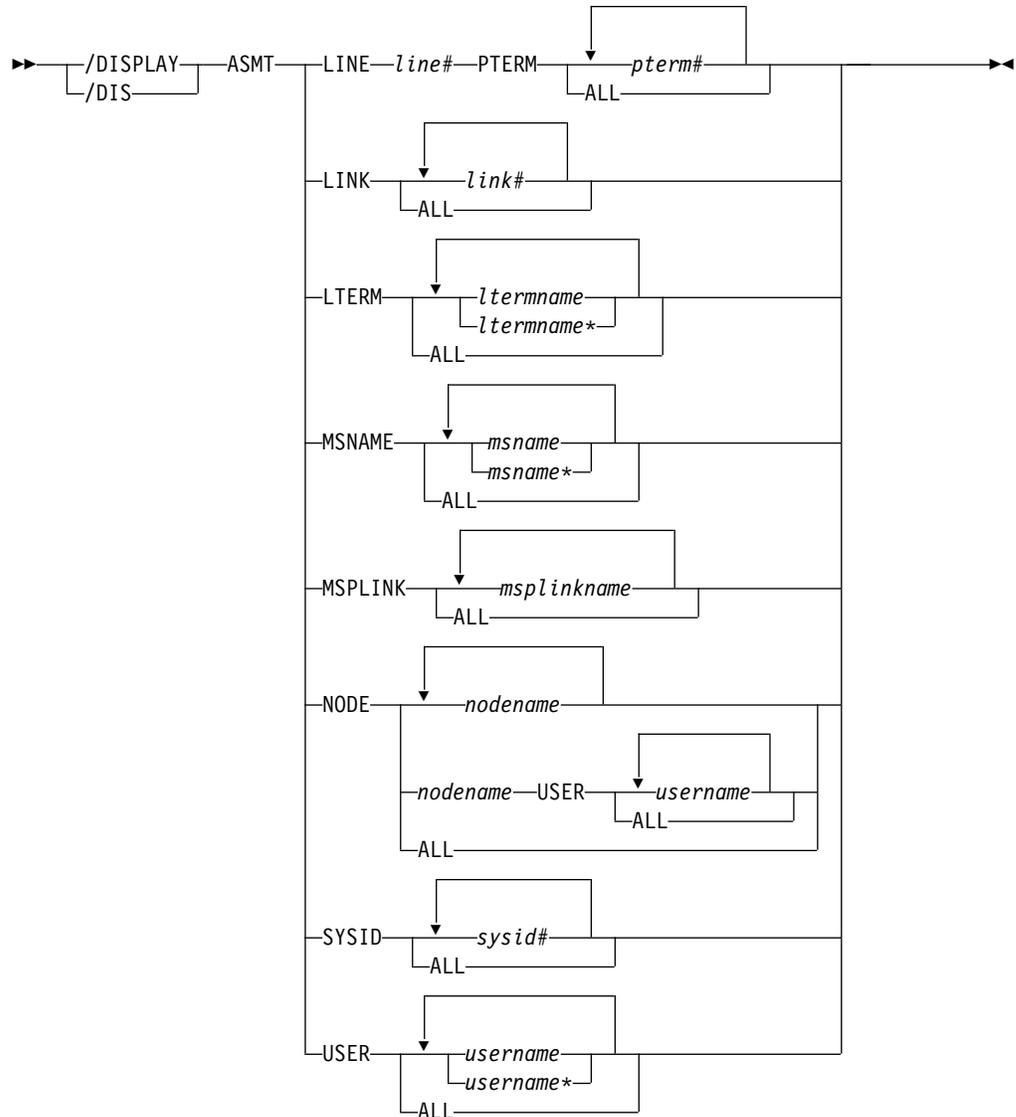
## Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

*Table 275. Valid environments for the /DISPLAY ASMT command and keywords*

Command / Keywords	DB/DC	DBCTL	DCCTL
/DISPLAY	X	X	X
ASMT	X		X
LINE	X		X
LINK	X		X
LTERM	X		X
MSNAME	X		X
MSPLINK	X		X
NODE	X		X
SYSID	X		X
USER	X		X

## Syntax



## Keywords

The following keywords are valid for the `/DISPLAY ASMT` command:

### LINE PTERM

The logical terminal names associated with the specified line and physical terminal.

**LINK** The physical link, SYSIDs, and logical link paths assigned to the specified logical link.

### LTERM

The communication line and physical terminal or node and component associated with the specified logical terminal name. When using ISC or a dynamic terminal, the displayed output for a logical terminal assigned to the VTAM pool shows the assigned node, if any, and the user name. The LTERM parameter can be generic.

**MSNAME**

The physical link and logical link assigned to the specified logical link path.

**MSPLINK**

The type, address, and logical link associated with the specified physical link.

For MSC with TCP/IP or VTAM, MSPLINK also displays the maximum number of allowed sessions and the identifier used by the remote IMS system. For TCP/IP physical links, the command output displays the IMS ID of the remote IMS system. For VTAM physical links, the output displays the VTAM node name of the remote IMS system.

If logons are stopped for a physical link that is used for MSC TCP/IP generic resources, the status of the link displays as PSTOPGEN.

**NODE**

The logical terminal names associated with the specified node. When using ISC, the displayed output shows at least one line of information for each half-session of the node, shows whether the half-session is active, and shows whether LTERMs are assigned. If the half-session is active, the user identifier is displayed. If no LTERMs are assigned, NONE is indicated. The USER keyword is valid for ISC nodes and non-ISC nodes with signed-on users. The NODE parameter can be generic.

**Restrictions for using NODE and USER parameters together:**

- Commands with the NODE USER keyword pair are valid only if:
  - The USER is signed on to the NODE
  - In an ISC environment, the USER is allocated to the NODE
  - The nodes and users exist
- /DISPLAY ASMT commands with NODE USER keyword pair are valid for ISC and non-ISC nodes and users.

**SYSID**

The physical link, logical link, and logical link path assignments associated with the specified system identification.

**USER** All the USER structures and USERIDs that match the parameter specified. These are the following:

- The USER field shows the USER structure name. N/A appears for all static USERIDs signed on to static nodes.
- The USERID field displays the RACF USERID that was used to sign on to the node. N/A appears if a USER structure is not associated with a node. In this case, the USER structure exists to hold status or has messages inserted to it. If no messages or status exist, the USER structure is deleted at the next checkpoint.
- The node name and logical terminal names are displayed if the user is allocated to a node or signed on to a node.
- For ISC sessions, the ID field shows the other half session qualifier.

The USER parameter can be generic.

**Equivalent IMS type-2 commands**

The following table shows variations of the /DISPLAY ASMT command and the IMS type-2 commands that perform similar functions.

Table 276. Type-2 equivalents for the /DISPLAY ASMT command.

Task	/DISPLAY ASMT command	Similar IMS type-2 command
Displays input and output components.	/DISPLAY ASMT LTERM <i>ltermname</i>	QUERY LTERM SHOW(COMPONENT)
Displays the node associated with the lterm.	/DISPLAY ASMT LTERM <i>ltermname</i>	QUERY LTERM SHOW(NODE)
Displays the user associated with the lterm.	/DISPLAY ASMT LTERM <i>ltermname</i>	QUERY LTERM SHOW(USER)
Displays the attributes of an MSC logical link	/DISPLAY ASMT MSPLINK <i>msplinkname</i>	QUERY MSLINK SHOW(ALL)
Displays the attributes of an MSC physical link	/DISPLAY ASMT MSPLINK <i>msplinkname</i>	QUERY MSPLINK SHOW(ALL)
Displays assigned lterms.	/DISPLAY ASMT NODE <i>node</i>	QUERY NODE SHOW(LTERM)
Displays the user associated with the node.	/DISPLAY ASMT NODE <i>node</i>	QUERY NODE SHOW(USER)
Displays the other half-session qualifier ID of the ISC node.	/DISPLAY ASMT USER <i>user</i>	QUERY USER SHOW(ID)
Displays assigned lterms.	/DISPLAY ASMT USER <i>user</i>	QUERY USER SHOW(LTERM)
Displays the node.	/DISPLAY ASMT USER <i>user</i>	QUERY USER SHOW(NODE) QUERY USERID SHOW(NODE)
Displays the user ID.	/DISPLAY ASMT USER <i>user</i>	QUERY USER SHOW(USERID)
Displays the user.	/DISPLAY ASMT USER <i>user</i>	QUERY USERID SHOW(USER)

## Examples

The following are examples of the /DISPLAY ASMT command.

### Example 1 for /DISPLAY ASMT command

Entry ET:

```
/DISPLAY ASMT LINE 2 PTERM ALL
```

Response ET:

```
TERMINAL
  2- 1 IN - L2740S2
      OUT - L2740S1, L2740S2
*90295/130910*
```

Entry ET:

```
/DISPLAY ASMT LINE 4 PTERM 1, 2
```

Response ET:

```
TERMINAL
  4- 1 IN - L2740SM1
      OUT- L2740SM1
  4- 2 IN - L2740SM2
      OUT- L2740SM2
*90295/130328*
```

### Example 2 for /DISPLAY ASMT command

Entry ET:

/DISPLAY ASMT LINK ALL

Response ET:

LINK	PLINK	SIDR	SIDL	MSNAME
1	BLCB1CTC	1	4	SYSTEM2A
		1	5	SYSTEM2B
		3	6	FRESNO
2	BLCB2BSC	21	23	BOSTON
		22	24	ALBANY
3	BLCB1MTM	19	20	TEST1

\*90280/072935\*

*Example 3 for /DISPLAY ASMT command*

Entry ET:

/DISPLAY ASMT LTERM DIALQ1 INQUIRY1 L2740S1 MASTER WTOR

Response ET:

LTERM	IN-TERMINAL	OUT-TERMINAL	USER
DIALQ1	11- 1	11- 1	
INQUIRY1	5- 1	5- 1	
L2740S1	3- 1	3- 1	
MASTER	3- 1	3- 1	
WTOR	1- SC	1- SC	

\*90295/130657\*

Entry ET:

/DISPLAY ASMT LTERM L1 L2 L3

Response ET:

LTERM	IN-TERMINAL	OUT-TERMINAL	USER
L1	14- 1	14- 1	
L2	NODEA -1	NODEA -1	A
L3	N/A -1	N/A -1	B

\*90231/143059\*

Entry ET:

/DISPLAY ASMT LTERM ALL

Response ET:

LTERM	IN-TERMINAL	OUT-TERMINAL	USER
CRDPUNCH	14- 1	14- 1	
DIALQ1	11- 1	11- 1	
DIALQ2	12- 1	12- 1	
DIALQ3	13- 1	13- 1	
INQUIRY1	5- 1	5- 1	
INQUIRY2	6- 1	6- 1	
INQUIRY3	7- 1	7- 1	
INQUIRY4	8- 1	8- 1	
INQUIRY5	9- 1	9- 1	
INQUIRY6	10- 1	10- 1	
L2740SM1	4- 1	4- 1	
L2740SM2	4- 2	4- 2	
L2740S1	3- 1	3- 1	
L2740S2	2- 1	2- 1	
MASTER	3- 1	3- 1	
PTPPUNCH	14- 1	14- 1	
WS12DS	WS12- 1	WS12- 1	
WTOR	1- SC	1- SC	

\*90295/160953\*

*Example 4 for /DISPLAY ASMT command*

Entry ET:

```
/DISPLAY ASMT MSNAME ALL
```

Response ET:

LINK	PLINK	SIDR	SIDL	MSNAME	IMS1
	N/A	32	3	LINK23B4	IMS1
10	PLNK12V	20	10	LINK12V1	IMS1
	N/A	35	3	LINK23U2	IMS1
	N/A	28	3	ELINK211	IMS1
1	PLNK12V	31	11	LINK13B3	IMS1
10	PLNK12V	266	2036	LINK13V	IMS1
10	PLNK12V	256	1012	LINK12V	IMS1
2	PLNK12C	33	13	LINK13C2	IMS1
10	PLNK12V	80	300	LINK13X	IMS1
11	PLNK12V	40	9	LINK12V2	IMS1
3	PLNK12M	34	14	LINK13M2	IMS1
12	PLNK13V	38	18	LINK13V2	IMS1
7	N/A	43	44	ELINK124	IMS1
	N/A	36	3	LINK23V1	IMS1
1	PLNK12V	21	11	LINK12B1	IMS1
1	PLNK12V	22	12	LINK12B2	IMS1
10	PLNK12V	90	500	LINK12Y	IMS1
5	N/A	26	16	ELINK120	IMS1
1	PLNK12V	32	12	LINK13B4	IMS1
	N/A	8	3	LINK21V3	IMS1
6	N/A	51	41	ELINK1MS	IMS1
3	PLNK12M	24	14	LINK12M1	IMS1
	N/A	266	3	LINK23V	IMS1
4	N/A	35	15	LINK13U2	IMS1
	N/A	31	3	LINK23B3	IMS1
	N/A	43	3	ELINK224	IMS1
	N/A	80	3	LINK23X	IMS1
	N/A	37	3	LINK23V2	IMS1
6	N/A	27	17	ELINK121	IMS1
	N/A	33	3	LINK23C2	IMS1
4	N/A	25	15	LINK12U1	IMS1
	N/A	34	3	LINK23M2	IMS1
5	N/A	29	19	ELINK122	IMS1
	N/A	29	3	ELINK222	IMS1
2	PLNK12C	23	13	LINK12C1	IMS1
10	PLNK12V	36	16	LINK13V1	IMS1
7	N/A	42	45	ELINK123	IMS1
16	LKV1TSTB	93	3	PTH3TSTB	IMS1

\*97258/170244\*

*Example 5 for /DISPLAY ASMT command*

The following example displays information about MSC physical link assignments. The output is displayed in tabular format. Each logical link assigned to a physical link is displayed on a single row. The attributes of the physical links are displayed in columns.

In the output of the /DISPLAY ASMT MSPLINK command:

**LINK** The link number of the logical link that is assigned to the physical link.

**PLINK**  
The name of the physical link.

**TYPE** The physical link type.

**ADDR**

The address of the physical link. ADDR applies to VTAM and CTC only.

**MAXSESS**

The maximum number of logical links or sessions that can be assigned to the physical link.

**NODE**

The identifier of the remote IMS system.

For VTAM physical links, NODE is the node name of the remote IMS system. For TCP/IP physical links, NODE is the IMSID of the remote IMS system.

**Entry ET:**

```
/DISPLAY ASMT MSPLINK ALL
```

**Response ET:**

DFS000I	LINK	PLINK	TYPE	ADDR	MAXSESS	NODE
DFS000I	3	PLNK12M	MTM	****		
DFS000I	9	PLNK12MA	MTM	****		
DFS000I	2	PLNK12C	CTC	0211		
DFS000I	N/A	PLNK12MU	MTM	****		
DFS000I	N/A	PLNK12CU	CTC	****		
DFS000I	1	PLNK12V	VTAM	00000000	3	L6APPL1
DFS000I	10	PLNK12V	VTAM	00000000	3	L6APPL1
DFS000I	11	PLNK12V	VTAM	00000000	3	L6APPL1
DFS000I	28	PLNK12V	VTAM	00000000	3	L6APPL1
DFS000I	N/A	PLNK12VU	VTAM	****	1	PZ606099
DFS000I	12	PLNK13V	VTAM	00000000	5	APPL7
DFS000I	13	PLNKSON1	VTAM	00000000	1	L6APPLU
DFS000I	14	PLNKSON2	VTAM	00000000	2	L6APPLV
DFS000I	15	PLNKSON3	VTAM	00000000	2	L6APPLW
DFS000I	16	PLNK12MB	MTM	****		
DFS000I	17	PLNK13MB	MTM	****		
DFS000I	18	PLNK12CB	CTC	****		
DFS000I	19	PLNK13CB	CTC	****		
DFS000I	20	PLNK12VB	VTAM	00000000	2	L6APPL1
DFS000I	21	PLNK13VB	VTAM	00000000	2	APPL7
DFS000I	22	PLNK12TA	TCPIP	****	2	IMS2
DFS000I	23	PLNK12TA	TCPIP	****	2	IMS2
DFS000I	24	PLNK13TA	TCPIP	****	2	IMS3
DFS000I	*11130/232948*					

**Example 6 for /DISPLAY ASMT command****Entry ET:**

```
/DISPLAY ASMT NODE NLU1 NLUT65 NLUT6P
```

**Response ET:**

NODE	USER	LTERM
NLU1		I/O- L1 , L2
NLUT65		I/O- L3
NLUT6P	A	I/O- L5 , L6
NLUT6P	N/A	I/O- NONE
NLUT6P	C	I/O- L7
*04008/113126*		

**Entry ET:**

```
/DISPLAY ASMT NODE NLUT6P USER C
```

**Response ET:**

```

NODE      USER      LTERM
NLUT6P    C           I/O- L7
*04008/113126*

```

Entry ET:

```
/DISPLAY ASMT NODE ALL
```

Response ET:

```

NODE      USER      LTERM
CT3275    I/O- VT3275 , VT3275P
CT3277A   I/O- VT3270A
CT3277B   I/O- VT3270B
CT3277C   I/O- VT3270C
CT3277D   I/O- VT3270P1
CT3277E   I/O- VT3270P2
L3270A    I/O- VT3270L1
L3270B    I/O- VT3270L2
L3284A    I/O- VT3270P3
W612      I/O- W610JP , W610D6 , W610FP
W613      I/O- W620JP , W620D6 , W620PB , W620M6
FCBOX1    I/O- W6CIT
*04008/113126*

```

*Example 7 for /DISPLAY ASMT command*

Entry ET:

```
/DISPLAY ASMT SYSID ALL
```

Response ET:

```

LINK PLINK  SIDR SIDL MSNAME
  1 BLCB1CTC  1   4 SYSTEM2A
  1 BLCB1CTC  2   5 SYSTEM2B
  1 BLCB1CTC  3   6 FRESNO
  3 BLCB1MTM 19  20 TEST1
  2 BLCB2BSC 21  23 BOSTON
  2 BLCB2BSC 22  24 ALBANY
*90280/072821*

```

Entry ET:

```
/DISPLAY ASMT SYSID ALL
```

Response ET:

```

LINK PLINK  SIDR SIDL MSNAME      IMS1
  N/A      8   3 LINK21V3  IMS1
  N/A      28  3 ELINK211  IMS1
  5 N/A     29  19 ELINK122  IMS1
  1 PLNK12V 31  11 LINK13B3  IMS1
  1 PLNK12V 32  12 LINK13B4  IMS1
  2 PLNK12C 33  13 LINK13C2  IMS1
  3 PLNK12M 34  14 LINK13M2  IMS1
  4 N/A     35  15 LINK13U2  IMS1
 10 PLNK12V 36  16 LINK13V1  IMS1
  N/A     37   3 LINK23V2  IMS1
 12 PLNK13V 38  18 LINK13V2  IMS1
  7 N/A     43  44 ELINK124  IMS1
  6 N/A     51  41 ELINK1MS  IMS1
 10 PLNK12V 80  300 LINK13X  IMS1
 10 PLNK12V 266 2036 LINK13V  IMS1
*97258/170653*

```

Entry ET:

/DISPLAY ASMT SYSID 22 23 24

Response ET:

```
LINK PLINK SIDR SIDL MSNAME
  2 BLCB2BSC 22 24 ALBANY
  23 IS LOCAL
  24 IS LOCAL
*90280/072902*
```

*Example 8 for /DISPLAY ASMT command*

Entry ET:

/DISPLAY ASMT USER A B C D

Response ET:

```
USER ID NODE
A CICSA NLUT6P I/O- L1 , L2
B I/O- L3
C CICSB NLUT6P I/O- L4
D I/O- NONE
*90229/083059*
```

Entry ET:

/DISPLAY ASMT USER IMSUS01 IMSUS02

Response ET:

```
USER USERID ID NODE
IMSUS01 IMSUS01 DTSLU201 I/O- IMSUS01
N/A IMSUS01 L3270A I/O- T3270LA
N/A IMSUS02 L3270B I/O- T3270LB
DT327002 IMSUS02 DT327002 I/O- DT327002
*92350/112229*
```

Entry ET:

/DISPLAY ASMT USER IMSUS01\*

Response ET:

```
USER USERID ID NODE
IMSUS01C N/A I/O- NONE
IMSUS01D N/A I/O- NONE
IMSUS01 IMSUS01 DTSLU201 I/O- IMSUS01
N/A IMSUS01 L3270A I/O- T3270LA
*92350/113904*
```

Entry ET:

/DISPLAY ASMT USER ALL

Response ET:

```
USER USERID ID NODE
CA13 I/O- CA1SP3T1, CA1SP3T2
IMSUS01C N/A I/O- NONE
IMSUS01D N/A I/O- NONE
LU6SPA N/A LU6NDPF I/O- LU6LTPA1
LU6SPB N/A I/O- LU6LTPB1, LU6LTPB2
DT327002 IMSUS02 DT327002 I/O- DT327002
IMSUS03 IMSUS03 DTSLU601 I/O- IMSUS03
IMSUS01 IMSUS01 DTSLU201 I/O- IMSUS01
IMSUS05 N/A I/O- NONE
```

```

N/A      IMSUS02      L3270B  I/O- T3270LB
N/A      IMSUS01      L3270A  I/O- T3270LA
N/A      IMSUS04      L3270C  I/O- T3270LC
*92350/114845*

```

**Example 9 for /DISPLAY ASMT command: ISC TCP/IP nodes**

In the following examples, the nodes that are shown in the command output are ISC TCP/IP nodes.

Entry ET:

```
/DIS ASMT NODE CICS
```

Response ET:

```

NODE      USER      LTERM
CICS      IMSUS01    I/O- IMSUS01
*12062/122917*  IMS1

```

Entry ET:

```
/DIS ASMT USER IMSUS01
```

Response ET:

```

USER      USERID   ID      NODE
IMSUS01   N/A           IMSUS01 CICS   I/O- IMSUS01
*12062/122949*  IMS1

```

Entry ET:

```
/DIS ASMT LTERM IMSUS01
```

Response ET:

```

LTERM     IN-TERMINAL  OUT-TERMINAL  USER
IMSUS01   CICS        -1           CICS   -1     IMSUS01
*12062/123336*  IMS1

```

**Related reference:**

- [➞ QUERY LTERM command \(Commands\)](#)
- [➞ QUERY NODE command \(Commands\)](#)
- [➞ QUERY USER command \(Commands\)](#)
- [➞ QUERY USERID command \(Commands\)](#)
- [➞ QUERY MSLINK command \(Commands\)](#)
- [➞ QUERY MSPLINK command \(Commands\)](#)

---

## **/DISPLAY CCTL command**

Use the /DISPLAY CCTL command to display all coordinator controllers that are connected to DBCTL along with the pseudorecovery token, recovery token, region ID, PSB name, and status.

**Restriction:** The /DISPLAY CCTL command does not support the ODBA interface to IMS DB. To display status information about units of work for protected resources that are managed by z/OS Resource Recovery Services, use the /DISPLAY UOR command.

Subsections:

- “Environment”
- “Syntax”
- “Keywords”
- “Output fields” on page 577
- “Examples” on page 577

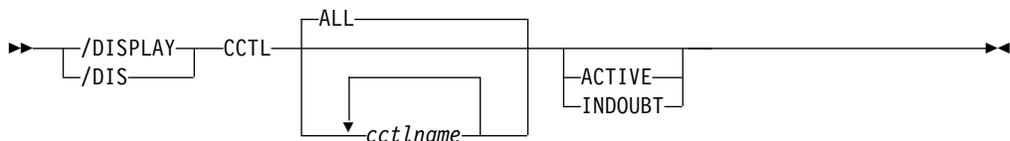
## Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

Table 277. Valid environments for the /DISPLAY CCTL command and keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
/DISPLAY	X	X	X
ACTIVE	X	X	X
CCTL	X	X	
INDOUBT	X	X	

## Syntax



## Keywords

The following keywords are valid for the /DISPLAY CCTL command:

### ALL | *cctlname*

Specifies the coordinator controllers to be displayed.

### ALL

Displays all coordinator controllers that are connected to IMS.

### *cctlname*

Specifies the name of the coordinator controller. You can specify one or more coordinator controller names.

### ACTIVE | INDOUBT

Mutually exclusive, optional keywords that you use to specify the type of threads to be displayed.

### ACTIVE

Displays all currently active threads that are associated with either the specified coordinator controller name (*cctlname*) or with all coordinator controllers.

### INDOUBT

Displays all in-doubt threads that are associated with either the specified coordinator controller name (*cctlname*) or with all coordinator controllers.

## Output fields

Coordinator controllers that are attached to IMS show a status of ATTACHED.

The status of active threads can be any of the following:

- ACTIVE
- ACTIVE-DBCMD
- AVAILABLE
- TERM
- UNAVAIL
- W-BLKMVR
- W-INTENT
- W-POOLSP
- W-SWITCH

The status of in-doubt threads is always displayed as INDOUBT.

## Examples

The following are examples of the /DISPLAY CCTL command.

### *Example 1 for /DISPLAY CCTL command*

Entry ET:

```
/DISPLAY CCTL
```

Response ET:

```
      CCTL           STATUS
      CICS1
      *89067/114857*
```

### *Example 2 for /DISPLAY CCTL command*

Entry ET:

```
/DISPLAY CCTL ALL
```

Response ET:

```
      CCTL      PSEUDO-RTKN  RECOVERY-TOKEN  REGID  PSBNAME  STATUS
      CICS1
                                0000000000000000    3      AVAILABLE
                                0000000000000000    2      AVAILABLE
                                0000000000000000    1      AVAILABLE
      *89067/113236*
```

### *Example 3 for /DISPLAY CCTL command*

Entry ET:

```
/DISPLAY CCTL CICS1
```

Response ET:

CCTL	PSEUDO-RTKN	RECOVERY-TOKEN	REGID	PSBNAME	STATUS
CICS1					ATTACHED
		9FFA9568FF594301	3	BMP255	ACTIVE
		9FFA956B7AE24E00	2	BMP255	ACTIVE
		0000000000000000	1		AVAILABLE

\*89067/113251\*

*Example 4 for /DISPLAY CCTL command*

Entry ET:

/DISPLAY CCTL CICS1 ACTIVE

Response ET:

CCTL	PSEUDO-RTKN	RECOVERY-TOKEN	REGID	PSBNAME	STATUS
CICS1					ATTACHED
		9FFA967C26D69802	2	BMP255	ACTIVE
		9FFA967ACF9EB802	1	BMP255	ACTIVE

\*89067/114557\*

*Example 5 for /DISPLAY CCTL command*

Entry ET:

/DISPLAY CCTL CICS1 INDOUBT

Response ET:

CCTL	PSEUDO-RTKN	RECOVERY-TOKEN	REGID	PSBNAME	STATUS
CICS1					ATTACHED
	000100C0	9FFA956B7AE24E00		BMP255	INDOUBT
	00010040	9FFA9568FF594301		BMP255	INDOUBT

\*89067/113413\*

**Related reference:**

“Status and attributes for the /DISPLAY command” on page 523

---

## **/DISPLAY CONV command**

Use the /DISPLAY CONV command to display all BUSY and HELD conversations in the IMS system if no other keywords or parameters are specified.

Subsections:

- “Environment”
- “Syntax” on page 579
- “Keywords and reserved parameters” on page 579
- “Usage notes” on page 580
- “Equivalent IMS type-2 commands” on page 580
- “Output fields” on page 580
- “Examples” on page 581

### **Environment**

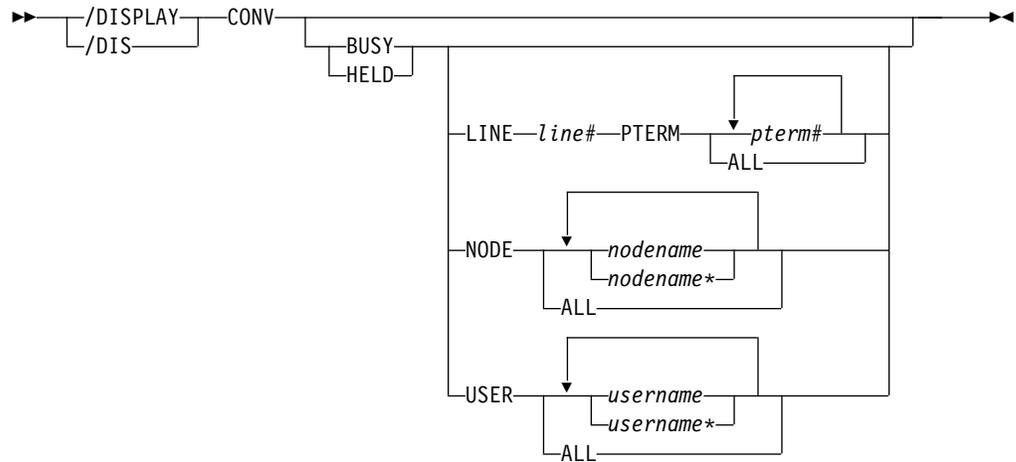
The /DISPLAY CONV command is valid in DB/DC and DCCTL environments.

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

Table 278. Valid environments for the /DISPLAY CONV command and keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
/DISPLAY	X	X	X
CONV	X		X
LINE	X		X
NODE	X		X
USER	X		X

## Syntax



## Keywords and reserved parameters

The following keywords and reserved parameters are valid for the /DISPLAY CONV command:

**BUSY** The conversations currently active for the system or active for a specific node, line and physical terminal, or user.

**HELD** The conversations currently held in the system or held from a specific node, line and physical terminal, or user.

**LINE** All conversations for the specified communication line are displayed.

### NODE

All conversations for the specified node are displayed. If the NODE keyword is used and the node is an ISC node, information for all applicable sessions of the parallel session node is displayed. For dynamic terminals, the conversations of the currently associated user are displayed. The NODE parameter can be generic.

### PTERM

All conversations for the specified physical terminal are displayed.

**USER** All conversations that apply to the specified user are displayed. The USER parameter can be generic.

## Usage notes

/DISPLAY CONVERSATION is accepted as a valid command from an IMS terminal because CONVERSATION is a valid synonym of CONV.

When ISC is used with parallel sessions, one line of information for each session in a conversation is displayed. The user name is also displayed.

**Attention:** In an IMSplex, use of the /DISPLAY CONV command without a specific resource name will result in extensive accesses to the Resource Manager for global information, and its use should be carefully considered.

For an IMS conversational transaction issued from an LU 6.2 application program, the /DISPLAY CONV command shows the LU name and user ID (if available) of the LU 6.2 conversation along with the IMS conversation ID and status.

For an IMS conversational transaction issued from an OTMA client, the /DISPLAY CONV command shows the TMEMBER and TPIPE name in addition to the IMS conversation ID and status.

On an IMS XRF alternate system, where TMEMBER and TPIPE name information is not available, the literal 'UNKNOWN.OTMA' is displayed instead of the TMEMBER and TPIPE names associated with the conversation.

## Equivalent IMS type-2 commands

The following table shows variations of the /DISPLAY CONV command and the IMS type-2 commands that perform similar functions.

Table 279. Type-2 equivalents for the /DISPLAY CONV command

Task	/DISPLAY CONV command	Similar IMS type-2 command
Displays nodes with active or held conversations.	/DISPLAY CONV	QUERY NODE STATUS(CONV)
Displays users with active or held conversations.	/DISPLAY CONV	QUERY USER STATUS(CONV)
Displays IMS conversation information for a particular node.	/DISPLAY CONV NODE <i>node</i>	QUERY NODE SHOW(CONV)
Displays IMS conversation information for a particular user.	/DISPLAY CONV USER <i>user</i>	QUERY USER SHOW(CONV)

## Output fields

The following output fields are displayed for the /DISPLAY CONV command:

**ID** A 4-byte conversation ID, which is unique for each user and terminal.

### STATUS

Conversation status. The status can be ACTIVE, HELD, or SCHEDULED.

### TERMINAL

Physical device with which the conversation is associated. This can be a non-VTAM device/pterm, a VTAM node, an ISC node, an APPC luname, or an OTMA tmember/tpipe.

**USER** The user signed on to a dynamic VTAM node, or the subpool associated with an ISC node, or the user ID for an LU 6.2 conversation.

## Examples

Entry ET:

```
/DISPLAY CONV
```

Response ET:

```
TERMINAL USER  ID STATUS  
NO CONVERSATIONS  
*91113/182917*
```

Entry ET:

```
/DISPLAY CONV
```

Response ET:

```
TERMINAL USER  ID STATUS  
11-  2          0001 HELD  
4-   2          0002 ACTIVE, SCHEDULED  
11-  4          0001 HELD  
2-   1          0001 HELD  
5-   1          0002 ACTIVE  
*91115/135706*
```

Entry ET:

```
/DISPLAY CONV
```

Response ET:

```
TERMINAL USER  ID STATUS  
LUNAME1 USERID1 0001 ACTIVE  
DT327001 IMSUS01 0005 ACTIVE  
LUNAME2          0002 ACTIVE, SCHEDULED  
LUNAME3 USERID4 0003 ACTIVE, SCHEDULED  
LUNAME1 USERID4 0004 SCHEDULED  
*90332/114253*
```

Entry ET:

```
/DISPLAY CONV BUSY
```

Response ET:

```
TERMINAL USER  ID STATUS  
NO BUSY CONVERSATIONS  
*91113/182959*
```

Entry ET:

```
/DISPLAY CONV BUSY
```

Response ET:

```
TERMINAL USER  ID STATUS  
2-   1          0001 ACTIVE, SCHEDULED  
4-   2          0010 ACTIVE, SCHEDULED  
4-   1          0011 ACTIVE, SCHEDULED  
*91113/192021*
```

Entry ET:

```
/DISPLAY CONV BUSY LINE 4 PTERM ALL
```

Response ET:

```

TERMINAL USER  ID STATUS
4- 2          0001 ACTIVE, SCHEDULED
4- 1          0002 ACTIVE, SCHEDULED
*91113/192101*

```

Entry ET:

```
/DISPLAY CONV BUSY
```

Response ET:

```

TERMINAL      USER      ID STATUS
NETWORK1.LUNAME1  USERID1  0001 ACTIVE
DT327001      IMSUS01  0005 ACTIVE
NID2.LUNAME2          0002 ACTIVE,SCHEDULED
LUNAME3        USERID4  0003 ACTIVE,SCHEDULED
NETWORK1.LUNAME1  USERID5  0004 ACTIVE
*92232/114253*

```

Entry ET:

```
/DISPLAY CONV HELD
```

Response ET:

```

TERMINAL USER  ID STATUS
NO HELD CONVERSATIONS
*91113/183022*

```

Entry ET:

```
/DISPLAY CONV HELD NODE NSLUTP6
```

Response ET:

```

TERMINAL USER  ID STATUS
NSLUTP6 SPOOLA 0007 HELD
NSLUTP6 SPOOLC 000A HELD
*91113/183033*

```

**Related reference:**

- [➞ QUERY NODE command \(Commands\)](#)
- [➞ QUERY USER command \(Commands\)](#)

---

## **/DISPLAY CPLOG command**

Use the /DISPLAY CPLOG command to display the value of the IMS execution parameter, CPLOG. CPLOG specifies the number of system log records between system-generated checkpoints.

Subsections:

- "Environment"
- "Syntax" on page 583

### **Environment**

The /DISPLAY CPLOG command is valid in DB/DC, DBCTL, and DCCTL environments.

## Syntax

▶▶ `/DISPLAY CPLOG`  
    └─┬─┘  
      └─┘  
      `/DIS`

---

## /DISPLAY CQS command

Use the `/DISPLAY CQS` command to display information about the Common Queue Server (CQS), including the CQS job name, version number, and current status. Valid status conditions are `CONNECTED` and `DISCONNECTED`.

Subsections:

- “Environment”
- “Syntax”
- “Examples”

### Environment

The `/DISPLAY CQS` command is valid in `DB/DC` and `DCCTL` environments.

This command is valid only in a shared-queues environment.

### Syntax

▶▶ `/DISPLAY CQS`  
    └─┬─┘  
      └─┘  
      `/DIS`

### Examples

Entry ET:

```
/DISPLAY CQS
```

Response ET:

```
JOBNAME    VERS#  STATUS  
CQS1       1.1   CONNECTED  
*95200/170817*
```

Explanation: IMS is connected to the Common Queue Server, CQS1.

---

## /DISPLAY DB command

Use the `/DISPLAY DB` command to display the status of the specified databases. If a specified database is a `DEDB`, the associated `DEDB` areas are also displayed.

The `/DISPLAY DB` command can be used on `HALDB` databases.

Subsections:

- “Environment” on page 584
- “Syntax” on page 584
- “Keywords and reserved parameters” on page 584
- “Usage notes” on page 586
- “Equivalent IMS type-2 commands” on page 587

- “Output fields” on page 587
- “Examples” on page 589

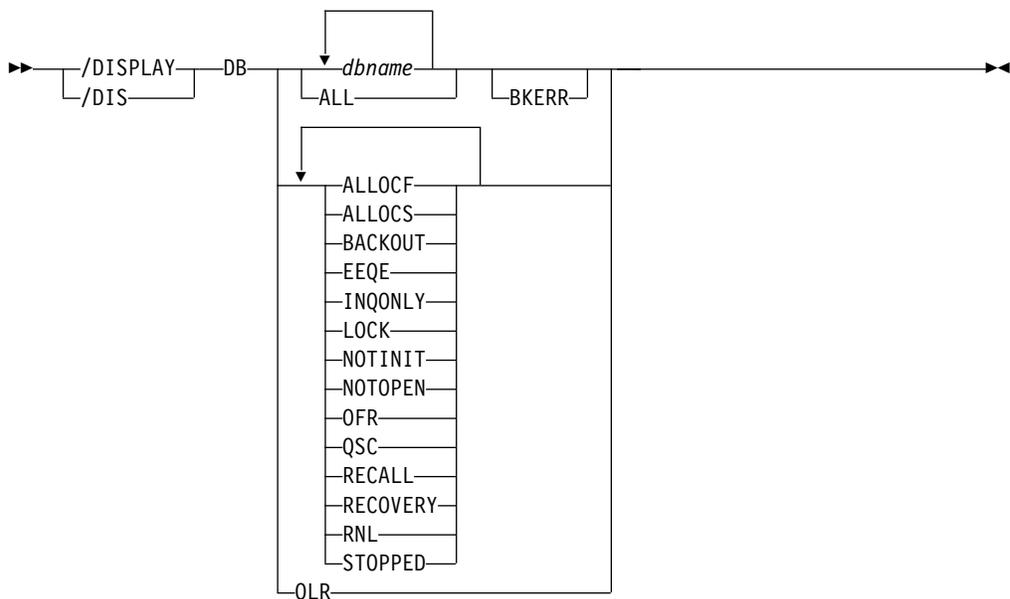
## Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

Table 280. Valid environments for the /DISPLAY DB command and keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
/DISPLAY	X	X	X
DB	X	X	
BKERR	X	X	

## Syntax



## Keywords and reserved parameters

The following keywords can be specified for the /DISPLAY DB command:

### BKERR

Displays the error queue elements associated with a DL/I or DEDB database. It also displays incomplete backouts preventing the use of a DL/I database.

### OLR

The /DISPLAY DB OLR command displays the status of a specified partition locally; that is, the status is only from the IMS where the command was issued. The /DISPLAY DB OLR command returns OLR information and status of all partitions that have HALDB online reorganization (OLR) cursor-active status. The following information is returned:

**RATE** The rate at which the HALDB OLR is running, from 1 to 100

**BYTES**

The total number of bytes moved to the output data sets

**SEGMENT**

The total number of segments moved to the output data sets

**ROOTS**

The total number of roots moved to the output data sets

**STARTTIME**

The local time the OLR was first started

**STATUS**

HALDB OLR status. One or more of the following statuses are returned:

- RUNNING - OLR is running.
- WAITLOCK - OLR is waiting for a lock.
- WAITRATE - OLR is waiting because of an intentional delay. This intentional delay occurred because a value of less than 100 was specified for the RATE parameter.
- OPTDEL - Output data sets are deleted at the end of OLR.
- OPTNODEL - Output data sets are not deleted at the end of OLR.
- RESUMED - OLR is resumed at this IMS after being stopped for some reason such as a TERMINATE OLREORG command or a user abend.
- OPTREL - Ownership of the OLR is released if IMS terminates.
- OPTNOREL - Ownership of the OLR is retained if IMS terminates.
- NOTOWNED - OLR is not owned by this IMS.

The /DISPLAY DB OLR command can be issued to obtain the above information for an OLR that has been stopped for some reason such as a TERMINATE OLREORG command or a user abend. The data associated with the terminated OLR is maintained and provided under any of the following conditions:

- The TERMINATE OLR command is issued.
- Abnormal OLR termination occurs (for example, DFS2971W message is issued)
- IMS is normally shut down.

**Note:** If the partition or HALDB master has been taken offline with a /DBR command, or if IMS has been restarted to resume the OLR and the OLR ownership is requested before IMS restart, the status will not be available through the /DISPLAY DB OLR command until the OLR is resumed. The status, however, can be obtained through the LIST.DB or LIST.RECON command.

The OLR attribute cannot be specified on the /DISPLAY DB command with any other attribute such as STOPPED or ALLOC. If additional attributes are specified, the command is rejected and message, DFS0135 is displayed.

If a /DISPLAY DB OLR command is issued on an RSR tracker or an XRF alternate, the message, DFS132I - ALL PARAMETERS ARE INVALID is displayed.

Attributes are reserved parameters for the /DISPLAY DB command and cannot be used to name databases. The attributes in the following list can be used with the DB keyword. Databases are selected for display based on the attribute or attributes

specified. The attribute display format is the same as the standard DB display. Attributes usually correspond to the conditions displayed for databases.

**ALLOCF**

Displays the databases whose allocation failed.

**ALLOCS**

Displays the databases whose allocation was successful.

**BACKOUT**

Displays the databases with incomplete backouts, which prevent the use of the databases.

**EEQE**

Displays the databases that are associated with one or more error queue elements.

**INQONLY**

Displays the databases for which a /DBDUMP command was issued.

**LOCK**

Displays the databases that are locked.

**NOTINIT**

Displays the databases for which directory initialization failed.

**NOTOPEN**

Displays the databases that are not in open status. This status is not applicable to MSDBs.

**OFR**

Displays the databases that are currently being recovered by online forward recovery (OFR).

**Note:** The OFR attribute is valid only when issued from an RSR tracking subsystem.

**OLR**

Displays the databases that have online reorganization in progress.

**Restriction:** The OLR attribute cannot be entered with any other attribute.

**QSC**

Displays the databases that have the status of quiesced or quiescing.

**RECALL**

Displays the databases that are in recall.

**STOPPED**

Displays the databases that are stopped.

## Usage notes

If you use the /DISPLAY command to identify OLRs that are terminated using the TERMINATE OLREORG command, it does not report status for partitions that have not been authorized. Alternately, use the commands QUERY OLREORG STATUS (NOTOWNED) with ROUTE(\*) on the command request, and /RMLIST DBRC='DB DBD(partname)'. If all systems on the PLEX show status NOTOWNED, and the DBRC database record shows an OLR active cursor, then OLR has been terminated.

When you enter this command, the database name can be an existing non-HALDB, a HALDB master, or a HALDB partition. A command against a HALDB partition operates exactly like a command against a non-HALDB except for the /START DB and the UPDATE DB START(ACCESS) command. A HALDB partition is not allocated during the command unless it was previously authorized but not allocated, the OPEN keyword was specified, or the partition has EEQEs. The partition is allocated at first reference.

The HALDB partition reflects conditions such as STOPPED, LOCKED, or NOTOPEN. When a HALDB partition is stopped, it must be explicitly started again. Commands with the keyword ALL and commands against a HALDB master do not change the STOPPED and LOCKED indicators in each HALDB partition.

When the command target is a HALDB master, processing acts on all HALDB partitions. For example, if the IMS command is /DBR on the HALDB master, all of the HALDB partitions are closed, deallocated, and deauthorized. Only the HALDB master displays STOPPED (each HALDB partition does not display STOPPED unless it was itself stopped). If a /DBR command was issued against a HALDB master, the display output of a /DISPLAY DB command shows the HALDB master (as STOPPED), but does not display the status of the partitions.

Each partition inherits the access limitations of its HALDB master. If the /DBD command is issued against a HALDB master, all of its partitions close. A subsequent reference to any of the partitions results in the partition opening for input, although the partition's access might be UPDATE or EXCLUSIVE. The DBRC authorization state reflects the limited access.

## Equivalent IMS type-2 commands

The following table shows variations of the /DISPLAY DB command and the IMS type-2 commands that perform similar functions.

Table 281. Type-2 equivalents for the /DISPLAY DB command

Task	/DISPLAY DB command	Similar IMS type-2 command
Displays the status of a database.	/DISPLAY DB <i>dbname1 ...dbnamen</i>   ALL	QUERY DB

## Output fields

The following output fields are displayed for the /DISPLAY DB command:

**DB** Name of the database.

**TYPE** Type of database:

- DL/I
- DEDB
- DHISNDX (refers to Fast Path secondary index databases)
- MSNR (refers to MSDB non-related databases)
- MSRD (refers to MSDB-related dynamic databases)
- MSRF (refers to MSDB-related fixed databases)
- PHIDAM
- PHDAM
- PSINDEX

- PART
- blank (if the database status is NOTINIT)

**TOTAL SEQ DEPEND**

Total control intervals defined for sequential dependent space, which applies to areas associated with DEDBs. This column appears only if Fast Path is installed.

**UNUSED SEQ DEPEND**

Unused control intervals for sequential dependent space. This column appears only if Fast Path is installed.

**TOTAL DIRECT ADDRESS**

Total control intervals for independent overflow part of direct addressable space, which applies to areas associated with DEDBs. This column appears only if Fast Path is installed.

**UNUSED DIRECT ADDRESS**

Total unused control intervals for independent overflow part of a direct addressable space, which applies to areas associated with DEDBs. This column appears only if Fast Path is installed. This IOVF count is updated on a timed basis, with a default timer of 2 hours.

**ACC** Type of access to the database, which can be one of the following:

- RO** Read only
- RD** Read
- UP** Update
- EX** Exclusive

**CONDITIONS**

Status conditions of the database, which can be one or more of the following:

- ALLOCF
- ALLOCS
- BACKOUT
- EEQE
- INQONLY
- LOCK
- NOTINIT
- NOTOPEN
- OFR
- OLR
- QUIESCED
- QUIESCING
- RECALL
- RECOVERY
- RNL
- STOPPED

The /DISPLAY DB command with the BKERR keyword displays error queue elements associated with a DL/I database and displays incomplete backouts.

The following output fields are displayed by the /DISPLAY DB command with the BKERR keyword:

**DATABASE**

Name of the database with error queue elements or backout pending data.

**ERROR DD**

DDNAME of the data set with the error queue element

**TYPE** Type of error queue element, which is one of the following:

**BACK PSB**

PSB needing backout

**BLOCK**

OSAM block number or VSAM CI RBA

**DATE** Date

**IDT** In-doubt

**IDT+WRT**

In-doubt and write error

**IDT+IOT**

In-doubt and I/O toleration

**IOT** I/O toleration

**IOT/VSO**

I/O toleration for VSO areas

**PRM** DBRC permanent

**RD** Read error

**TIME** Time

**USE** DBRC user created

**WRT** Write error

**BLOCK**

The /DISPLAY DB command with BKERR does not display individual RBAs for I/O tolerated non-SDEP CIs of a DEDB area defined with the Virtual Storage Option (VSO). Instead, a single entry with a value of 00000001 is displayed; the EEQE type is set to IOT/VSO. I/O tolerated SDEP CIs for the area are displayed individually by RBA.

The EEQE information might not be current if the database is registered with DBRC and is not open, because IMS updates EEQE information from DBRC when the database is opened.

**Examples**

The following are examples of the /DISPLAY DB command.

*Example 1 for /DISPLAY DB command*

Entry ET:

```
/DISPLAY DB ALL
```

Response ET:

DATABASE	TYPE	TOTAL	UNUSED	TOTAL	UNUSED	ACC	CONDITIONS
DD41M702	DL/I					EX	NOTOPEN
DD41M803	DL/I					EX	NOTOPEN
DEDBJN21	DEDB	SEQ	DEPEND	DIRECT	ADDRES	EX	NOTOPEN
DB21AR0	AREA	N/A	N/A	N/A	N/A		NOTOPEN
DB21AR1	AREA	N/A	N/A	N/A	N/A		NOTOPEN
DB21AR2	AREA	N/A	N/A	N/A	N/A		NOTOPEN
DB21AR3	AREA	N/A	N/A	N/A	N/A		NOTOPEN
DB21AR4	AREA	N/A	N/A	N/A	N/A		NOTOPEN
DB21AR5	AREA	N/A	N/A	N/A	N/A		NOTOPEN
DB21AR6	AREA	N/A	N/A	N/A	N/A		NOTOPEN
DB21AR7	AREA	N/A	N/A	N/A	N/A		NOTOPEN
DB21AR8	AREA	N/A	N/A	N/A	N/A		NOTOPEN
DB21AR9	AREA	N/A	N/A	N/A	N/A		NOTOPEN
DB21AR10	AREA	N/A	N/A	N/A	N/A		NOTOPEN
DB21AR11	AREA	N/A	N/A	N/A	N/A		NOTOPEN
DEDBJN22	DEDB	SEQ	DEPEND	DIRECT	ADDRES	EX	NOTOPEN
DB22AR0	AREA	N/A	N/A	N/A	N/A		NOTOPEN
DB22AR1	AREA	N/A	N/A	N/A	N/A		NOTOPEN
DEDBJN23	DEDB	SEQ	DEPEND	DIRECT	ADDRES	EX	NOTOPEN
DB23AR0	AREA	N/A	N/A	N/A	N/A		NOTOPEN
DB23AR1	AREA	N/A	N/A	N/A	N/A		NOTOPEN
DIMSRN01	DL/I					EX	NOTOPEN
DIMSRN02	DL/I					EX	NOTOPEN
DIMSRN03	DL/I					EX	NOTOPEN

\*89184/142639\*

Entry ET:

/DISPLAY DB DD41M803 BKERR

Response ET:

```

DATABASE
DD41M803
  ERROR DD  TYPE      BLOCK
  DD41M803  IOT      0000003F
*90135/161902*

```

Explanation: An I/O toleration error queue element exists for database DD41M803 on a newly created active system in an XRF environment.

Following an /UNLOCK SYSTEM command on the above system, the /DISPLAY DB DD41M803 BKERR command would yield the following display:

```

DATABASE
DD41M803
  NO EEQE OR INCOMPLETE BACKOUT INFORMATION AVAILABLE
*90135/163500*

```

*Example 2 for /DISPLAY DB command*

Entry ET:

/DIS DB BE3ORDER BE3PARTS.

Response ET:

DATABASE	TYPE	TOTAL	UNUSED	TOTAL	UNUSED	ACC	CONDITIONS
BE3ORDER	DL/I					EX	STOPPED, LOCK, NOTOPEN, RECALL
BE3PARTS	DL/I					EX	NOTOPEN, RECALL

\*94277/124039\*

*Example 3 for /DISPLAY DB command*

Entry ET:  
/DIS DB RECALL.

Response ET:

DATABASE	TYPE	TOTAL UNUSED	TOTAL UNUSED	ACC	CONDITIONS
BE3ORDER	DL/I			EX	STOPPED, LOCK, NOTOPEN, RECALL
BE3PARTS	DL/I			EX	NOTOPEN, RECALL
IVPDB1	DL/I			UP	STOPPED, LOCK, NOTOPEN, RECALL
IVPDB2	DL/I			UP	NOTOPEN, RECALL

\*94277/124119\*

*Example 4 for /DISPLAY DB command*

Entry ET:  
/DISPLAY DB DEDBJN21 BKERR

Response ET:

DATABASE	ERROR DD	TYPE	BLOCK
DEDBJN21	DB21AR0	IOT	00015000
	DB21AR0	IOT/VSO	00000001
	DB21AR1	IOT	00054000
	DB21AR1	IOT/VSO	00000001

Explanation: The /DISPLAY DB command with the BKERR keyword does not display individual RBAs for I/O tolerated non-SDEP CIs of a DEDB area defined with the Virtual Storage Option (VSO). Instead, a single entry with a value of 00000001 is displayed and the EEQE type is set to IOT/VSO. I/O tolerated SDEP CIs for the area are displayed individually by RBA.

*Example 5 for /DISPLAY DB command*

If a full function database data set is undergoing recovery with Online Recovery Service, the output of the /DISPLAY DB command includes this information in the command response.

Entry ET:  
/DISPLAY DB DD41M702

Response ET:

DATABASE	TYPE	TOTAL UNUSED	TOTAL UNUSED	ACC	CONDITIONS
DD41M702	DL/I			EX	STOPPED, LOCK, NOTOPEN RECOVERY

\*97184/142639\*

*Example 6 for /DISPLAY DB command*

Display the HALDB master for the PHIDAM database called PARTMAST.

Entry ET:  
/DIS DB PARTMAST

Response ET:

DATABASE	TYPE	TOTAL UNUSED	TOTAL UNUSED	ACC	CONDITIONS
PARTMAST	PHIDAM			UP	
PART1	PART			UP	ALLOCS
PART2	PART			UP	NOTOPEN
PART3	PART			UP	STOPPED

\*99166/092514\*

*Example 7 for /DISPLAY DB command*

Display the HALDB partition for PHIDAM PART2.

Entry ET:

/DIS DB PART2

Response ET:

DFS000I	DATABASE	TYPE	TOTAL UNUSED	TOTAL UNUSED	ACC	CONDITIONS
DFS000I	PARTMAST	PHIDAM			UP	
DFS000I	PART2	PART			UP	NOTOPEN

\*99166/092737\*

*Example 8 for /DISPLAY DB command*

Display output for an online reorganization with ownership retention. If ownership is not retained during a termination, OPTREL is replaced with OPTNOREL.

Entry ET:

/DIS DB OLR

Response ET:

DATABASE	PART	RATE	BYTES	SEGS	ROOTS	STARTTIME
STATUS						
DBHDOJ01	PDHDOJB	1	0	0	0	07295/143354
WAITRATE, OPTNODEL						
DBHDOJ01	PDHDOJA	1	3330	17	1	07295/143354
WAITRATE, OPTDEL, OPTREL						
DBHDOJ01	PDHDOJC	1	1400	5	2	07295/143354
WAITLOCK, OPTNODEL						

\*07295/143358\*

Explanation: Issue the /DISPLAY DB OLR command to obtain information about all HALDB online reorganizations that are running in cursor-active status. The following information is returned:

- Master database name
- Partition name
- OLR rate
- Number of bytes moved
- Number of segments moved
- Number of roots moved
- Start time of the OLR when it was first started
- Status of the progress of OLR
- OLR inactive data set option
- Date/Time stamp when the command was processed
- OLR options, including whether OLR ownership is released if IMS terminates before completing the reorganization
- OLR wait status, if any

**Note:** For partitions in NOTOWNED status, if the OLR statistics information is unavailable, only the master database name, partition name, and status are displayed.

*Example 9 for /DISPLAY DB command*

Entry ET:

/DISPLAY DB QSC

Response ET:

DATABASE	TYPE	TOTAL UNUSED	TOTAL UNUSED	ACC	CONDITIONS
AUTODB	DL/I			UP	NOTOPEN, QUIESCING
DBHIDJ03	PHIDAM			UP	
POHIDJC	PART			UP	NOTOPEN, QUIESCED
DBOHIDK5	PHIDAM			UP	QUIESCED
POHIDKA	PART			UP	NOTOPEN, ALLOCS, QUIESCED
POHIDKB	PART			UP	NOTOPEN, QUIESCED
POHIDKC	PART			UP	NOTOPEN, QUIESCED
POHIDKD	PART			UP	NOTOPEN, QUIESCED
DBOVLFPC	DL/I			UP	NOTOPEN, QUIESCING

Explanation: Issue the /DISPLAY DB QSC command to obtain information about all databases that have the status of quiesced or quiescing.

*Example 10 for /DISPLAY DB command*

In this example, TYPE is set to DHISNDX for a HISAM or a SHISAM secondary index database.

Entry ET:

/DISPLAY DB ALL

Response ET:

DATABASE	TYPE	TOTAL UNUSED	TOTAL UNUSED	ACC	CONDITIONS	IMS1		
DBHDOJ01	PHDAM			UP		IMS1		
PDHDOJA	PART			UP	NOTOPEN	IMS1		
PDHDOJB	PART			UP	NOTOPEN	IMS1		
PDHDOJC	PART			UP	NOTOPEN	IMS1		
PDHDOJD	PART			UP	NOTOPEN	IMS1		
DEDBJN24	DEDB	SEQ	DEPEND	DIRECT	ADDRES	UP	NOTOPEN	IMS1
DB24A000	AREA	N/A	N/A	N/A	N/A	NOTOPEN	IMS1	
DB24A001	AREA	N/A	N/A	N/A	N/A	NOTOPEN	IMS1	
DB24A239	AREA	N/A	N/A	N/A	N/A	NOTOPEN	IMS1	
DEHSJX24	DHISNDX					UP	NOTOPEN	IMS1
*09043/155658*								IMS1

Explanation: DBHDOJ01 is a PHDAM database with four partitions: PDHDOJA, PDHDOJB, PDHDOJC, and PDHDOJD. DEDBJN24 is a DEDB database with three areas: DB24A000, DB24A001, and DB24A239. DEHSJX24 is a HISAM or a SHISAM secondary index database. The local access of each database is UPDATE.

*Example 11 for /DISPLAY DB command*

Entry ET:

/DISPLAY DB OFR

Response ET:

DATABASE	TYPE	TOTAL UNUSED	TOTAL UNUSED	ACC	CONDITIONS
BE2PCUST	DL/I			EX	ALLOCS OFR
BE3ORDER	DL/I			EX	ALLOCS OFR
BE3ORDRX	DL/I			EX	ALLOCS OFR
BE3PARTS	DL/I			EX	ALLOCS OFR
BE3PSID1	DL/I			EX	ALLOCS OFR
*91240/132406*		SYS3			

**Related reference:**

“Status and attributes for the /DISPLAY command” on page 523

-  Parameter descriptions for IMS procedures (System Definition)
-  QUERY DB command (Commands)

## /DISPLAY DBD command

Use the /DISPLAY DBD command to display, for databases that are being accessed, their type, the PSBs accessing them, and the type of access.

Subsections:

- “Environment”
- “Syntax”
- “Equivalent IMS type-2 commands”
- “Examples”

### Environment

The /DISPLAY DBD command is valid in DB/DC and DBCTL environments.

### Syntax



### Equivalent IMS type-2 commands

The following table shows variations of the /DISPLAY DBD command and the IMS type-2 commands that perform similar functions.

Table 282. Type-2 equivalents for the /DISPLAY DBD command

Task	/DISPLAY DBD command	Similar IMS type-2 command
Displays the programs being accessed by databases.	/DISPLAY DBD <i>dbname</i>	QUERY DB NAME( <i>dbname</i> ) SHOW(PGM)

### Examples

Entry ET:

```
/DISPLAY DBD MSDBLM01 MSDBLM02 MSDBLM03 MSDBLM04
```

Response ET:

DBD-NAME	TYPE	PSB-NAME	ACCESS
MSDBLM01	MSDB	DDLTLMO6	R/W
		DDLTLMO7	R/W
MSDBLM02	MSDB	DDLTLMO6	R/W
		DDLTLMO7	R/W
MSDBLM03	MSDB	DDLTLMO6	R/W
		DDLTLMO7	R/W
MSDBLM04	MSDB	DDLTLMO6	R/W
		DDLTLMO7	R/W

\*91068/110958\*

**Related reference:**

 [QUERY DB command \(Commands\)](#)

## /DISPLAY DESC command

Use the /DISPLAY DESC command to display LU 6.2 descriptors. All specified values are displayed, as are defaults/blanks for values not specified in the descriptor.

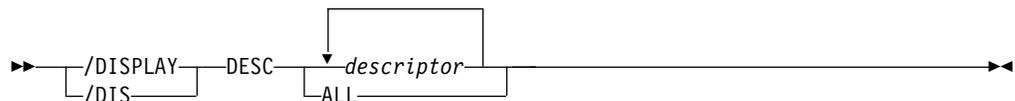
Subsections:

- “Environment”
- “Syntax”
- “Output fields”
- “Examples” on page 596

### Environment

The /DISPLAY DESC command is valid in DB/DC and DCCTL environments.

### Syntax



### Output fields

The following output fields are displayed for the /DISPLAY DESC command:

**DESC** Name of the descriptor.

**LUNAME**

LU name value in the descriptor. The network-qualified LU name is optional for the LUNAME keyword.

**MODE**

VTAM LOGON mode table entry value in the descriptor.

**OUTBNDLU**

Shows which local-LU will be used for APPC outbound processing when the descriptor is used.

**SIDE** APPC z/OS side table entry value in the descriptor.

**SYNCLEVEL**

APPC synchronous level in the descriptor; either NONE or CONFIRM.





**OPTION**

A data space option (which can be DREF) or an area option (which can be PREL or PRE0). Area options are defined to DBRC with the INIT.DBDS or CHANGE.DBDS command.

**STRUCTURE**

Name of the coupling facility structure allocated for this area.

**ENTRIES**

Number of area CIs in the structure.

**CHANGED**

Number of area CIs in the structure that have been updated. IMS writes these updated CIs to DASD (and resets this value) when you enter any of the following commands:

- /CHECKPOINT
- /STOP AREA
- /VUNLOAD

**AREACI#**

Total number of area CIs that can be loaded into the structure.

**POOLNAME**

Name of the private buffer pool used by the area.

**Examples**

Entry ET:

```
/DISPLAY FPV
```

Response ET:

DATASPACE	MAXSIZE(4K)	AREANAME	AREASIZE(4K)	OPTION	SYS3
001	524188				
		DB21AR1		76 PRE0	
		DB21AR11		152	
000	524188			DREF	
		DB21AR0		19 PRE0, PREL	
		DB21AR10		19 PRE0, PREL	
		DB21AR2		38 PRE0, PREL	
*93076/173254*					

Entry ET:

```
/DISPLAY FPV
```

Response ET:

DATASPACE	MAXSIZE(4K)	AREANAME	AREASIZE(4K)	OPTION	SYS3
000	524263			DREF	
		DB21AR4		167 PRE0, PREL	
		DB21AR2		42 PRE0, PREL	
001	524263				
		DB21AR3		84 PRE0	
		DB21AR1		84	
CF	POOL	CACHE	AREANAME	AREASIZE(4K)	OPTION
CF1	4K	N	AREAFR01	1000	PRE0, PREL
CF2	512	Y	AREA2	100	PRE0, PREL
*95225/173254*					

Entry ET:

```
/DISPLAY FPV
```

Response ET:

```
DATASPACE MAXSIZE(4K) AREANAME AREASIZE(4K) OPTION      SYS3
      000      524238                DREF      SYS3
                DB21AR1                19 PRE0, PREL
                DB21AR0                19 PRE0, PREL
                DB23AR1                19 PRE0, PREL
      001      524238                                SYS3
NO AREAS LOADED INTO NORM DATASPACE 001. SYS3
      002      524238                                SYS3
NO AREAS LOADED INTO NORM DATASPACE 002. SYS3
AREANAME STRUCTURE ENTRIES CHANGED AREA CI# POOLNAME OPTIONS  SYS2
NO BLOCK LEVEL SHARED AREAS EXIST.      SYS3
*03129/145840*  SYS3
```

Entry ET:

```
/DISPLAY FPV
```

Response ET:

```
AREANAME STRUCTURE          ENTRIES CHANGED AREA CI# POOLNAME OPTIONS
DB33FR01 DB33FR01STRUCTUR 0000150 0000018 00000150 FR01      PRE0, PREL
DB21AR9  MAS1KSTR          0000043 0000043 00000270 MAS1K      PRE0
DB21AR8  MAS1KSTR          0000044 0000044 00000150 MAS1K      PRE0
DD01AR0  DD01AR0STR1      0000045 0000025 00000045 DD01      PRE0, PREL
DB21AR10 MAS512STR         0000084 0000084 00000150 V$$$512  PRE0
DB21AR3  MAS2KSTR          0000150 0000015 00000150 MAS2K      PRE0, PREL
DB21AR1  MAS2KSTR          0000150 0000126 00000150 MAS2K      PRE0, PREL
DB21AR0  MAS512STR         0000150 0000001 00000150 V$$$512  PRE0, PREL
DB21AR4  MAS4KSTR          0000010 0000010 00000150 MAS4K      PRE0
DB21AR11 MAS4KSTR          0000010 0000010 00000150 MAS4K      PRE0
DB21AR5  MAS1KSTR          0000044 0000044 00000150 MAS1K      PRE0
DB21AR6  MAS512STR         0000077 0000077 00000150 V$$$512  PRE0
DB21AR7  MAS1KSTR          0000040 0000040 00000150 MAS1K      PRE0
DB21AR2  MAS1KSTR          0000044 0000044 00000150 MAS1K      PRE0
```

Explanation: Each area using a multi-area structure is listed individually with its own statistics. Column AREACI# displays the total number of CIs in the root addressable part. This is the total number of CIs loaded into the structure if the entire Area was preloaded.

---

## **/DISPLAY HSB command**

Use the /DISPLAY HSB command to display the system-related information in an Extended Recovery Facility (XRF) environment.

If the command is entered on the active system, information about the active system is displayed. If the command is entered on the alternate system, information about both systems is displayed.

Subsections:

- "Environment"
- "Syntax" on page 600
- "Output fields" on page 600
- "Examples" on page 601

### **Environment**

The /DISPLAY HSB command is valid in DB/DC and DCCTL environments.

## Syntax



## Output fields

The following output fields are displayed for the /DISPLAY HSB command:

### RSENAME

Recoverable service element name.

### STATUS

System entering command (ACTIVE or BACKUP).

### MODETBL

Indication of whether (IOT) or not (blank) the system is in I/O toleration mode. This field is displayed only if the command is entered on the active system.

### PHASE

Displays only if the command is entered on the alternate system; indicates the phase of the alternate system. The following phases can be displayed:

**INIT** Initialization phase

**SYN** Processing SNAPQ checkpoint

**TRK** Tracking active system

**TKO** Taking over workload from active

### IMS-ID

IMS identifier of system entering command.

### VTAM UVAR

VTAM generic name.

### ACT-ID

IMS identifier of active system; field is displayed only if the command is entered on the alternate system.

### LOG-TIME

Time associated with the log record currently being read by the alternate system in the process of tracking the active system. The length of time the alternate system lags behind the active system in reading the log is the difference between the current time, as shown by the time stamp, and the log time. Some of this time difference might be caused by differences in the synchronization of the clocks of the processors.

### SURVEILLANCE

Indicates one of the following surveillance mechanisms:

**LNK** IMS ISC link

**LOG** IMS system log

**RDS** IMS restart data set

### INTERVAL

The surveillance interval, in seconds, for the specific surveillance mechanism

## TIMEOUT

The surveillance timeout value, in seconds, for the specific surveillance mechanism

## STATUS

The status of the specific surveillance mechanism (ACTIVE or INACTIVE).

## TAKEOVER CONDITIONS

The following takeover conditions may apply:

### ALARM

The alarm on the alternate processor will be sounded when takeover is requested.

### AUTO

Automatic takeover is in effect.

Following the display line of ALARM, AUTO, or both, the takeover conditions specified on the SWITCH control statement of the DFSHSBxx PROCLIB member are shown. (xx is the SUF= parameter of the IMS procedure.) There are two groups per line and each group is separated by an asterisk. Each group can include one or more of the following:

### VTAM

VTAM TPEND exit

**LNK** IMS ISC link

**LOG** IMS system log

**RDS** IMS restart data set

## Examples

Entry ET (Active System):

```
/DISPLAY HSB
```

Response ET:

RSENAME	STATUS	MODE	IMS-ID	VTAM UVAR
DFSRSENM	ACTIVE		IMSA	USERVAR1
SURVEILLANCE	INTERVAL	TIMEOUT	STATUS	
LOG	10	99	ACTIVE	
LNK	3	9	ACTIVE	
RDS	10	99	ACTIVE	

\*89340/094236\*

Entry ET (Active System - MNPS environment):

```
/DISPLAY HSB
```

Response ET:

RSENAME	STATUS	MODE	IMS-ID	MNPS NAME
DFSRSENM	ACTIVE		IMSA	USERVAR1
SURVEILLANCE	INTERVAL	TIMEOUT	STATUS	
LOG	10	99	ACTIVE	
LNK	3	9	ACTIVE	
RDS	10	99	ACTIVE	

\*89340/094236\*

Entry ET (Alternate System):

```
/DISPLAY HSB
```

Response ET:

RSENAME	STATUS	PHASE	IMS-ID	VTAM UVAR	ACT-ID	LOG-TIME
DFSRSENM	BACKUP	TRK	IMSB	USERVAR1	IMSA	09:42:56
	BACKUP SYSTEM			ACTIVE SYSTEM		
SURVEILLANCE	INTERVAL	TIMEOUT	STATUS	INTERVAL	TIMEOUT	STATUS
LOG	10	99	INACTIVE	10	99	INACTIVE
LNK	3	9	INACTIVE	3	9	INACTIVE
RDS	10	99	INACTIVE	10	99	INACTIVE
TAKEOVER CONDITIONS - ALARM						
VTAM			*RDS LOG			
*89340/094256*						

Entry ET (Alternate System - MNPS environment):

/DISPLAY HSB

Response ET:

RSENAME	STATUS	PHASE	IMS-ID	MNPS NAME	ACT-ID	LOG-TIME
DFSRSENM	BACKUP	TRK	IMSB	USERVAR1	IMSA	09:42:56
	BACKUP SYSTEM			ACTIVE SYSTEM		
SURVEILLANCE	INTERVAL	TIMEOUT	STATUS	INTERVAL	TIMEOUT	STATUS
LOG	10	99	INACTIVE	10	99	INACTIVE
LNK	3	9	INACTIVE	3	9	INACTIVE
RDS	10	99	INACTIVE	10	99	INACTIVE
TAKEOVER CONDITIONS - ALARM						
VTAM			*RDS LOG			
*89340/094256*						

---

## /DISPLAY HSSP command

Use the /DISPLAY HSSP command to display information about the current HSSP (high-speed sequential processing) activities.

Subsections:

- "Environment"
- "Syntax"
- "Output fields"
- "Examples" on page 603

### Environment

The /DISPLAY HSSP command is valid in DB/DC and DBCTL environments.

### Syntax

►► [ /DISPLAY ] HSSP —————►►  
    [ /DIS ]

### Output fields

The following output fields are displayed for the /DISPLAY HSSP command:

**RGN** Region ID as displayed in the /DISPLAY ACTIVE command (three digit PST number)

**TYP** Region type (BMP only)

**JOBNAME**  
Job name

**PSB** Program Specification Block name

**DB** Database name

### **AREAS-OPTIONS**

Information on the SETO statement:

**AREA** Area name

**PCB** Information on whether the named area has updated intent.

### **PROCOPT**

Whether PROCOPT H is specified.

**IC** Image Copy in process.

**UPD** Information on whether the area is updated, which is one of the following:

**N** The PCB has no update intent on the named area.

**Y** The PCB has update intent on the named area.

### **OPTION**

Information on the execution

**IC** Image Copy in process

## **Examples**

Entry ET (Active System):

```
/DISPLAY HSSP
```

Response ET:

```
RGN TYP JOBNAME PSB DATABASE AREAS-OPTIONS SYS3
  1 BMP CSSP020B BMPFPE02 DEDBJN21 IC=(1,C) SYS3
*89122/094325* SYS3
```

---

## **/DISPLAY LINE command**

The `/DISPLAY LINE` command, when followed by a valid line number or `ALL`, displays the status and queue counts for the specified communication line or lines and physical terminal.

The queue count for `/DISPLAY LINE` is reset to zero after an IMS cold start.

Subsystems:

- "Environment"
- "Syntax" on page 604
- "Keywords" on page 604
- "Reserved parameters" on page 604
- "Usage notes" on page 606
- "Output fields" on page 606
- "Examples" on page 606

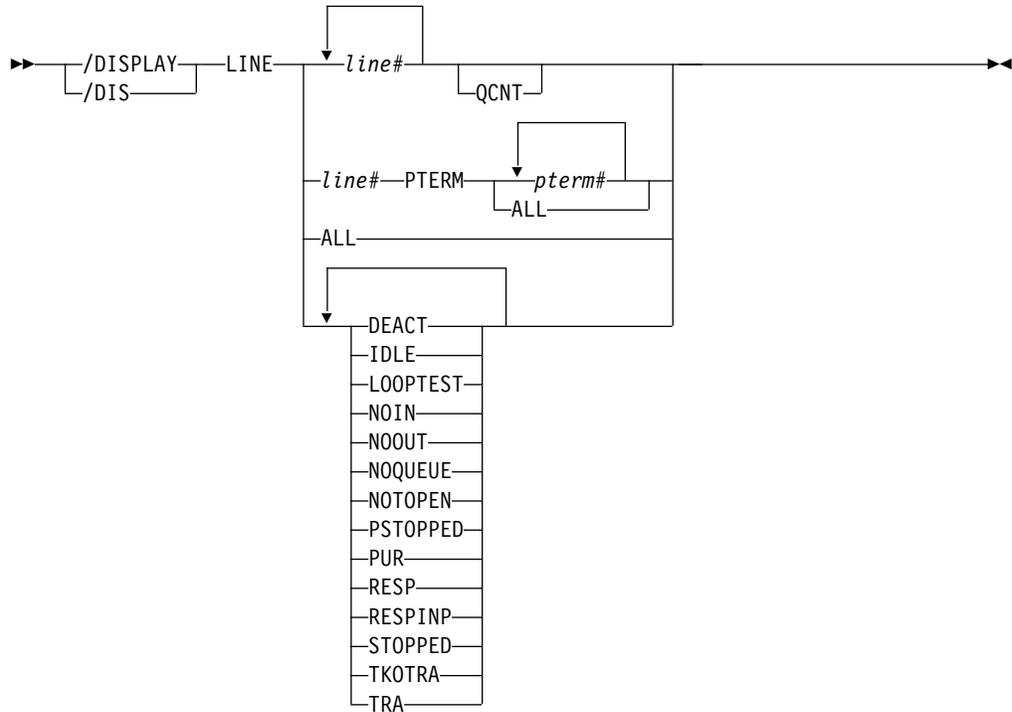
## **Environment**

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

Table 283. Valid environments for the /DISPLAY LINE command and keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
/DISPLAY	X	X	X
LINE	X		X
QCNT	X		X
PTERM	X		X

## Syntax



## Keywords

The following keywords are valid for the /DISPLAY LINE command:

### QCNT

Specifies that global queue count information is to be displayed. If QCNT is not specified, local queue counts are displayed; when it is specified, no local queue counts are displayed.

This keyword is valid only in a shared-queues environment.

### PTERM

Is used to refer to a relative physical terminal. Valid parameters are number corresponding to the relative position of the terminal on the line. When used without the LINE keyword on the /DISPLAY command, valid parameters are physical terminal attributes.

## Reserved parameters

Lines are selected for display based on the attributes specified. Attributes that can be used with the LINE keyword are shown in the following table. The attribute

display format is the same as the standard LINE display. The attributes correspond to the conditions displayed for lines. For example, /DISPLAY LINE LOOPTEST displays all lines that are currently in looptest mode.

Table 284. /DISPLAY LINE command attributes

DEACT	IDLE	LOOPTEST
NOIN	NOOUT	NOQUEUE
NOTOPEN	PSTOPPED	PUR
RESP	RESPINP	STOPPED
TKOTRA	TRA	

**DEACT**

Displays the lines that have been permanently deactivated. Message DFS2473 in the system console log might contain information regarding the reason this status was set. DFS2473 can occur more than once in the system console log.

**IDLE**

Displays the lines for which no activity of any kind is in progress for them.

**LOOPTEST**

Displays the lines that are in looptest mode.

**NOIN**

Displays the lines that are stopped for input.

**NOOUT**

Displays the lines that are stopped for output.

**NOQUEUE**

Displays the lines that have had message queuing stopped for message switching.

**NOTOPEN**

Displays the lines that are not in open status.

**PSTOPPED**

Displays the lines that have been stopped by the /PSTOP command. This status might indicate that a different, necessary resource is unavailable.

**PUR**

Displays the lines that are purging all of their output.

**RESP**

Displays the lines that are in input response mode and the response reply message is available for output or in the process of being sent.

**RESPINP**

Displays the lines that are in full-function input response mode and the response mode input is still in-doubt; for example, the response reply message is not available for output.

**STOPPED**

Displays the lines that are stopped.

**TKOTRA**

Displays the lines that are to be traced only during takeover to help diagnose XRF terminal switch problems.

**TRA**

Displays the lines that are being traced.

## Usage notes

In a shared-queues environment, use the QCNT keyword to obtain the current message queue counts on the shared queues structure. The local queue counts (ENQCT, DEQCT, and QCT) value that is returned on this command when the QCNT keyword is not specified represents the messages being processed by the IMS system where this command is issued. Do not use local queue counts for queue evaluation purposes.

## Output fields

The following output fields are displayed by the /DISPLAY LINE command.

- LIN/PTE
- TYPE
- ADDR
- RECD
- ENQCT
- DEQCT
- QCT
- SENT
- Status
  - IDLE
  - NOTOPEN
  - STOPPED
  - COMPINOP

## Examples

Entry ET:

```
/DISPLAY LINE 1, 3, 10
```

Response ET:

LINE	TYPE	ADDR	RECD	ENQCT	DEQCT	QCT	SENT
1	CONSOLE	****	0	1	0	1	6
3	3270 R	0C9	43	51	43	8	79

\*85098/141118\*

Entry ET:

```
/DISPLAY LINE ALL
```

Response ET:

LINE	TYPE	ADDR	RECD	ENQCT	DEQCT	QCT	SENT
1	CONSOLE	****	4	1	1	0	6
2	3270 R	0C9	43	51	43	8	79
3	3270 L	****	0	0	0	0	0 STOPPED IDLE
4	3270 L	1234	20	22	20	2	0
5	RDR/PTR	****	0	0	0	0	0 STOPPED IDLE
6	RDR/PTR	****	0	0	0	0	0 STOPPED IDLE
7	RDR/PTR	****	0	0	0	0	0 STOPPED IDLE
8	RDR/PTR	****	0	0	0	0	0 STOPPED IDLE
9	DIAL	POOL		0	0	0	POOL STOPPED IDLE

\*85098/141244\*

Entry ET:

```
/DISPLAY LINE 5 PTERM ALL
```

Response ET:

```

LIN/PTE TYPE ADDR RECD ENQCT DEQCT QCT SENT
5- 1 3270 R 404081 3 6 6 0 8
5- 2 3270 R C14081 0 6 6 0 7
5- 3 3270 R C1C181 40 30 28 2 61
5- 4 3270 R C1C281 0 3 0 3 0 COMPINOP
5- 5 3270 R C1C381 0 3 0 3 0 COMPINOP
5- 6 3270 R C1C481 0 3 3 0 3
*85098/141412*

```

Entry ET:

```
/DISPLAY LINE PSTOPPED, PUR
```

Response ET:

```

LINE TYPE ADDR RECD ENQCT DEQCT QCT SENT PUR IDLE NOTOPEN
4 3270 R **** 4 8 8 0 33 PUR IDLE NOTOPEN
18 3270 R 022 0 4 4 0 4 PSTOPPED IDLE
*85098/141283*

```

---

## /DISPLAY LINK command

The /DISPLAY LINK command, when followed by a valid link number or ALL, displays the status and queue counts for the specified logical link.

The queue count for the /DISPLAY LINK command is reset to zero after an IMS cold start.

In addition, the /DISPLAY LINK command shows if the logical link is assigned to a physical link and the partner identification that is assigned to it during system definition.

Subsections:

- "Environment"
- "Syntax" on page 608
- "Keywords" on page 608
- "Usage notes" on page 608
- "Output fields" on page 608
- "Examples" on page 609

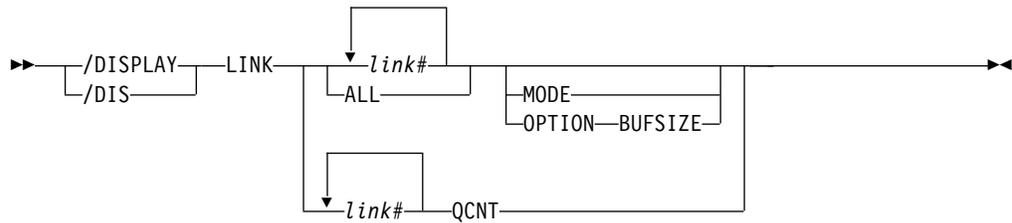
### Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

*Table 285. Valid environments for the /DISPLAY LINK command and keywords*

Command / Keywords	DB/DC	DBCTL	DCCTL
/DISPLAY	X	X	X
LINK	X		X
MODE	X		X
OPTION	X		X
QCNT	X		X

## Syntax



## Keywords

The following keywords are valid for the /DISPLAY LINK command:

### MODE

Displays various modes in which VTAM terminals can operate.

### OPTION

Specifies the following:

### BUFSIZE

Displays the link parameters such as the link number, link name, bandwidth, buffer size, and the link status.

### QCNT

Specifies that global queue count information is to be displayed. If QCNT is not specified, local queue counts are displayed; when it is specified, no local queue counts are displayed.

This count represents the message counts of messages that can be processed by the IMS system where the command is issued. This count includes messages that can be processed by any IMS system (messages with no affinity), plus messages that can be processed by the IMS system where the command is issued (messages with an affinity to the IMS system that issued the command).

This keyword is valid only in a shared-queues environment.

## Usage notes

The /DISPLAY LINK link# MODE command, when entered from the primary IMS system, displays in the ACT MODETABL field the mode table name from the /RESTART command or LOGON exit. The same field does not display on the secondary IMS system.

**Note:** N/A is displayed by /DISPLAY LINK ALL MODE for non-VTAM MSC links.

In a shared-queues environment, use the QCNT keyword to obtain the current message queue counts on the shared queues structure. The local queue counts (ENQCT, DEQCT, and QCT) value that is returned on this command when the QCNT keyword is not specified represents the messages being processed by the IMS system where this command is issued. Do not use local queue counts for queue evaluation purposes.

## Output fields

The following output fields are displayed by the /DISPLAY LINK command:

**ACT MODETBL**

Mode table name actually used to initiate the session. This name is displayed only if the MODE keyword is specified and only while the session is active. The field is blank at normal session termination.

**BUFSIZE**

Link buffer size. The link buffer size is displayed only if the OPTION BUFSIZE keyword is specified.

**DEF MODETBL**

Default mode table name set by system definition or /CHANGE command. This name can be overridden with the /RSTART command or, for non-IMS session initiations, the LOGON exit. This name is displayed only if the MODE keyword is specified.

**DEQCT**

Count of messages dequeued.

**ENQCT**

Count of messages enqueued.

**LINK** Logical link number.

**LINKNAME**

Name of the logical link. The link name is displayed only if the OPTION BUFSIZE keyword is specified on the /DISPLAY LINK command.

**PARTNER**

The identification specified during IMS system definition for this multiple system partnership.

**QCT** The current count of messages queued.

**RECD** Count of messages received.

**SENT** Count of messages sent.

**STATUS**

Displays the link status.

**Examples**

The following are example of the /DISPLAY LINK command.

**Example 1 for /DISPLAY LINK command**

Entry ET:

/DISPLAY LINK ALL

Response ET:

LINK	PARTNER	RECD	ENQCT	DEQCT	QCT	SENT	
1	AB	0	0	0	0	0	PSTOPPED IDLE COLD
2	AC	8	21	12	9	12	NOTIDLE-C0C ACTV TRA
3	AD	4	6	6	0	6	IDLE ACTV TRA
4	BC	0	0	0	0	0	PSTOPPED IDLE COLD
5	AE	0	3	0	3	0	PSTOPPED IDLE

\*91293/171240\*

**Example 2 for /DISPLAY LINK command**

Entry ET:

/DISPLAY LINK 1 3

Response ET:

LINK PARTNER	RECD	ENQCT	DEQCT	QCT	SENT
1 AB	0	1	0	1	0 IDLE
3 AD	0	0	0	0	0 PSTOPPED IDLE

\*90280/072548\*

*Example 3 for /DISPLAY LINK command*

Entry ET:

/DISPLAY LINK ALL MODE

Response ET:

LINK PARTNER	DEF	MODETBL	ACT	MODETBL
1 AB	N/A		N/A	
2 AC	N/A		N/A	
3 AD	N/A		N/A	
4 AE	N/A		N/A	
5 AK	MSC12V			
6 AL	MS12V			

\*90179/102238\*

Explanation: N/A is displayed in the mode table name fields of non-VTAM MSC links. MSC12V and MS12V are mode table names defined at system definition or established using the /CHA command.

*Example 4 for /DISPLAY LINK command*

Entry ET:

/DISPLAY LINK 10 OPTION BUFSIZE

Response ET:

DFS000I	LINK	LINKNAME	BUFSIZE	STATUS
DFS000I	10	LNK12V02	4096	IDLE ACTV PRI FORCE TRA BANDWIDTH

\*06108/153639\*

Explanation: The link buffer size of 4096 and the bandwidth status are displayed.

*Example 5 for /DISPLAY LINK command*

The following two examples show the output of the /DISPLAY LINK command when a link uses TCP/IP as the physical link type. TCP/IP link types always use MSC bandwidth mode, so in the second example, BANDWIDTH indicates that the bandwidth mode is used by the link.

Entry ET:

/DIS LINK 4

Response ET:

DFS000I	LINK	PARTNER	RECD	ENQCT	DEQCT	QCT	SENT		
DFS000I	4	TA	0	0	0	0	0	PSTOPPED	IDLE COLD

Entry ET:

/DIS LINK 4 OPTION BUFSIZE

Response ET:

DFS000I	LINK	LINKNAME	BUFSIZE	STATUS
DFS000I	4	LNK12T01	8192	PSTOPPED IDLE COLD BANDWIDTH

---

## /DISPLAY LTERM command

Use the /DISPLAY LTERM command to display status and queue counts for the specified logical terminal name.

Subsections:

- “Environment”
- “Syntax”
- “Keywords”
- “Usage notes” on page 612
- “Equivalent IMS type-2 commands” on page 612
- “Output fields” on page 613
- “Examples” on page 613

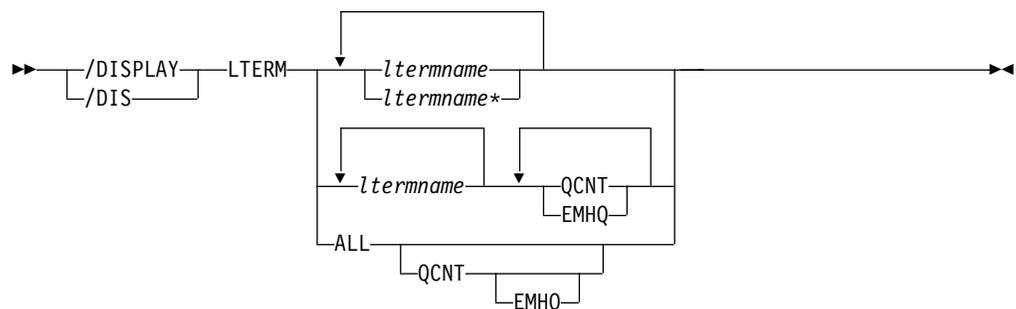
### Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

Table 286. Valid environments for the /DISPLAY LTERM command and keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
/DISPLAY	X	X	X
EMHQ	X		X
LTERM	X		X
QCNT	X		X

### Syntax



### Keywords

The following keywords can be specified for the /DISPLAY LTERM command:

#### EMHQ

Specifies that global queue count information on the Expedited Message Handler Queues (EMHQ) displays when both EMHQ and QCNT are specified. If EMHQ is specified, QCNT must also be specified. This keyword is only valid in a shared-queues environment.

#### QCNT

Specifies that global queue count information is to be displayed. If QCNT is not specified, local queue counts are displayed; when it is specified, no local

queue counts are displayed. If QCNT is specified and EMHQ is not, global queue count information on the shared message queues (MSGQ) is displayed.

This count represents the message counts of messages that can be processed by the IMS system where the command is issued. This count includes messages that can be processed by any IMS system (messages with no affinity), plus messages that can be processed by the IMS system where the command is issued (messages with an affinity to the IMS system that issued the command).

IMS does not check if the specified name is a valid LTERM in the IMS subsystem on which the command is entered.

The /DISPLAY LTERM ALL QCNT command displays all of the LTERMs on the shared queues with a global queue count.

This command does not display queue counts for Fast Path output messages for the NODE unless EMHQ and QCNT are specified.

This keyword is only valid in a shared-queues environment.

For IMS subsystems in a shared-queues environment with the time control option, TCO, enabled, AFFIN has been added to the queue names of TCO LTERMs. When the /DISPLAY LTERM DFSTCFI QCNT command is issued, the queue count of messages for the DFSTCFI LTERM pertains to the TCO LTERM on the local IMS subsystem. The queue count is zero prior to adding AFFIN to the queue names.

## Usage notes

The LTERM parameter can be generic where the generic parameter specifies lterms that already exist.

/DISPLAY LTERM ALL does not display lterms in alphabetical order.

In a shared-queues environment, there are certain situations that cause an LTERM message to be retrieved from the global queue and moved to the local queue, but the message has not been delivered. If this occurs, the global queue count is zero and you need to display the local queue to see the output message.

In a shared-queues environment, use the QCNT keyword to obtain the current message queue counts on the shared queues structure. The local queue counts (ENQCT, DEQCT, and QCT) value that is returned on this command when the QCNT keyword is not specified represents the messages being processed by the IMS system where this command is issued. Do not use local queue counts for queue evaluation purposes.

## Equivalent IMS type-2 commands

The following table shows variations of the /DISPLAY LTERM command and the IMS type-2 commands that perform similar functions.

*Table 287. Type-2 equivalents for the /DISPLAY LTERM command.*

Task	/DISPLAY LTERM command	Similar IMS type-2 command
Displays logical link path for remote lterms.	/DISPLAY LTERM <i>ltermname</i>	QUERY LTERM SHOW(MSNAME)
Displays the lterm message queue count.	/DISPLAY LTERM <i>ltermname</i> /DISPLAY LTERM <i>ltermname</i> QCNT	QUERY LTERM SHOW(QCNT)

Table 287. Type-2 equivalents for the /DISPLAY LTERM command (continued).

Task	/DISPLAY LTERM command	Similar IMS type-2 command
Displays status of the lterm.	/DISPLAY LTERM <i>ltermname</i>	QUERY LTERM SHOW(STATUS)
Displays queue count in the EMH queues.	/DISPLAY LTERM <i>ltermname</i> QCNT EMHQ	QUERY LTERM SHOW(EMHQ)

## Output fields

The status displayed can be one of the following:

- LOCK
- PSTOP
- PUR
- QERR
- QLOCK
- STATIC
- STOP

## Examples

The following are examples of the /DISPLAY LTERM command.

### Example 1 for /DISPLAY LTERM command

Entry ET:

```
/DISPLAY LTERM DIALQ1 INQUIRY1 L2740S1 L2740SM1 MASTER WTOR
```

Response ET:

```
LTERM      ENQCT      DEQCT      QCT
DIALQ1      0           0           0
INQUIRY1    0           0           0
L2740S1     1           1           0  STOP
L2740SM1    1           0           1  STOP
MASTER     12          12          0
WTOR        0           0           0
*90295/123755*
```

### Example 2 for /DISPLAY LTERM command

Entry ET:

```
/DISPLAY LTERM ALL
```

Response ET:

```
LTERM      ENQCT      DEQCT      QCT
DSPLY1     0           0           0
DSPLY2     0           0           0
DSPLY3     0           0           0
MASTER     8           8           0
WTOR        0           0           0
2740AA1    0           0           0
2740AA2    0           0           0
2740AA3    0           0           0
2740C1     0           0           0
2740C2     0           0           0
2740SM1    2           2           0  STOP
2740SM2    2           2           0  STOP
```

```

2740S1      1      1      0
2740S2      1      1      0 QERR
LU6A        1      1      0 QLOCK
*91276/125448*

```

**Example 3 for /DISPLAY LTERM command**

Entry ET:

```
/DISPLAY LTERM LTERMA LTERM123 ABCD QCNT
```

Response ET:

```

LTERM      GBLQCT
LTERMA    1000000000
LTERM123  25
ABCD      0
*95200/170817*

```

**Example 4 for /DISPLAY LTERM command**

Entry ET:

```
/DISPLAY LTERM EMHQ QCNT
```

(one user is on the queue)

Response ET:

```

LTERM      GBLQCT
FPE0001    1
*98203/145857*

```

**Example 5 for /DISPLAY LTERM command**

Entry ET:

```
/DIS LTERM ALL QCNT EMHQ
```

Response ET:

```

LTERM      GBLQCT  SYS3
IMSUS01    1       SYS3
IMSUS02    1       SYS3
*04310/132048*  SYS3

```

**Related reference:**

“Status and attributes for the /DISPLAY command” on page 523

 [QUERY LTERM command \(Commands\)](#)

## **/DISPLAY LUNAME command**

Use the /DISPLAY LUNAME command to display information about a specific LU 6.2 application program.

After restart and any checkpoint, only LU 6.2 resources with status or messages queued are displayed.

If you use a side information entry name to place messages on queues, specify the side information entry name for the LU name and the character string DFFSIDE for the TP name to display those messages.

Subsections:

- “Environment”
- “Syntax”
- “Keywords”
- “Usage notes” on page 616
- “Output fields” on page 616
- “Examples” on page 617

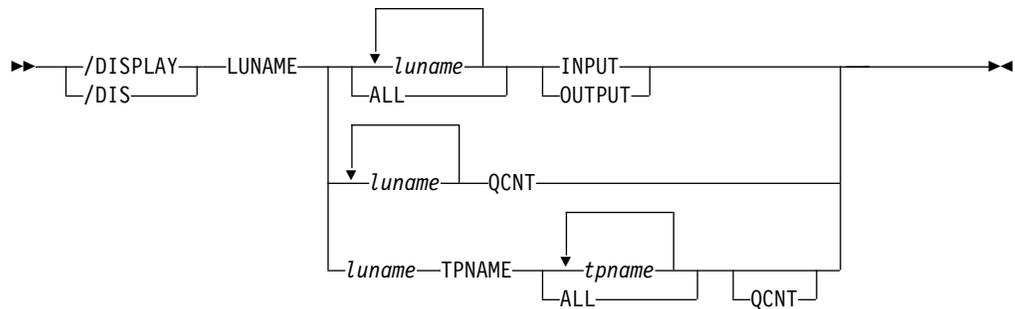
## Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

Table 288. Valid environments for the /DISPLAY LUNAME command and keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
/DISPLAY	X	X	X
INPUT	X		X
LUNAME	X		X
OUTPUT	X		X
QCNT	X		X
TPNAME	X		X

## Syntax



## Keywords

The following keywords are valid for the /DISPLAY LUNAME command:

### LUNAME

The LU name of the LU 6.2 application program about which you want to display information.

### INPUT

Displays the count of LU 6.2 inbound conversations and synchronous outbound activities for the specified LU names and their status.

/DISPLAY LUNAME INPUT can indicate that a command such as the /STO LU ALL INPUT command or the /TRACE SET ON LU ALL INPUT command was entered before the display that applies to all future LU 6.2 inbound conversations.

### OUTPUT

Displays the count of asynchronous output messages that are being enqueued and dequeued for the LU name and the count of LU 6.2 asynchronous outbound conversations for that LU name.

The /DISPLAY LUNAME OUTPUT command can indicate that a command such as the /STO LU ALL OUTPUT command or the /TRACE SET ON LU ALL OUTPUT command was entered that applies to all future LU 6.2 outbound conversations.

#### **QCNT**

Displays the number of output messages on the global queue for the specified LU name. QCNT specifies that global queue count information is to be displayed. Local queue counts are not displayed.

This count represents the message counts of messages that can be processed by the IMS system where the command is issued. This count includes messages that can be processed by any IMS system (messages with no affinity), plus messages that can be processed by the IMS system where the command is issued (messages with an affinity to the IMS system that issued the command).

The /DISPLAY LUNAME QCNT form of the command is only valid in a shared-queues environment.

#### **TPNAME**

Displays the number of asynchronous output messages being enqueued and dequeued and their status.

If the LU name includes a network identifier, you must specify the network-qualified LU name when using the TPNAME keyword. A network-qualified LU name refers to a different resource than a non-qualified LU name.

### **Usage notes**

In a shared-queues environment, use the QCNT keyword to obtain the current message queue counts on the shared queues structure. The local queue counts (ENQCT, DEQCT, and QCT) value that is returned on this command when the QCNT keyword is not specified represents the messages being processed by the IMS system where this command is issued. Do not use local queue counts for queue evaluation purposes.

### **Output fields**

The following output fields are displayed for the /DISPLAY LUNAME command:

#### **LUNAME**

LU name of the LU 6.2 application program

#### **#APPC-CONV**

The number of LU 6.2 inbound conversations

#### **ENQCT**

Total number of messages enqueued on this LU name. In a shared-queues environment, only shows messages enqueued for the local subsystem.

#### **DEQCT**

Total number of messages dequeued from this LU name. In a shared-queues environment, only shows messages dequeued for the local subsystem.

**QCT** Total number of messages still in the queue. In a shared-queues environment, only shows messages enqueued for the local subsystem.

**CONVCT**

Number of LU 6.2 asynchronous outbound conversations.

Status can be stopped (STO), traced (TRA), or stopped and traced.

The CONVCT output field only applies if the OUTPUT keyword was specified.

**GBLQCNT**

The total number of APPC output messages on the global queue. The global queue count includes messages that were placed on the global queue prior to the latest cold start of the owning IMS. Synchronous and asynchronous APPC output messages are included in the global queue count.

**AFFINITY**

The IMSID (or the first seven bytes of the RSENAME if XRF capable) of the IMS to which the output messages have affinity.

**Status** The status displayed can be none, stopped (STO), traced (TRA), or both stopped and traced. The status information does not have a header.

**Examples**

Entry ET:

```
/DISPLAY LU luname1 luname2 INPUT
```

Response ET:

```
LUNAME #APPC-CONV
LUNAME1      15 TRA
LUNAME2      5
*90332/114253*
```

Entry ET:

```
/DISPLAY LU luname1 luname3 BADLUNME OUTPUT
```

Response ET:

```
LUNAME      ENQCT      DEQCT      QCT      CONVCT
LUNAME1      5          3          2          2
LUNAME3      5          4          1          1
BADLUNME IS INVALID
*90332/114253*
```

Entry ET:

```
/DISPLAY LU LUNAME1 TPNAME TPNAME1 BADTPNME
```

Response ET:

```
LUNAME/TPNAME      ENQCT      DEQCT      QCT
LUNAME1
-TPNAME1            0          0          0 STO
BADTPNME IS INVALID
*90332/114253*
```

Entry ET:

```
/DISPLAY LU LUNAME1 TPNAME ALL OUTPUT
```

Response ET:

LUNAME/TPNAME	ENQCT	DEQCT	QCT
LUNAME1			
-TPNAME1	0	0	0 STO
-TPNAME7890123456+	5	3	2 TRA,STO
789END			
-DFSSIDE	2	2	0
*90332/114253*			

Entry ET:

/DISPLAY LU LUNAME2 TPNAME ALL

Response ET:

LUNAME/TPNAME	ENQCT	DEQCT	QCT
NO TPNAME(S) ASSOCIATED WITH LUNAME			
*90332/114253*			

Entry ET:

/DISPLAY LU 'network1.luname1' luname3 'nid2.luname4' OUTPUT

Response ET:

LUNAME	ENQCT	DEQCT	QCT	CONVCT
NETWORK1.LUNAME1	5	3	2	2
LUNAME3	5	4	1	1
NID2.LUNAME4	0	0	0	0
*92232/114253*				

Entry ET:

/DISPLAY LU 'network1.luname1' TPNAME tpname1 badtpnme

Response ET:

LUNAME/TPNAME	ENQCT	DEQCT	QCT
NETWORK1.LUNAME1			
-TPNAME1	0	0	0 STO
-BADTPNME	IS INVALID		
*92232/114253*			

Entry ET:

/DISPLAY LU ALL INPUT

Response ET

LUNAME	#APPC-CONV
NET1.LU1	0 STO,TRA
NET2.LU1	0
LU3	0 STO,TRA
NET4.LU3	0 STO
NET5.LU5	0 TRA
FUTURE INPUT:	STO,TRA
*93069/114435*	

Entry ET:

/DISPLAY LU ALL OUTPUT

Response ET:

LUNAME	ENQCT	DEQCT	QCT	CONVCT
NET1.LU1	0	0	0	0 STO
NET2.LU1	0	0	0	0 TRA
LU3	0	0	0	0 STO,TRA

```

NET4.LU3          0      0      0      0  TRA
NET5.LU5          0      0      0      0  STO
FUTURE OUTPUT:  STO,TRA
*93069/114435*

```

Entry ET:

```
/DISPLAY LU L62MVS1 LU1 CTA73 QCNT
```

Response ET:

LUNAME	GBLQCT	AFFINITY
DEBVMU00.CTA73	8	SYS3
L62MVS1	22	SYS3
CTA73	6	SYS3
NTWKA.CTA73	16	SYS3
L62MVS1	26	IMS2
NTWKA.CTA73	4	IMS2
LU1	0	

\*03202/143132\*

Entry ET:

```
/DIS LU CTA73 TPNAME ALL QCNT
```

Response ET:

LUNAME/TPNAME	GBLQCT	AFFINITY
DEBVMU00.CTA73		
-DFAASYNC	8	SYS3
CTA73		
-TPNM012345678901+	6	SYS3
234567890123456789012345678901234567890123456789		
NTWKA.CTA73		
-TP100	4	IMS2
-TP100	16	SYS3

\*03202/143204\*

Entry ET:

```
/DIS LU LU100 TPNAME TP100 TP300 TP200 QCNT
```

Response ET:

LUNAME/TPNAME	GBLQCT	AFFINITY
NTWK2.LU100		
-TP100	8	IMS2
NTWK1.LU100		
-TP100	8	IMS2
-TP200	10	IMS2
LU100		
-TP300	0	

\*03202/143201\*

Entry ET:

```
/DIS LU SIDENAME TPNAME DFSSIDE QCNT
```

Response ET:

LUNAME/TPNAME	GBLQCT	AFFINITY
SIDENAME		
-DFSSIDE	9	IMS2

\*03202/143231\*

**Related reference:**

Chapter 2, "Command keywords and their synonyms," on page 63

---

## /DISPLAY MASTER command

Use the /DISPLAY MASTER command to display the logical terminal name and the line and physical terminal number associated with the master terminal.

If the 3270 master terminal capability was generated during IMS system definition, the logical terminal name, line, and physical terminal number of the secondary master terminal are also displayed.

Subsections:

- "Environment"
- "Syntax"
- "Equivalent IMS type-2 commands"
- "Examples"

### Environment

The /DISPLAY MASTER command is valid in DB/DC and DCCTL environments.

### Syntax

▶▶ `/DISPLAY MASTER` ◀◀  
    └─/DIS─┘

### Equivalent IMS type-2 commands

The following table shows variations of the /DISPLAY MASTER command and the IMS type-2 commands that perform similar functions.

Table 289. Type-2 equivalents for the /DISPLAY MASTER command

Task	/DISPLAY MASTER command	Similar IMS type-2 command
Displays the primary and secondary master terminal.	/DISPLAY MASTER	QUERY LTERM STATUS(MTO,SMTO)

### Examples

Entry ET:

```
/DISPLAY MASTER
```

Response ET:

```
LTERM MASTER  
PTERM 3- 1  
*89117/130245*
```

**Related reference:**

 [QUERY LTERM command \(Commands\)](#)

---

## /DISPLAY MODIFY command

Use the /DISPLAY MODIFY command to display online change status for this IMS, which is participating in a local or global online change.

The command displays the following information:

- Online change libraries
- Work in progress for resources to be deleted or changed by a /MODIFY COMMIT or a INITIATE OLC PHASE(PREPARE) command
- Resources to be added, changed, and deleted by a /MODIFY COMMIT or an INITIATE OLC PHASE(COMMIT) command

Subsections:

- “Environment”
- “Syntax” on page 622
- “Keywords” on page 622
- “Usage notes” on page 624
- “Equivalent IMS type-2 commands” on page 624
- “Output fields” on page 624
- “Examples” on page 629

### Environment

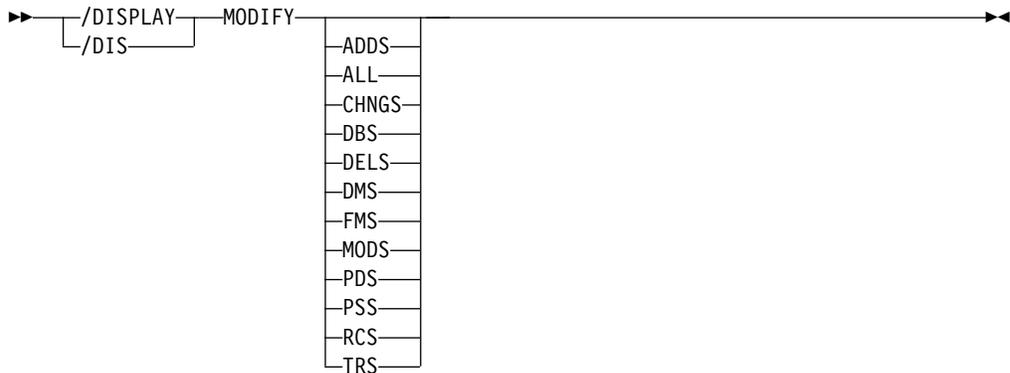
The /DISPLAY MODIFY command is valid in DB/DC, DBCTL, and DCCTL environments.

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

*Table 290. Valid environments for the /DISPLAY MODIFY command and keywords*

Command / Keywords	DB/DC	DBCTL	DCCTL
/DISPLAY	X	X	X
MODIFY	X	X	X
ADDS	X	X	X
ALL	X	X	X
CHNGS	X	X	X
DBS	X	X	
DELS	X	X	X
DMS	X	X	
FMS	X		X
MODS	X	X	X
PDS	X	X	X
PSS	X	X	X
RCS	X		X
TRS	X		X

## Syntax



## Keywords

The following keywords are valid for the /DISPLAY MODIFY command:

### <blanks>

Displays the status of online change libraries.

### ADDS

Displays the resources to be added by online change. Each display line contains a resource type, resource name, and a status of ADDED. The resource type can be: DATABASE, DMB, FORMAT, PROGRAM, PSB, RT CODE, or TRAN.

### ALL

Displays the status of online change libraries and any work in progress for database, program, routing code, or transaction resources to be deleted or changed by the /MODIFY COMMIT or the INITIATE OLC PHASE(COMMIT) command.

For DBCTL, transactions do not apply. For DCCTL, databases do not apply.

### CHNGS

Displays the resources to be changed by online change. Each display line contains a resource type, resource name, and a status of CHANGED. Resource type can be: DATABASE, DMB, FORMAT, PROGRAM, PSB, RT CODE, or TRAN.

### DBS

Displays the databases to be changed or deleted with work in progress that would prevent /MODIFY COMMIT from succeeding. If DBS is specified with the ADDS, CHNGS, DELS, or MODS parameters, the databases in MODBLKS to be added, changed, or deleted display. Each display line contains a resource type of DATABASE, the database name, and a status of ADDED, CHANGED, or DELETED.

### DELS

Displays the resources to be deleted by online change. Each display line contains a resource type, resource name, and a status of DELETED. The resource type can be: DATABASE, DMB, FORMAT, PROGRAM, PSB, RT CODE, or TRAN.

**DMS**

If DMS is specified with the ADDS, CHNGS, DELS, or MODS parameters, the DMBs in ACBLIB to be added, changed, or deleted display as ADDED, CHANGED, or DELETED.

If the status of the associated database in MODBLKS is ADDED, DMS displays a DMB as ADDED. If the status of the associated database in MODBLKS is DELETED, DMS displays a DMB as DELETED.

**FMS**

If FMS is specified with the ADDS, CHNGS, DELS, or MODS parameters, the MFS formats to be added, changed, or deleted in FMTLIB display. Each display line contains a resource type of FORMAT, the format name, and a status of ADDED, CHANGED, or DELETED.

**MODS**

Displays the resource to be modified by online change, which are resources to be added, changed, and deleted. Each display line contains the resource type, resource name, and a status of ADDED, CHANGED, or DELETED. The resource type can be: DATABASE, DMB, FORMAT, PROGRAM, PSB, RT CODE, or TRAN.

/DISPLAY MODIFY MODS shows all of the resources to be added, changed, or deleted by /MODIFY COMMIT, which is equivalent to the output from /MODIFY PREPARE with the LTERM keyword specified.

**PDS**

Displays the programs to be changed or deleted in MODBLKS with work in progress that would prevent a /MODIFY COMMIT or an INITIATE OLC PHASE(COMMIT) command from succeeding. If PDS is specified with the ADDS, CHNGS, DELS, or MODS parameters, the programs in MODBLKS to be added, changed, or deleted display. Each display line contains a resource type of PROGRAM, the program name, and a status of ADDED, CHANGED, or DELETED.

**PSS**

If PSS is specified with the ADDS, CHNGS, DELS, or MODS parameters, the PSBs in ACBLIB to be added, changed, or deleted display as ADDED, CHANGED, or DELETED.

If the status of the associated program in MODBLKS is ADDED, PSS displays a PSB as ADDED. If the status of the associated program in MODBLKS is DELETED, PSS displays a PSB as DELETED.

**RCS**

Displays the routing codes to be changed or deleted in MODBLKS with work in progress that would prevent a /MODIFY COMMIT or an INITIATE OLC PHASE(COMMIT) command from succeeding. If RCS is specified with the ADDS, CHNGS, DELS, or MODS parameters, RTCODEs in MODBLKS to be added, changed, or deleted display.

**TRS**

Displays the transactions to be changed or deleted in MODBLKS with work in progress that would prevent a /MODIFY COMMIT or an INITIATE OLC PHASE(COMMIT) command from succeeding. If TRS is specified with the ADDS, CHNGS, DELS, or MODS parameters, transactions in MODBLKS to be added, changed, or deleted display.

## Usage notes

The /DISPLAY MODIFY command with one or more keywords (not including ALL) is only valid if IMS is in the MODIFY PREPARE state, which means that a /MODIFY PREPARE command or an INITIATE OLC PHASE(PREPARE) command completed successfully.

If IMS cannot access the global queue counts, IMS displays the GLOBAL QUEUE COUNT INTERNAL ERROR status. If IMS cannot obtain sufficient storage to query global queue counts, IMS displays the GLOBAL QUEUE COUNT STORAGE ERROR status. In either case, online change will fail if there are global queue counts for resources being changed or deleted.

For an IMS that has online change for MODBLKS disabled and no MODBLKS data sets are included in the IMS JCL, no information about active and inactive MODBLKS data sets is displayed.

## Equivalent IMS type-2 commands

The following table shows variations of the /DISPLAY MODIFY command and the IMS type-2 commands that perform similar functions.

Table 291. Type-2 equivalents for the /DISPLAY MODIFY command.

Task	/DISPLAY MODIFY command	Similar IMS type-2 command
Displays work in progress for runtime resource definitions that would cause a DELETE, /MODIFY COMMIT, INITIATE OLC PHASE(COMMIT), or UPDATE command to change resource definitions to fail.	/DISPLAY MODIFY ALL	QUERY DB NAME( <i>dbname</i> ) SHOW(WORK)
		QUERY PGM NAME( <i>pgmname</i> ) SHOW(WORK)
		QUERY RTC NAME( <i>rtcode</i> ) SHOW(WORK)
		QUERY TRAN NAME( <i>tranname</i> ) SHOW(WORK)

## Output fields

The online change libraries are described in the output in the following order: the word "LIBRARY", the library name, the library status displayed as A for active, I for inactive, or U for unallocated, and the data set in which the library resides. If the ACBLIB data sets are allocated by JCL, the status displayed for each library is displayed as A for active or I for inactive. It is followed by the DD names and data set names that make up the library. If the ACBLIB data sets are allocated by DFSMDA, the status of the inactive ACBLIB data sets is displayed as U for unallocated. The data set names displayed are those specified in the DFSMDA member for that ACBLIB DD statement.

Status information is displayed for the following libraries:

- IMSACBA and IMSACBB
- FORMATA and FORMATB
- MODBLKSA and MODBLKSB

The following output fields are displayed for the /DISPLAY MODIFY command.

**Resource type**

Resource type can be LIBRARY, DATABASE, PROGRAM, RTCODE, or TRAN.

**Resource name**

The name of the library, database, program, routing code, or transaction.

**Status** Status of the resource or data set name, for a library.

The following status terms are displayed by the /DISPLAY MODIFY command:

**ACTIVE**

The routing code rcode is active.

**ADDED**

One of the following:

- The database, DMB, MFS format, program, PSB, Fast Path routing code (RTCODE), or transaction resource is to be added by online change.
- The DMB already exists in ACBLIB and its associated database is to be added to MODBLKS, making the database usable.
- The PSB already exists in ACBLIB and its associated program is to be added to MODBLKS, making the program usable.

**AREAS OPEN:**

A DEDB database has open areas. One or more lines listing the open areas follow this line.

**CHANGED**

One of the following:

- The database, DMB, MFS format, program, PSB, Fast Path routing code (RTCODE), or transaction resource is to be changed by online change. The prepare command has stopped queuing to transactions affected by the online change. Affected transactions are rejected from an input terminal until the online change is committed or aborted. Program-to-program switches are not rejected. Commit will fail for the resource to be changed, if it has work in progress.
- The program is not to be changed, but is associated with a PSB whose JAVA attribute is to be changed. The program is treated as if it is to be changed, because a PSB scheduling attribute is changing.
- The routing code is not to be changed, but references a PSB that is to be deleted. The routing code is treated as if it is to be changed, because it will no longer be usable.
- The transaction is not to be changed, but references a PSB that is to be deleted. The transaction is treated as if it is to be changed, because it will no longer be usable.

**CHANGED (DSID CHANGED)**

The area is to be changed and the data set ID (area number) is to be changed.

**Attention:** Actions must be taken to prevent damage to the area. See The online change function, DEDBs, and Availability of IFP and MPP Regions (Database Administration).

### **CHANGED (TIMESTAMP ONLY)**

The PSB is to be changed, but only the time stamp is to be changed. The PSB is considered to be affected by the online change, even though nothing is really changing.

### **CONVERSATION TERM/USER**

The transaction to be changed or deleted or the transaction referencing a program or database to be changed or deleted is in conversation. The terminal originating the conversational transaction and the conversation ID are also displayed.

The terminal information displayed is:

- nodename for VTAM terminals
- nodename and username if an ETO user is signed on to the node
- username for ETO users that signed off while in conversation
- lin#-pte# (line and pterm number) for IMS-supported terminals
- 1-SC for the system console

### **CONVERSATION LUNAME**

An APPC logical unit originated the conversational transaction that is to be changed or deleted or that references a program or database to be changed or deleted. The LUNAME and conversation ID are also displayed. The LUNAME information is in the format of luname, or networkid.luname, if networkid-qualified.

### **CONVERSATION TMEM/TPIPE**

An OTMA tmember/tpipe originated a conversational transaction that is to be changed or deleted or that references a program or database to be changed or deleted. The tmember, tpipe and conversation ID are also displayed. The tmember and tpipe information is in the format of tmember.tpipe, where the tmember name can be up to 8 characters.

### **DATABASE IN USE**

The database is in use by one or more active regions referencing the database.

### **/DBD ACTIVE**

A /DBDUMP command is in progress for a database to be changed or deleted.

### **/DBR ACTIVE**

A /DBRECOVERY command is in progress for a database to be changed or deleted.

### **/DBR OR /STOP IN PROGRESS**

A /DBRECOVERY command or /STOP command is in progress for a DEDB database.

### **DBR NEEDED FOR PARTITION**

A /DBRECOVERY command must be issued for the DB partition before the database can be changed or deleted by online change

### **DELETED**

One of the following:

- The database, DMB, MFS format, program, PSB, Fast Path routing code (RTCODE), or transaction resource is to be deleted

by online change. The prepare command has stopped queuing to transactions affected by the online change. Affected transactions are rejected from an input terminal until the online change is committed or aborted. Program-to-program switches are not rejected. Commit will fail for the resource to be deleted, if it has work in progress.

- The DMB is not to be changed, but references a database that is to be deleted. The DMB is treated as if it is to be deleted, because IMS will no longer have any information about the DMB.
- The PSB is not to be changed, but references a program that is to be deleted. The PSB is treated as if it is to be deleted, because IMS will no longer have any information about the PSB.

#### **ELIGIBLE FOR SCHEDULING**

The transaction is eligible for scheduling and cannot be deleted or changed by online change. The transaction itself is being changed or deleted, or the program it references is being changed. Stop the transaction before attempting another online change commit. If the only thing being changed is the PSB associated with the program that the transaction references, the transaction might not need to be stopped.

#### **IN USE**

Queuing is in progress for this transaction (either terminal input or program-to-program switch).

#### **NOT CHANGED (DMB CHANGED ONLY)**

The PSB is not to be changed, but references a DMB that is to be changed or deleted. The PSB is considered to be affected by the online change, where commit will fail for work in progress.

#### **NOT CHANGED (PDIR/PSB CHANGE ONLY)**

One of the following:

- The routing code is not to be changed, but references a PDIR or PSB that is to be changed. The routing code is considered to be affected by the online change, where commit will fail for work in progress.
- The transaction is not to be changed, but references a PSB that is to be changed. This status does not apply to PSB attribute changes that affect scheduling, such as the JAVA attribute. The transaction is not affected by online change, so commit will succeed even if there is work in progress.

#### **NOT OPEN, BUT DBR NEEDED ON TRACKER**

The database is not open, but the area can be open, so the database needs to be taken offline by use of the /DBRECOVERY command. This can happen if the area is not stopped and the database is not open and prepared to be online changed on a tracker environment. The area can be open if log records type 5701 or 5950 are routed to the RSR tracking IMS from the active IMS while online change is in progress. It is then required that the database be taken offline in the case of database level change. The tracking suspend point (Suspended Log Sequence Number) is registered in the RECON data set when the area is stopped. After the online change committed successfully, and when the /STA AREA command is

entered, the online forward recovery (OFR) will open the area and all log records from the suspended point will be applied to the area.

#### **PSB SCHEDULED**

A program referencing a database to be changed or deleted is scheduled.

#### **QUEUING *nn***

Messages are queued to the transaction or program; *nn* is the number of messages queued. Global queue counts represent the number of messages on the coupling facility list structure for the transaction or program.

#### **Q STRUCTURE IS UNAVAILABLE**

The queue structure is unavailable. IMS is unable to check the global queue count for transactions to be added, changed, or deleted. Commit is not permitted to proceed. The structure may be unavailable for one of the following reasons:

- CQS is unavailable.
- The MSGQ structure failed and it has not been rebuilt yet.
- CQS lost connectivity to the MSGQ structure.

#### **RANDOMIZER: *rndmname***

A DEDB database that has no work in progress lists the randomizer name (*rndmname*). The randomizer display lines are information only, not work in progress that would prevent online change commit from succeeding.

This line is followed by a line with the randomizer name and status of LOADED or NOT LOADED.

If the randomizer is shared by one or more DEDBs, another line is displayed with 'DEDBS SHARING RANDOMIZER:' followed by the randomizer name. Online change will not delete a randomizer that is shared by other DEDBs. This line is followed by one line for each DEDB sharing the same randomizer.

#### **RECOVERY IN PROGRESS**

A /RECOVER START command is in progress to recover one or more databases with the database recovery service.

#### **SAME (DSID CHANGED)**

The area is not to be changed, but the data set ID (area number) is to be changed.

**Attention:** Actions must be taken to prevent damage to the area. See The online change function, DEDBs, and Availability of IFP and MPP Regions (Database Administration).

#### **SCHEDULED**

The named resource (a transaction or program to be changed or deleted, or a program referencing a database to be changed or deleted) is scheduled.

#### **SUSPENDED**

The transaction to be changed or deleted is on the suspend queue.

Work in progress for resources to be deleted or changed causes the /MODIFY COMMIT command to fail. You must wait for the work in progress to complete, or

force it to complete (for example, by issuing a command), before issuing /MODIFY COMMIT. The following table lists the resource type, resource name, and the status displays for work in progress that would prevent the /MODIFY COMMIT command from completing successfully:

*Table 292. Work in progress for resources that prevent a successful /MODIFY COMMIT*

Resource type	Resource name	Status
DATABASE	dbname	AREAS OPEN: /DBD ACTIVE /DBR ACTIVE /DBR OR /STOP IN PROGRESS DATABASE IN USE DBR NEEDED FOR PARTITION nnnn PSB SCHEDULED RANDOMIZER: rndmname RECOVERY IN PROGRESS
AREA	areaname	OPEN RECOVERY IN PROGRESS
PROGRAM	pgmname	SCHEDULED QUEUING nn GLOBAL QUEUE COUNT INTERNAL ERROR GLOBAL QUEUE COUNT STORAGE ERROR
RTCODE	rtcode	ACTIVE
TRAN	trancode	IN USE QUEUING nn SCHEDULED SUSPENDED CONVERSATION TERM/USER nodename ID= nnnn nodename username ID= nnnn username ID= nnnn lin#-pte# ID= nnnn 1- SC ID= nnnn GLOBAL QUEUE COUNT INTERNAL ERROR GLOBAL QUEUE COUNT STORAGE ERROR ELIGIBLE FOR SCHEDULING

In a shared-queues environment, /DISPLAY MODIFY only shows local work in progress, except for the TRAN QUEUING status and the PROGRAM QUEUING status, for which this command shows the sum of local and global queue counts. That is, /DISPLAY MODIFY does not show whether there is work in progress for other IMS subsystems sharing the message queues. If the /DISPLAY MODIFY command shows no work in progress for the IMS subsystem which the command was issued, a /MODIFY COMMIT command can be issued for that IMS subsystem.

## Examples

The following are examples of the /DISPLAY MODIFY command.

### *Example 1 for /DISPLAY MODIFY command*

In these examples for /DISPLAY MODIFY, /MODIFY PREPARE has already succeeded.

Entry ET:

/DISPLAY MODIFY ALL

Response ET:

```
LIBRARY IMSACBA (A) OLC.ACBLIB.ALL
LIBRARY FORMATA (I) IMSQA.FMT1
LIBRARY MODBLKSA (A) I41RTS42.CMODBLKS
LIBRARY IMSACBB (I) OLC.ACBLIB.ALL
LIBRARY FORMATB (A) IMSQA.FMT1
LIBRARY MODBLKSB (I) I41RTS42.CMODBLK1
DATABASE OLCDB088 /DBR ACTIVE
DATABASE OLCDB101 PSB SCHEDULED
PROGRAM OLCPB021 SCHEDULED
PROGRAM OLCPB109 SCHEDULED
RTCODE OLCRC056 ACTIVE
TRAN OLCB105 QUEUING 1
TRAN CDEBTRN5 CONVERSATION TERM/USER 4- 2 ID= 0009
TRAN OLCB109 SCHEDULED
TRAN CDEBTRN8 CONVERSATION TERM/USER L3270D ID= 0002
TRAN OLCB111 QUEUING 5
TRAN CDEBTRN2 CONVERSATION TERM/USER DYNT0001 IMSUS01 ID= 0005
TRAN CDEBTRN3 CONVERSATION TERM/USER 12- 1 ID= 0008
TRAN CDEBTRN1 CONVERSATION TERM/USER IMSUS12 ID= 0001
DISPLAY MODIFY COMPLETE
*93336/093025*
```

Entry ET:

/DISPLAY MODIFY DBS

Response ET:

```
DATABASE CALENDER AREAS OPEN:
  AREA FEBRUARY
  AREA APRIL
  AREA JULY
  AREA AUGUST
  AREA SEPTEMBER
  AREA OCTOBER
  AREA DECEMBER
DATABASE CALENDER RANDOMIZER: RMOD3
DEDBS SHARING RANDOMIZER: RMOD3
DATABASE DEDBJN21 RANDOMIZER: RMOD3
RMOD3 NOT LOADED
DATABASE DEDBJN22 RANDOMIZER: RMOD5
RMOD5 NOT LOADED
DISPLAY MODIFY COMPLETE *95299/161529*
```

Entry ET:

/DISPLAY MODIFY TRS

Response ET:

```
TRAN OLCFT112 QUEUING 2
TRAN OLCFT115 QUEUING 23
PROGRAM OLCFP115 QUEUING 1
TRAN CDEBTRN8 GLOBAL QUEUE COUNT INTERNAL ERROR
TRAN OLCFT116 QUEUING 6
DISPLAY MODIFY COMPLETE *96193/135935*
```

Explanation: Work is in progress that will cause online change to fail, because several transactions and a FP program have a global queue count on the shared queues. The global queue count for transaction CDEBTRN8 could not be determined because of an internal error.

*Example 2 for /DISPLAY MODIFY command*

Entry ET:

/DISPLAY MODIFY MODS

Response ET:

```
DFS000I   FORMAT   0C01_FF02A  _O  ADDED
DFS000I   FORMAT   0C01_FF02B  _O  ADDED
DFS000I   FORMAT   0C7F_FN01A  _O  CHANGED
DFS000I   FORMAT   0C7F_FN01B  _O  CHANGED
DFS000I   FORMAT   0C7F_FN01C  _O  CHANGED
DFS000I   FORMAT   0C7F_FF04A  _O  DELETED
DFS000I   FORMAT   MF04E        _O  DELETED
DFS000I   DATABASE OLCDB101      ADDED
DFS000I   DATABASE OLCDB102      ADDED
DFS000I   DATABASE OLCDB103      ADDED
DFS000I   DATABASE OLCDB104      ADDED
DFS000I   DATABASE OLCDB157      ADDED
DFS000I   DATABASE OLCDB111      DELETED
DFS000I   DATABASE OLCDB159      DELETED
DFS000I   DATABASE OLCDI105      DELETED
DFS000I   DATABASE OLCDI106      DELETED
DFS000I   DMB      OLCDB101      ADDED
DFS000I   DMB      OLCDB102      ADDED
DFS000I   DMB      DEDBJN24      CHANGED
DFS000I   AREA     DB24A000      CHANGED
DFS000I   DMB      OLCDB105      DELETED
DFS000I   PROGRAM  OLCFP107      ADDED
DFS000I   PROGRAM  OLCFP108      ADDED
DFS000I   PROGRAM  OLCPB101      ADDED
DFS000I   PROGRAM  OLCFP109      CHANGED
DFS000I   PROGRAM  OLCFP110      CHANGED
DFS000I   PROGRAM  OLCFP111      CHANGED
DFS000I   PROGRAM  OLCPB105      DELETED
DFS000I   PROGRAM  OLCPB106      DELETED
DFS000I   PSB      OLCFP107      ADDED
DFS000I   PSB      OLCFP108      ADDED
DFS000I   PSB      CDEBS         CHANGED
DFS000I   PSB      OLCPB105      DELETED
DFS000I   PSB      OLCPB106      DELETED
DFS000I   RT CODE  OLCFT108      ADDED
DFS000I   RT CODE  OLCFR119      CHANGED
DFS000I   RT CODE  OLCFR117      DELETED
DFS000I   TRAN     OLCFT108      ADDED
DFS000I   TRAN     OLCTBG17      ADDED
DFS000I   TRAN     CDEBTRNA      CHANGED
DFS000I   TRAN     CDEBTRN1      CHANGED
DFS000I   TRAN     CONV31X       DELETED
DFS000I   DISPLAY  MODIFY COMPLETE *98356/094452*   SYS3
```

Explanation: Displays resources to be modified (added, changed, or deleted) by online change.

### *Example 3 for /DISPLAY MODIFY command*

Entry ET:

/DISPLAY MODIFY ALL

Response ET:

```
LIBRARY  IMSACBA  (A)  IMSTESTG.DELTALIB
          (A)  IMSTESTL.TNUC2
          (A)  IMSTESTG.IMS61RC.ACBLIB
          (A)  IMSTESTG.IMS61R.ACBLIB
LIBRARY  FORMATA  (A)  IMSTESTG.MFS.OVERRIDE.FORM
          (A)  IMSTESTG.MFS.FORMAT
```

```

LIBRARY MODBLKSA (A) IMSQA.FMT1
LIBRARY MODBLKSA (A) IMSBLD.I61RTS25.CMODBLKS
LIBRARY IMSACBB (I) IMSTESTG.DELTALIB
LIBRARY IMSACBB (I) IMSTESTL.TNUC3
LIBRARY IMSACBB (I) IMSTESTG.IMS61RC.ACBLIB
LIBRARY IMSACBB (I) IMSTESTG.IMS61R.ACBLIB
LIBRARY FORMATB (I) IMSTESTG.MFS.OVERRIDE.FORM
LIBRARY FORMATB (I) IMSTESTG.MFS.FORMAT
LIBRARY FORMATB (I) IMSQA.FMT1
LIBRARY MODBLKSB (I) IMSBLD.I61RTS25.CMODBLKS
DATABASE DEDBJN21 RANDOMIZER: RMOD3 (1)
RMOD3 LOADED (2)
DEDBS SHARING RANDOMIZER: RMOD3 (3)
DEDB DEDBJN22
DEDB DEDBJN23
DEDB DEDB3301
DATABASE DEDBJN22 RANDOMIZER: RMOD3
RMOD3 NOT LOADED
DEDBS SHARING RANDOMIZER: RMOD3
DEDB DEDBJN21
DEDB DEDBJN23
DEDB DEDB3301
DATABASE DEDBJN23 RANDOMIZER: RMOD3
RMOD3 LOADED
DEDBS SHARING RANDOMIZER: RMOD3
DEDB DEDBJN21
DEDB DEDBJN22
DEDB DEDB3301
NO WORK PENDING *99328/110535* SYS3

```

Explanation: In this example:

- DEDB databases DEDBJN21, DEDBJN22, and DEDBJN23 have undergone Online Change.
- After a database line for DEDBS displays, a line follows containing randomizer status (loaded or not loaded).
- After the randomizer status displays, a line might follow containing DEDBS SHARING RANDOMIZER; this line is followed by DEDB lines listing the DEDBs that share the randomizer.

**Example 6 for /DISPLAY MODIFY command**

Entry ET:

```
/DISPLAY MODIFY ALL
```

Response ET:

```

LIBRARY IMSACBA (A) IMSTESTG.DELTA1
LIBRARY IMSACBA (A) IMSTESTG.IMS10AC.ACBLIB1
LIBRARY IMSACBA (A) IMSTESTG.IMS10A.ACBLIB1
LIBRARY FORMATA (A) IMSTESTG.MFS.FORMAT1
LIBRARY FORMATA (A) IMSTESTG.MFS.FORMAT2
LIBRARY FORMATA (A) IMSQA.FMT1
LIBRARY MODBLKSA (A) IMSBLD.I10ATS17.COMBLKS1
LIBRARY IMSACBB (U) IMSTESTG.DELTA2
LIBRARY IMSACBB (U) IMSTESTG.IMS10AC.ACBLIB2
LIBRARY IMSACBB (U) IMSTESTG.IMS10A.ACBLIB2
LIBRARY FORMATB (I) IMSTESTG.MFS.FORMAT3
LIBRARY FORMATB (I) IMSTESTG.MFS.FORMAT4
LIBRARY FORMATB (I) IMSQA.FMT1
LIBRARY MODBLKSB (I) IMSBLD.I10ATS17.COMBLKS2
DISPLAY MODIFY COMPLETE *07003/110121* SYS3

```

TSO SPOC output:

```

MBRNAME  MESSAGES
-----
SYS3      LIBRARY  IMSACBA  (A)  IMSTESTG.DELTA1
SYS3      (A)  IMSTESTG.IMS10AC.ACBLIB1
SYS3      (A)  IMSTESTG.IMS10A.ACBLIB1
SYS3      LIBRARY  FORMATA  (A)  IMSTESTG.MFS.OVERRIDE.FORMAT1
SYS3      (A)  IMSTESTG.MFS.FORMAT1
SYS3      (A)  IMSQA.FMT1
SYS3      LIBRARY  MODBLKSA (A)  IMSBLD.I10ATS17.COMBLKS1
SYS3      LIBRARY  IMSACBB  (U)  IMSTESTG.DELTA2
SYS3      (U)  IMSTESTG.IMS10AC.ACBLIB2
SYS3      (U)  IMSTESTG.IMS10A.ACBLIB2
SYS3      LIBRARY  FORMATB  (I)  IMSTESTG.MFS.OVERRIDE.FORMAT2
SYS3      (I)  IMSTESTG.MFS.FORMAT2
SYS3      (I)  IMSQA.FMT2
SYS3      LIBRARY  MODBLKSB (I)  IMSBLD.I10ATS17.COMBLKS2
SYS3      DISPLAY MODIFY COMPLETE *07003/114249*

```

Explanation: In this example, IMSACBA is the active ACBLIB; IMSACBB is the inactive ACBLIB. The inactive ACBLIB data sets are displayed even though they are not allocated. The “U” indicates that they are unallocated.

*Example 7 for /DISPLAY MODIFY command*

Entry ET:

```
/DISPLAY MODIFY ALL
```

Response ET:

```

DFS000I  LIBRARY  IMSACBA  (A)  IMSTESTG.DELTALIB
DFS000I  (A)  IMSTESTG.IMS10AC.ACBLIB
DFS000I  (A)  IMSTESTG.IMS10A.ACBLIB
DFS000I  LIBRARY  FORMATA  (A)  IMSTESTG.MFS.FORMAT1
DFS000I  (A)  IMSTESTG.MFS.FORMAT2
DFS000I  (A)  IMSQA.FMT1
DFS000I  LIBRARY  MODBLKSA (A)  IMSBLD.I10ATS17.COMBLKS1
DFS000I  LIBRARY  IMSACBB  ( )  NO DFSMDA MEMBER
DFS000I  LIBRARY  FORMATB  (I)  IMSTESTG.MFS.FORMAT3
DFS000I  (I)  IMSTESTG.MFS.FORMAT4
DFS000I  (I)  IMSQA.FMT1
DFS000I  LIBRARY  MODBLKSB (I)  IMSBLD.I10ATS17.COMBLKS1
DFS000I  DISPLAY MODIFY COMPLETE *07003/110121*  SYS3

```

Explanation: In this example, IMSACBA is the active ACBLIB; IMSACBB is the inactive ACBLIB. There is no DFSMDA member for the inactive ACBLIB data sets. The data set name field contains “NO DFSMDA MEMBER”. The status field is left blank.

*Example 8 for /DISPLAY MODIFY command*

The following example shows an online HALDB database, DBVHJD05, that is currently being altered.

Entry ET:

```
/DISPLAY MODIFY DBS
```

TSO SPOC output:

```

MbrName  Messages
-----
IMS1      DATABASE DBOHIDK5      DBR NEEDED FOR PARTITION

```

```

IMS1      DATABASE DBVHDJ05      ALTER INCOMPLETE
IMS1      DATABASE DBVHDJ05      4 PSBS SCHEDULED
IMS1      DATABASE DBVHDJ05      DATABASE IN USE

```

*Example 9 for /DISPLAY MODIFY command*

Entry ET:

```
/DISPLAY MODIFY ALL
```

Response ET:

```

LIBRARY IMSACBA (A) IMSTESTG.DELTALIB
                (A) IMSTESTL.TEMPA.ACBLIB
                (A) IMSTESTG.IMS800C.ACBLIB
                (A) IMSTESTG.IMS800.ACBLIB
LIBRARY FORMATA (A) IMSTESTG.MFS.OVERRIDE.FORMAT
                (A) IMSTESTG.MFS.FORMAT
                (A) IMSQA.FMT1
LIBRARY MODBLKSA (A) TEMPA.MODBLKS
LIBRARY IMSACBB (I) IMSTESTG.DELTALIB
                (I) IMSTESTL.TEMPB.ACBLIB
                (I) IMSTESTG.IMS800C.ACBLIB
                (I) IMSTESTG.IMS800.ACBLIB
                (I) IMSTESTG.MFS.OVERRIDE.FORMAT
LIBRARY FORMATB (I) IMSTESTG.MFS.FORMAT
                (I) IMSQA.FMT1
LIBRARY MODBLKSB (I) TEMPB.MODBLKS
DATABASE DEDBJN21 NOT OPEN, BUT DBR NEEDED ON TRACKER (1)
DATABASE DEDBJN21 RANDOMIZER: RMOD3
RMOD3 NOT LOADED
DATABASE DEDBJN22 RANDOMIZER: RMOD5
RMOD5 NOT LOADED
DATABASE DEDBJN24 RANDOMIZER: RMOD4
RMOD4 NOT LOADED
DATABASE THFP1WEB RANDOMIZER: DBFHDC44
DBFHDC44 NOT LOADED
DISPLAY          MODIFY COMPLETE *01303/085213*  SYS3

```

Explanation: The DEDB database DEDBJN21 is prepared to be online changed on a tracker environment. If the AREA is not stopped and the database is not open, the AREA can be open if log records type 5701 or 5950 are routed to the RSR tracking IMS from the active IMS while Online Change is in progress. It is then required that the database be taken offline by use of the /DBRECOVERY command in the case of database level change. The tracking suspend point (Suspended Log Sequence Number) is registered in RECON when the AREA is stopped. After the online change committed successfully, and when the /STA AREA command is entered, the Online Forward Recovery will open the AREA and all log records from the suspended point will be applied to the AREA.

**Related tasks:**

 The online change function, DEDBs, and Availability of IFP and MPP Regions (Database Administration)

**Related reference:**

“Status and attributes for the /DISPLAY command” on page 523

-  QUERY DB command (Commands)
-  QUERY PGM command (Commands)
-  QUERY RTC command (Commands)
-  QUERY TRAN command (Commands)

## /DISPLAY MSNAME command

Use the /DISPLAY MSNAME command to display the queue counts for the specified logical link path. Generic parameters are supported for the MSNAME keyword.

Subsections:

- “Environment”
- “Syntax”
- “Keywords”
- “Usage notes” on page 636
- “Examples” on page 636

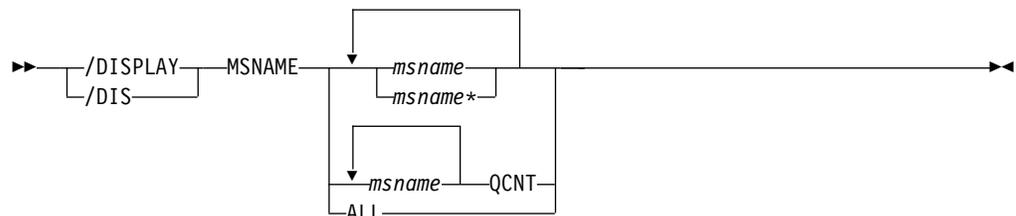
### Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

Table 293. Valid environments for the /DISPLAY MSNAME command and keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
/DISPLAY	X	X	X
MSNAME	X		X
QCNT	X		X

### Syntax



### Keywords

The following keyword can be specified for the /DISPLAY MSNAME command:

#### QCNT

Specifies that global queue count information is to be displayed. If QCNT is not specified, local queue counts are displayed; when it is specified, no local queue counts are displayed.

This count represents the message counts of messages that can be processed by the IMS system where the command is issued. This count includes messages that can be processed by any IMS system (messages with no affinity), plus messages that can be processed by the IMS system where the command is issued (messages with an affinity to the IMS system that issued the command).

This keyword is valid only in a shared-queues environment.

## Usage notes

In a shared-queues environment, use the QCNT keyword to obtain the current message queue counts on the shared queues structure. The local queue counts (ENQCT, DEQCT, and QCT) value that is returned on this command when the QCNT keyword is not specified represents the messages being processed by the IMS system where this command is issued. Do not use local queue counts for queue evaluation purposes.

## Examples

Entry ET:

```
/DISPLAY MSNAME ALL
```

Response ET:

MSNAME	ENQCT	DEQCT	QCT
SYSTEM2A	0	0	0
SYSTEM2B	0	0	0
FRESNO	0	0	0
BOSTON	0	0	0
ALBANY	0	0	0
TEST1	0	0	0
*91350/053859*			

Entry ET:

```
/DISPLAY MSNAME ALL
```

Response ET:

MSNAME	ENQCT	DEQCT	QCT	IMS1	
LINK23B4	0	0	0	DYNAMIC	IMS1
LINK12V1	0	0	0	IMS1	
LINK23U2	0	0	0	DYNAMIC	IMS1
ELINK211	0	0	0	DYNAMIC	IMS1
LINK13B3	0	0	0	IMS1	
LINK13V	0	0	0	IMS1	
LINK12V	0	0	0	IMS1	
LINK13C2	0	0	0	IMS1	
LINK13X	0	0	0	IMS1	
LINK12V2	0	0	0	IMS1	
LINK13M2	0	0	0	IMS1	
LINK13V2	0	0	0	IMS1	
ELINK124	0	0	0	IMS1	
LINK23V1	0	0	0	DYNAMIC	IMS1
LINK12B1	0	0	0	IMS1	
LINK12B2	0	0	0	IMS1	
LINK12Y	0	0	0	IMS1	
ELINK120	0	0	0	IMS1	
LINK13B4	0	0	0	IMS1	
LINK21V3	0	0	0	DYNAMIC	IMS1
ELINK1MS	0	0	0	IMS1	
LINK12M1	0	0	0	IMS1	
LINK23V	0	0	0	DYNAMIC	IMS1
LINK13U2	0	0	0	IMS1	
LINK23B3	0	0	0	DYNAMIC	IMS1
ELINK224	0	0	0	DYNAMIC	IMS1
LINK23X	0	0	0	DYNAMIC	IMS1
LINK23V2	0	0	0	DYNAMIC	IMS1
ELINK121	0	0	0	IMS1	
LINK23C2	0	0	0	DYNAMIC	IMS1
LINK12U1	0	0	0	IMS1	
LINK23M2	0	0	0	DYNAMIC	IMS1
ELINK122	0	0	0	IMS1	

```

ELINK222      0      0      0  DYNAMIC  IMS1
LINK12C1      0      0      0    IMS1
LINK13V1      0      0      0    IMS1
ELINK123      0      0      0    IMS1
PTH3TSTB     0      0      0    IMS1
*97258/175636*

```

---

## /DISPLAY NODE command

The /DISPLAY NODE command, when followed by a valid node name or ALL, displays status, queue counts, and number of messages sent to and received from the specified node.

Subsections:

- “Environment”
- “Syntax”
- “Keywords” on page 639
- “Usage notes” on page 641
- “Equivalent IMS type-2 commands” on page 642
- “Output fields” on page 642
- “Examples” on page 643

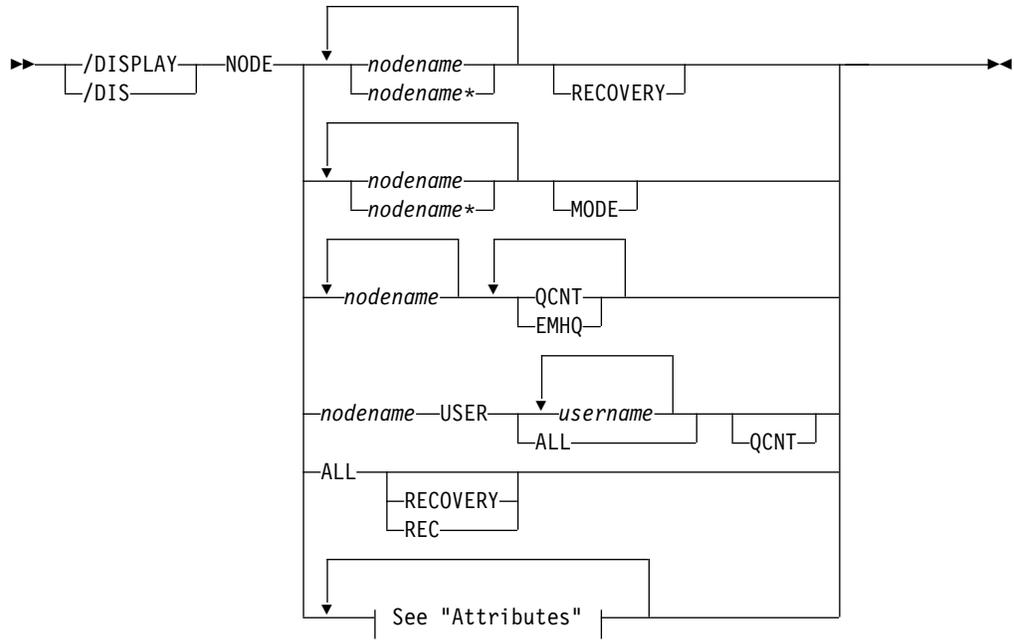
### Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

*Table 294. Valid environments for the /DISPLAY NODE command and keywords*

Command / Keywords	DB/DC	DBCTL	DCCTL
/DISPLAY	X	X	X
EMHQ	X		X
NODE	X		X
QCNT	X		X
RECOVERY	X	X	
USER	X		X

### Syntax



**Attributes:**

ACTIV
AUTOSR
BCKUP
CLSDST
CON
CONVACT
CONVHLD
C1INOP
C2INOP
C3INOP
C4INOP
DEACT
EXCL
FORCES
IDLE
INOP
LOCK
LOST
MFST
OPNDST
PAGE
PRI
PRST
QUI
RELREQ
RESP
RESPINP
RESYNC
SEC
SHUT
SIGN
SIMLOGON
STATIC
STOPPED
TEST
TKOTRA
TRA

## Keywords

The following keywords can be specified for the /DISPLAY NODE command:

### MODE

Displays the mode table names that are associated with the specified nodes.

### EMHQ

Specifies that global queue count information on the Expedited Message Handler Queues (EMHQ) displays when both EMHQ and QCNT are specified. If EMHQ is specified, QCNT must also be specified. This keyword is valid only in a shared-queues environment.

### QCNT

Specifies that global queue count information is to be displayed. If QCNT is not specified, local queue counts are displayed; when it is specified, no local queue counts are displayed. If QCNT is specified and EMHQ is not, global queue count information on the shared message queues (MSGQ) is displayed.

This count represents the message counts of messages that can be processed by the IMS system where the command is issued. This count includes messages that can be processed by any IMS system (messages with no affinity), plus

messages that can be processed by the IMS system where the command is issued (messages with an affinity to the IMS system that issued the command).

IMS does not check if the specified name is a valid LTERM in the IMS subsystem on which the command is entered.

This command does not display queue counts for Fast Path output messages for the NODE unless EMHQ and QCNT are specified.

This keyword is valid only in a shared-queues environment.

#### **RECOVERY**

The /DISPLAY NODE|USER RECOVERY command can be used to display the recovery values that pertain to the node or user. The recovery values may be displayed from the local control blocks, if available, or from values saved for the node or user in the Resource Manager, if resource information is being kept in Resource Manager, and the node or user is defined to have its status kept in Resource Manager.

#### **Attributes**

Reserved parameters for the /DISPLAY NODE command and cannot be used to name nodes.

Attributes listed in “Keywords and reserved parameters” on page 560 for the /DISPLAY AREA command can be used with the NODE keyword. Nodes are selected for display based on the attribute or attributes specified. The attribute display format is the same as the standard NODE display. Attributes usually correspond to the conditions displayed for nodes. Any exceptions are flagged in the following list.

For example, /DISPLAY NODE TRACE displays all nodes that are currently being traced. Following is a list of the attributes that can be specified with the NODE keyword.

- ACTIV
- AUTOSR (Corresponds to the NODE conditions ASR)
- BCKUP
- CLSDST
- CON
- CONVACT (Corresponds to the NODE condition CONV-ACT)
- CONVHLD (Corresponds to the NODE condition CONV-HLD)
- C1INOP
- C2INOP
- C3INOP
- C4INOP
- DEACT
- EXCL
- FORCES (Corresponds to the NODE condition FORCE)
- IDLE
- INOP
- LOCK
- LOST
- MFST
- OPNDST
- PAGE

- PRI
- PRST
- QUI
- RECOVER
- RELREQ
- RESP
- RESPINP
- RESYNC
- SEC
- SHUT
- SIGN
- SIMLOGON
- STATIC
- STOPPED
- TEST
- TKOTRA
- TRA

**Note:** The /DISPLAY NODE RESPINP command returns all nodes that have RESP-INP and RESP-INP-FP status.

N/A is displayed by the /DISPLAY NODE ALL MODE command for VTAM 3270 nodes.

## Usage notes

In a shared-queues environment, use the QCNT keyword to obtain the current message queue counts on the shared queues structure. The local queue counts (ENQCT, DEQCT, and QCT) value that is returned on this command when the QCNT keyword is not specified represents the messages being processed by the IMS system where this command is issued. Do not use local queue counts for queue evaluation purposes.

When using ISC, the output for a parallel-session node shows one line of information for each session of the node whether the node is active or not and whether logical terminals are assigned or not. If a session is active, the user identifier (preceded by a dash) and the status and queue counts are displayed on a subsequent line. The /DISPLAY NODE command displays queue counts up to a maximum of 32 KB. The queue count for the /DISPLAY NODE command is reset to zero when sessions are not active, or when a signed-on user issues /SIGN OFF.

The USER keyword is valid for ISC nodes with users or dynamic nodes with signed on users. The NODE parameter can be generic where the USER keyword is not present. The generic parameter specifies nodes that already exist. A status of STATIC is displayed for nodes defined to IMS by way of system definition. STATIC is also supported as an attribute parameter. Static nodes are used with the /DISPLAY NODE command in the same way that dynamic nodes are used.

The /DISPLAY NODE command shows signed on users, where the user ID follows the status SIGN in parentheses.

### Restrictions for using NODE and USER parameters together:

- Commands with the NODE USER keyword pair are valid only if:
  - The USER is signed on to the NODE
  - In an ISC environment, the USER is allocated to the NODE
  - The nodes and users already exist
- /DISPLAY NODE USER commands are valid for ISC and non-ISC nodes and users.

## Equivalent IMS type-2 commands

The following table shows variations of the /DISPLAY NODE command and the IMS type-2 commands that perform similar functions.

Table 295. Type-2 equivalents for the /DISPLAY NODE command.

Task	/DISPLAY NODE command	Similar IMS type-2 command
Displays VTAM connection identifier (CID).	/DISPLAY NODE <i>node</i>	QUERY NODE SHOW(CID)
Displays send/receive message counts.	/DISPLAY NODE <i>node</i>	QUERY NODE SHOW(COUNT)
Displays preset destination.	/DISPLAY NODE <i>node</i>	QUERY NODE SHOW(PRESET)
Displays queue count.	/DISPLAY NODE <i>node</i> /DISPLAY NODE <i>node</i> QCNT	QUERY NODE SHOW(QCNT)
Displays status for a particular node.	/DISPLAY NODE <i>node</i>	QUERY NODE SHOW(STATUS)
Displays terminal type.	/DISPLAY NODE <i>node</i>	QUERY NODE SHOW(TYPE)
Displays user ID.	/DISPLAY NODE <i>node</i>	QUERY NODE SHOW(USERID)
Displays user.	/DISPLAY NODE <i>node</i>	QUERY NODE SHOW(USER)
Displays VTAM mode table names.	/DISPLAY NODE <i>node</i> MODE	QUERY NODE SHOW(MODETBL)
Displays message queue count in the Expedited Message Handler (EMH) queues.	/DISPLAY NODE <i>node</i> QCNT EMHQ	QUERY NODE SHOW(EMHQ)
Displays owner IMSID in RM resource structure.	/DISPLAY NODE <i>node</i> RECOVERY	QUERY NODE SHOW(OWNER)
Displays status recovery information.	/DISPLAY NODE <i>node</i> RECOVERY	QUERY NODE SHOW(RECOVERY)

## Output fields

The output fields displayed for the /DISPLAY NODE command include the following:

### NODE-USR

Node name and user identifier.

**TYPE** Type of node.

### DEF MODETBL

Default mode table name set by way of system definition or logon descriptor or /CHANGE command. This name can be overridden using the /OPNDST command or the LOGON exit for non-IMS session initiations.

## ACT MODETBL

Mode table name actually used to initiate the session. This name is only displayed while the session is active. The field is blank at normal session termination.

## OWNER

The IMS ID (RSEname if an XRF system) of the IMS system that currently owns this resource. If the resource is not currently owned, this field will be displayed as NONE

**SRM** The scope of recovery for a resource, and from where the end-user significant status is maintained and recovered.

## CONV

Frequently changes for a resource, and thus there are performance considerations related to maintaining end-user significant status. The installation can tell IMS how to recover end-user significant status by specifying the level of recovery for each status. Shown for each resource is the level of recovery for each end-user significant status.

**STSN** Frequently changes for a resource, and thus there are performance considerations related to maintaining end-user significant status. The installation can tell IMS how to recover end-user significant status by specifying the level of recovery for each status. Shown for each resource is the level of recovery for each end-user significant status.

## FPATH

Frequently changes for a resource, and thus there are performance considerations related to maintaining end-user significant status. The installation can tell IMS how to recover end-user significant status by specifying the level of recovery for each status. Shown for each resource is the level of recovery for each end-user significant status.

## Examples

The following are examples of the /DISPLAY NODE command.

### *Example 1 for /DISPLAY NODE command*

Entry ET:

```
/DISPLAY NODE DTSLU* LUP1 ENDS01 LU6NDPA LU6NDPH
```

Response ET:

NODE-USR	TYPE	CID	RECD	ENQCT	DEQCT	QCT	SENT	SYS3
DTSLU002	SLUP	010000D3	0	0	0	0	0	SIGN(IMSUS05 ) IDLE CON
DTSLU202	SLU2	010000CB	0	0	0	0	1	SIGN(IMSUS02 ) IDLE CON
DTSLU603	LUT6							
-IMSUS04		010000C7	0	0	0	0	0	SIGN(IMSUS04 ) IDLE CON PRI
DTSLU205	SLU2	010000A5	2	1	1	0	3	SIGN(IMSUS14 ) IDLE CONV-ACT CON
DTSLU204	SLU2	010000A4	3	0	0	0	4	SIGN(IMSUS13 ) IDLE CON EXC
DTSLU203	SLU2	010000A3	3	0	0	0	4	SIGN(IMSUS12 ) PRST(WTOR ) IDLE MFST CON
LUP1	SLUP	010000CD	0	0	0	0	0	IDLE CON STATIC
ENDS01	SLU2	010000A6	1	0	0	0	2	SIGN(IMSUS08 ) IDLE CON STATIC

```

LU6NDPA  LUT6
-LU6SPG      010000BB    0    0    0    0    0 IDLE CON PRI
-CA12        010000BD    0    0    0    0    0 IDLE CON PRI
-CA11        010000BF    0    0    0    0    0 IDLE CON PRI
LU6NDPH  LUT6
-CA14        010000C1    0    0    0    0    0 IDLE CON PRI
-N/A         00000000    0    0    0    0    0 IDLE STATIC
-N/A         00000000    0    0    0    0    0 IDLE STATIC
*90240/134730*

```

Explanation: User IMSUS05 is signed on to dynamic node DTSLU002. User IMSUS02 is signed on to dynamic node DTSLU202. User IMSUS04 is allocated to ISC node DTSLU603. User IMSUS14 is signed on to dynamic node DTSLU205 and has an active conversation. User IMSUS13 is signed on to dynamic node DTSLU204 and is in exclusive mode. User IMSUS12 is signed on to dynamic node DTSLU203 and has a preset destination of LTERM WTOR and is in MFS test mode. Static node LUP1 is logged on. User IMSUS08 is signed on to static node ENDS01. Users LU6SPG, CA12, and CA11 are allocated to static ISC node LU6NDPA, which means node LU6NDPA has 3 active parallel sessions. User CA14 is allocated to static ISC node LU6NDPH, which means node LU6NDPH has one active parallel session.

*Example 2 for /DISPLAY NODE command*

Entry ET:

```
/DISPLAY NODE NSLUTP1 NSLUTP6 NPLUTP6
```

Response ET:

NODE-USR	TYPE	CID	RECD	ENQCT	DEQCT	QCT	SENT	
NSLUTP1	SLU1	00000000	0	0	0	0	0	IDLE
NSLUTP6	LUT6	00000000	5	3	2	1	5	IDLE
NPLUTP6	LUT6							
-A		00000000	1	1	1	0	1	IDLE FORCE
-B		00000000	10	25	13	12	9	IDLE
-N/A		00000000	0	0	0	0	0	IDLE
-C		00000000	0	4	4	0	4	IDLE

\*90229/083059\*

Entry ET:

```
/DISPLAY NODE ALL
```

Response ET:

NODE-USR	TYPE	CID	RECD	ENQCT	DEQCT	QCT	SENT	SYS3
WRIGHT	SLU2	00000000	0	0	0	0	0	IDLE
LUNS01	NT0	00000000	0	0	0	0	0	IDLE
WS12	FIN	00000000	0	0	0	0	0	IDLE
VR3275A1	3277	00000000	0	0	0	0	0	IDLE
VR3286A1	3286	00000000	0	0	0	0	0	IDLE
LUP1	SLUP	00000000	0	0	0	0	0	IDLE
Q3290A	SLU2	00000000	0	0	0	0	0	IDLE
LU6FP02	LUT6	00000000	0	0	0	0	0	IDLE
LU6NDPA	LUT6							
-N/A		00000000	0	0	0	0	0	IDLE
-N/A		00000000	0	0	0	0	0	IDLE
-N/A		00000000	0	0	0	0	0	IDLE
NDNTOXA1	NT0	00000000	0	0	0	0	0	ASR
ND3287A1	SLU1	00000000	0	0	0	0	0	ASR
ND3287A2	SLU1	00000000	0	0	0	0	0	IDLE
ND3287A3	SLU1	00000000	0	0	0	0	0	IDLE
ND3287A4	SLU1	00000000	0	0	0	0	0	ASR
NDSLUA1	SLU2	00000000	0	0	0	0	0	ASR

```

NDSL2A2 SLU2 00000000 1 1 0 1 1 ASR
NDSLUPA1 SLUP 00000000 0 0 0 0 0 ASR
NDSLUPA2 SLUP 00000000 0 0 0 0 0 IDLE
NDLU6XSG LUT6 00000000 0 0 0 0 0 IDLE
NDLU6XPG LUT6
-N/A 00000000 0 0 0 0 0 ASR
*91213/100107* SYS3

```

Entry ET:

```
/DISPLAY NODE IN902D1 IN902D4
```

Response ET:

```

NODE-USR TYPE  CID      RECD ENQCT  QCT  SENT
IN902D1  SLU1  100121C5  125  125    0  125 CON
*91012/192544*

```

Entry ET:

```
/DISPLAY NODE NLUTP6P USER C
```

Response ET:

```

NODE-USR TYPE  CID      RECD ENQCT DEQCT  QCT  SENT
NLUTP6P  LUT6
-C 00000000 0 10 8 2 8
*90229/083059*

```

### *Example 3 for /DISPLAY NODE command*

The following commands illustrate the use of /DISPLAY to monitor changes in the mode table values for a node.

Entry ET:

```
/DISPLAY NODE LUTYPEP1 MODE
```

Response ET:

```

NODE-USR TYPE  DEF MODETBL  ACT MODETBL
LUTYPEP1  SLUP  DEFRESP
*90179/100206*

```

Explanation: DEFRESP is the mode table name defined for node LUTYPEP1 at system definition. The session is not active so the ACT MODETBL field is blank.

Entry ET:

```
/OPN NODE LUTYPEP1
```

Response ET:

```
DFS058I OPNDST COMMAND COMPLETED
```

Entry ET:

```
/DISPLAY NODE LUTYPEP1 MODE.
```

Response ET:

```

NODE-USR TYPE  DEF MODETBL  ACT MODETBL
LUTYPEP1  SLUP  DEFRESP  DEFRESP
*90179/100508*

```

Explanation: A mode table name was not specified with the /OPNDST command so the default value defined at system definition was used to initiate the session.

Entry ET:

```
/CLS NODE LUTYPE1
```

Response ET:

```
DFS058I CLSDST COMMAND COMPLETED
```

Entry ET:

```
/DISPLAY NODE LUTYPE1 MODE
```

Response ET:

```
NODE-USR TYPE  DEF MODETABL  ACT MODETABL
LUTYPE1 SLUP  DEFRESP
*90179/100630*
```

Explanation: Active mode table name displays as blank at normal session termination.

Entry ET:

```
/OPN NODE LUTYPE1 MODE ALPHA.
```

Response ET:

```
DFS058I OPNDST COMMAND COMPLETED
```

Entry ET:

```
/DISPLAY NODE LUTYPE1 MODE
```

Response ET:

```
NODE-USR TYPE  DEF MODETABL  ACT MODETABL
LUTYPE1 SLUP  DEFRESP      ALPHA
*90179/100805*
```

Explanation: The mode table name specified with the /OPNDST command (ALPHA) is used to initiate the session. The default value specified at system definition (DEFRESP) is overridden by the /OPNDST command.

#### *Example 4 for /DISPLAY NODE command*

The following example shows the display of all nodes with the automatic session restart designation.

Entry ET:

```
/DISPLAY NODE AUTOSR
```

Response ET:

NODE-USR	TYPE	CID	RECD	ENQCT	DEQCT	QCT	SENT
WS12	FIN	00000000	0	0	0	0	0 ASR
NODE0005	FIN	00000000	0	0	0	0	0 ASR
NODE0006	FIN	00000000	0	0	0	0	0 ASR
NODE0007	FIN	00000000	0	0	0	0	0 ASR
NTOLUNS	NTO	00000000	0	0	0	0	0 ASR IDLE
NT02741	NTO	00000000	0	0	0	0	0 ASR IDLE
LU6NODE2	LUT6						

```

-N/A          00000000    0    0    0    0    0 ASR IDLE
-N/A          00000000    0    0    0    0    0 ASR IDLE
-N/A          00000000    0    0    0    0    0 ASR IDLE
*90179/101609*

```

**Example 5 for /DISPLAY NODE command**

Entry ET:

```
/DISPLAY NODE NODE1 NODE2 QCNT
```

Response ET:

```

NODE          GBLQCT
NODE1         100000
NODE2         25
*95200/170817*

```

Entry ET:

```
/DISPLAY NODE NODE1 USER USER1 USER2 QCNT
```

Response ET:

```

NODE/USER     GBLQCT
NODE1
-USER1        10
-USER2        0
*95200/170817*

```

**Example 6 for /DISPLAY NODE command**

Entry ET:

```
/DISPLAY NODE FPEN0001 EMHQ QCNT
```

Response ET:

```

LTERM         GBLQCT
FPEN0001      0
*98203/143542*

```

**Example 7 for /DISPLAY NODE command**

Entry ET:

```
/DISPLAY NODE DTSLU* DTSLU202 L3270A RECOVERY
```

Response ET:

```

NODE-USR OWNER SRM   CONV STSN FPATH
DTSLU002 IMSA  GLOBAL Y   Y   Y
DTSLU603
-IMSUS04 IMSA  LOCAL  Y   Y   Y
DTSLU202 IMSB  GLOBAL Y   N   N
L3270A   IMSA  LOCAL  Y   N   Y
*99240/134730*

```

**Example 8 for /DISPLAY NODE command: ISC TCP/IP nodes**

In the following example, the value TCP in the TYPE column of the command response indicates that the node CIC4 is an ISC TCP/IP node. The string of



## Examples

Entry ET:

```
/DISPLAY OASN SUBSYS ALL
```

Response ET:

```
      SUBSYS  OASN          ESSTYPE
      V10A           0000000003  DB2
                   0000000004
      V10B           0000000005  DB2
*13242/145734*
```

TSO SPOC input:

```
DIS OASN SUBSYS ALL
```

TSO SPOC output:

```
Log for . . : DIS OASN SUBSYS ALL
```

```
IMSpIex . . . . . : PLEX1
Routing . . . . . : SYS3
Start time. . . . : 2013.256 16:01:46.39
Stop time . . . . : 2013.256 16:01:46.48
Return code . . . : 00000000
Reason code . . . : 00000000
Reason text . . . :
Command master. . : SYS3
```

```
MbrName  Messages
-----
SYS3     SUBSYS  OASN          ESSTYPE
SYS3     V10A           0000000003  DB2
SYS3           0000000004
SYS3     V10B           0000000005  DB2
SYS3           0000000005
SYS3     *13256/160146*
```

---

## /DISPLAY OLDS command

Use the /DISPLAY OLDS command to display the system logging status.

The ddnames for the online log data sets that are currently online are listed in the sequence used. That is, the one currently being used is listed first, with the one most recently used listed next, and so on. OLDS status information is displayed with each ddname. The meanings of these status messages is provided. In the case of dual logging, two lines of status output are displayed per OLDS ID; the first for the primary OLDS and the second for the secondary OLDS.

Subsections:

- "Environment" on page 650
- "Syntax" on page 650
- "Keywords" on page 650
- "Usage notes" on page 650
- "Output fields" on page 651
- "Examples" on page 652

## Environment

The /DISPLAY OLDS command is valid in DB/DC, DBCTL, and DCCTL environments.

## Syntax



## Keywords

The following keywords can be specified for the /DISPLAY OLDS command. These keywords specify selection criteria used to select which OLDS are to be included in the display output. All these keywords are optional. If none are specified, ALL is the default action. When multiple keywords are specified, an OLDS that satisfies any one of the specified criteria is included in the output.

### ALL

Displays status of each online OLDS.

### INUSE

Displays current output OLDS.

### BACKOUT

Displays OLDS that contains dynamic backout data for currently active PSTs.

### ERROR

Displays OLDS that have encountered error conditions.

### STOPPED

Displays stopped OLDS.

## Usage notes

/DISPLAY OLDS shows the percentage of the OLDS used (shown under % FULL in the output) and the rate at which it is being filled (shown under RATE in the output). The RATE is the average number of blocks written to the current OLDS since the last /DISPLAY OLDS was issued. If no blocks were written, the RATE is 0, or, if elapsed time is 0, the previous RATE will be displayed.

When the archive status of an OLDS is scheduled or started and DBRC has generated the JCL for the archive job, the archive job name will be included in the output.

Following the detailed OLDS data set information, the dual OLDS state (DUAL, DEGRADED DUAL, NONDEGRADABLE DUAL or SINGLE) and the WADS recording state (DUAL, SINGLE, or NO) are displayed. Also, the DDNAMEs of all available WADS are displayed. The DDNAMEs of the current WADS are preceded by an asterisk (\*).

The SLDSREAD status is also displayed. If SLDSREAD ON is displayed, IMS is enabled to retrieve both SLDS and OLDS data. If SLDSREAD OFF is displayed, IMS is not enabled to retrieve SLDS data.

If the backout of one or more active dependent regions would require the SLDS (all of the region's log records are not available from the OLDS), the following text

is displayed in front of the SLDSREAD ON or SLDSREAD OFF character string: SLDS REQUIRED FOR BACKOUT - RGN *nnnnn*, where *nnnnn* is the region number of the dependent region that owns the oldest record required for backout.

If there are one or more active dependent regions that could potentially require backout, but the SLDS is not required, the following text is displayed: OLDS REQUIRED FOR BACKOUT - RGN *nnnnn*, where *nnnnn* is the region number of the dependent region that owns the oldest record required for backout.

An OLDS that is stopped or had an I/O error is dynamically deallocated after it is no longer needed for backout.

The /DISPLAY OLDS command also shows the following fields:

**BSN** Represents a block of data that is written to the OLDS.

**LSN** Represents an individual log record. An underscore is printed to separate the two halves of the 8-byte LSN for readability even though the LSN does not contain an underscore.

You can use the LSN and BSN values to determine whether they are approaching their finite limits. The limit for the BSN is X'FFFFFFFF' and the limit for the LSN is X'FFFFFFFF\_FFFFFFFF'.

**Restriction:** Under the dual logging environment, the number of primary OLDS and secondary OLDS must be the same; otherwise, the status of the OLDS is incorrect.

## Output fields

The /DISPLAY OLDS command shows one of the following as the archive status of each OLDS:

### NEEDED

This OLDS needs to be archived. It cannot be reused for output logging until it has been archived.

### SCHEDULED

An archive job has been generated for this OLDS.

### STARTED

A job to archive this OLDS has started execution.

### AVAILABLE

Either this OLDS is empty or it has been archived. It can be reused for output logging.

The /DISPLAY OLDS command also shows the following status information for OLDS:

### IN USE

This is the current output log OLDS. The display line for this OLDS also has an asterisk (\*) in the left margin.

### STOPPED

/STOP command entered for this OLDS or the OLDS is being internally stopped because of write errors.

### WRT-ERR

A write I/O error occurred on this OLDS data set.

**CLSE** An error occurred when closing this data set. The Log Recovery utility can be used to correct this condition.

**PREV CLSER**

The previous OLDS could not be closed because of an I/O error and this OLDS is required by the Log Recovery utility to correct the condition.

**NOT USABLE**

This data set is not being used for logging. When operating in “Degraded Dual” state, the other data set in this OLDS pair was used for output but this data set was not used because of a previous data set error.

**BACKOUT**

This OLDS is potentially required for backout.

**Examples**

Entry ET:

/DISPLAY OLDS

Response ET:

```
OLDS-DDNAME % FULL RATE ARCH-JOB    ARCH-STATUS  OTHER-STATUS
*DFSOLP02      8      0                AVAILABLE    IN USE
DFSOLP01                AVAILABLE
DFSOLP00                AVAILABLE
DFSOLP03                AVAILABLE
SINGLE OLDS LOGGING, SINGLE WADS LOGGING
AUTOMATIC ARCHIVE = 01
WADS = *DFSWADS0 DFSWADS1
SLDSREAD ON
CURRENT BSN = 0000005E, LSN = 00000000_00000580
*13164/112613*
```

---

**/DISPLAY OTMA command**

Use the /DISPLAY OTMA command to display the current status for IMS Open Transaction Manager Access (OTMA) clients and servers.

Subsections:

- “Environment”
- “Syntax”
- “Usage notes” on page 653
- “Output fields” on page 653
- “Examples” on page 655

**Environment**

The /DISPLAY OTMA command is valid in DB/DC and DCCTL environments.

**Syntax**



## Usage notes

This command is functionally equivalent to the /DISPLAY TMEMBER ALL command.

If a super member is defined, information about the super member is also displayed. Because a super member does not interface with z/OS cross-system coupling facility (XCF), the XCF\_STATUS field is left blank. Because super members do not process input messages, OTMA security authorization is not performed. The SECURITY field is also left blank.

## Output fields

This command displays the following output fields:

### ACEEAGE

The accessor environment element (ACEE) aging value for the OTMA client. This value is used to perform the refresh for cached OTMA ACEEs for all user IDs.

If the /SECURE OTMA ACEEAGE command is issued to specify a global ACEE aging value, this field displays the aging value that is specified with the /SECURE OTMA ACEEAGE command for the OTMA server.

### ACEECT

The total number of cached OTMA ACEEs for the OTMA server.

### DRUEXIT

The name of any OTMA Destination Resolution exit routine that is associated with the tmember.

### GROUP/MEMBER

Each member in each XCF group. The server is always the first member that is displayed.

**INPT** The maximum number of input messages from this tmember that can be waiting at the same time to be processed. This value is primarily used for send-then-commit or CM1 messages.

### MAXTP

The maximum TPIPE limit that is specified in the OTMA client descriptor. If the MAXTP parameter is specified for DFSOTMA in the OTMA client descriptor, the MAXTP limit is displayed for the OTMA server. If there are one or more OTMA clients with the MAXTP limit defined and MAXTP is not specified for DFSOTMA, the highest limit defined among the members is displayed for the OTMA server.

### SECURITY

The security status for each member or server. The security status can be one of the following:

- NONE
- JOIN
- CHECK
- FULL
- PROFILE

### SMEM

The name of the super member.

**TIB** For an OTMA target member (tmember), the current number of input

messages from this tmember that IMS is processing. For an OTMA server, the total number of input messages from all tmembers that IMS is processing plus the reserved message blocks for message processing. This value is primarily used for send-then-commit or CM1 messages.

For the OTMA internal member DFSYICAL, this field shows the number of IMS regions that are waiting for responses to synchronous program switch requests.

**T/O** The timeout value for OTMA to wait for an ACK or NACK for an OTMA output message.

#### **TOACEE**

Indicates whether the TOACEE= parameter on the DFSOTMA client descriptor is YES to enable ACEE flood control or NO to disable ACEE flood control. This status is valid only for the OTMA server.

#### **TPCNT**

The maximum TPIPE count for each OTMA client.

For an OTMA client other than the IMS server, this field indicates the current TPIPE count for that member. For the IMS server, this field indicates the global TPIPE limit, which is the highest TPIPE limit set for the OTMA clients. IMS displays a nonzero value only when the MAXTP= keyword is specified for this member in the OTMA client descriptor DFSYDTx member of the IMS PROCLIB data set. If the MAXTP= keyword is not specified for any of the members in the OTMA client descriptor, the TPCNT field of the IMS server member is zero.

#### **USER-STATUS**

The user status for each member. The status can be one of the following:

- ACCEPT TRAFFIC
- DISCONNECTED
- FLOOD
- IN SLOWDOWN
- MAX TPIPE
- SERVER
- SERVER+FLOOD
- SMQ BACKEND
- SYNC P2P
- SYNC P2P+FLOOD
- WAIT BID
- WAIT RESPONSE

#### **XCF-STATUS**

The XCF status for each member. The status can be one of the following:

- ACTIVE
- CREATED
- FAILED
- NOT DEFINED
- QUIESCED
- STO-INPUT
- UNKNOWN

## Examples

The following are examples of the /DISPLAY OTMA command.

### *Example 1 for /DISPLAY OTMA command*

Entry ET:

```
/DISPLAY OTMA
```

Response ET:

GROUP/MEMBER	XCF-STATUS	USER-STATUS	SECURITY	TIB	INPT	SMEM
	DRUEXIT	T/O				
HARRY						
-IMS1	ACTIVE	SERVER	FULL			
-IMS1	N/A					
-HWS001	ACTIVE	ACCEPT TRAFFIC	FULL	0	0	SM01
-HWS001	HWSYDRU0	5				
-SM01		SUPER MEMBER				SM01
-SM01	N/A					
-MQS001	ACTIVE	ACCEPT TRAFFIC	FULL	0	0	
-MQS001	MQSYDRU0	10				

Explanation: The /DISPLAY OTMA command can be issued to display the status of IMS server and members. This command is functionally equivalent to the /DIS TMEMBER ALL command. The member information is displayed in random order. In the previous example, HARRY is the XCF group name for IMS and its OTMA clients. IMS1 is the server member running with OTMA activated. HWS001 and MQS001 are OTMA client members. The SM01 is an OTMA super member for asynchronous output messages. Each OTMA client has two-line status information.

### *Example 2 for /DISPLAY OTMA command*

Entry ET:

```
/DISPLAY OTMA
```

Response ET:

GROUP/MEMBER	XCF-STATUS	USER-STATUS	SECURITY
HOANG			
-APPL8	NOT DEFINED	SERVER	FULL
-CLIENT1	ACTIVE	ACCEPT TRAFFIC	
*94165/165753*			

### *Example 3 for /DISPLAY OTMA Command*

Entry ET:

```
/DISPLAY OTMA
```

Response ET:

ROUP/MEMBER	XCF-STATUS	USER-STATUS	SECURITY	TIB	INPT	SMEM
HARRY						
-HWS001	ACTIVE	ACCEPT TRAFFIC	FULL	0	0	SM01
-HWS002	ACTIVE	ACCEPT TRAFFIC	FULL	0	0	SM01
-SM01		SUPER MEMBER				SM01
-HWS003	ACTIVE	ACCEPT TRAFFIC	FULL	0	0	

Explanation: The /DISPLAY OTMA command can be issued to display the status for OTMA clients and servers. This command is functionally equivalent to the

/DIS TMEMBER ALL command. The member information is displayed in random order.

*Example 4 for /DISPLAY OTMA command*

Entry ET:

/DISPLAY OTMA

Response ET:

GROUP/MEMBER	XCF-STATUS DRUEXIT	USER-STATUS T/O ACEEAGE	SECURITY	TIB	INPT	SMEM
XCFGRP1						
-IMS1	ACTIVE	SERVER	FULL		8000	
-IMS1	N/A	0				
-HWS1	ACTIVE	ACCEPT TRAFFIC	FULL	0	5000	
-HWS1	HWSYDRU0	239 3600				
-HWS2	ACTIVE	ACCEPT TRAFFIC	FULL	0	5000	
-HWS2	HWSYDRU0	239 7200				
-HWS3	ACTIVE	ACCEPT TRAFFIC	FULL	0	5000	
-HWS3	HWSYDRU0	239 0				
*09121/172200*	IMS1					

Explanation: The /DISPLAY OTMA command can be issued to display the ACEE aging value for the OTMA client or member. OTMA uses this value to determine if a cached ACEE for a user ID needs to be refreshed. In this example, the aging value for HWS1 is 3600 seconds, and the aging value for HWS2 is 7200 seconds. Any cached ACEE that only HWS1 uses has an aging value of 3600 seconds, and any cached ACEE that only HWS2 uses has an aging value of 7200 seconds. If a cached ACEE is used by both HWS1 and HWS2, the aging value is 3600 seconds, which is the lowest value between 3600 and 7200 seconds. Because no aging value is specified for HWS3, IMS always creates a non-cached ACEE for users from HWS3.

*Example 5 for /DISPLAY OTMA command*

Entry ET:

/DISPLAY OTMA

Response ET:

GROUP/MEMBER	XCF-STATUS DRUEXIT	USER-STATUS T/O TPCNT ACEEAGE	SECURITY	TIB	INPT	SMEM
XCFGRP1						
-IMS1	ACTIVE	SERVER	FULL	0	8000	
-IMS1	N/A	200 3000				
-HWS1	ACTIVE	MAX TPIPE	FULL	0	5000	
-HWS1		200 1000 999999				
-HWS2	ACTIVE	ACCEPT TRAFFIC	FULL	0	5000	
-HWS2		200 500 999999				

Explanation: In this example, the TPCNT field indicates the current TPIPE count for each member. For the IMS server, this field indicates the global TPIPE limit, which is the highest TPIPE limit set for the OTMA clients. MAX TPIPE indicates that the TPIPE count for the OTMA client has reached the limit set. This status is cleared when the global TPIPE count falls below 50%.

*Example 6 for /DISPLAY OTMA command*

Entry ET:

/DISPLAY OTMA

Response ET:

GROUP/MEMBER	XCF-STATUS DRUEXIT	USER-STATUS T/O	SECURITY	TIB	INPT	SMEM
IMSXCFGP						
-IMS1	ACTIVE	SERVER	FULL	30	8000	
-IMS1	N/A					
-HWS001	ACTIVE	ACCEPT TRAFFIC	FULL	20	5000	
-HWS001	HWSYDRU0	5				
-DFSYICAL		SYNC P2P	FULL	10	5000	
-DFSYICAL	N/A	5				

In this example, IMSXCFGP is the XCF group name for IMS and its OTMA clients. The server member name for OTMA is IMS1. HWS001 and DFSYICAL are the names of the OTMA client members. The DFSYICAL member is an internal OTMA client that processes synchronous program switch requests initiated by DL/I ICAL calls from application programs. The DFSYICAL member has the USER-STATUS of SYNC P2P. The TIB field for the DFSYICAL member shows that there are 10 IMS regions waiting for responses to DL/I ICAL synchronous program switch requests. The HWS001 member is a standard OTMA client member, connected via XCF, that has 20 active messages. The total TIB for this IMS system is 30. None of the members are using an OTMA Destination Resolution exit routine.

#### *Example 7 for /DISPLAY OTMA command*

Entry ET:

/DISPLAY OTMA

Response ET:

GROUP/MEMBER	XCF-STATUS DRUEXIT	USER-STATUS T/O	ACEEAGE	SECURITY	TIB	INPT	SMEM
XCFGRP1							
-IMS1	ACTIVE	SERVER+FLOOD		FULL	11000	10000	
-IMS1	N/A	0					
-HWS1	ACTIVE	ACCEPT TRAFFIC		FULL	3000	5000	
-HWS1	HWSYDRU0	120 0					
-HWS2	ACTIVE	ACCEPT TRAFFIC		FULL	3000	5000	
-HWS2	HWSYDRU0	120 0					
-HWS3	ACTIVE	ACCEPT TRAFFIC		FULL	4000	5000	
-HWS3	HWSYDRU0	120 0					
*09121/172200*	IMS1						

The /DISPLAY OTMA command can be issued to display the active OTMA messages (TIB) in the system and in each OTMA client. The flood limit (INPT) for the entire system and each OTMA client can also be displayed. In this example, because the global flood limit of the system is set to 10000, the total active OTMA messages (3000 + 3000 + 4000) is 11000, which is over the global limit. The USER-STATUS shows SERVER+FLOOD to indicate that the global flood limit was specified in client descriptor or in the /START TMEMBER ALL INPUT command, and IMS has reached this OTMA global flood limit. The applications that use OTMA clients HWS1, HWS2, and HWS3 must reroute the OTMA transaction to a different IMS. All the new input transactions from all the OTMA clients are rejected with OTMA sense code x'0030' and reason code x'0002'.

**Related reference:**

“/DISPLAY TMEMBER command” on page 721

---

## /DISPLAY OVERFLOWQ command

Use the /DISPLAY OVERFLOWQ command to display a list of queue names that are in overflow mode for those coupling facility list structures specified (for the STRUCTURE keyword) that are in overflow mode.

It also displays the queue type of the resource for each queue.

Subsections:

- "Environment"
- "Syntax"
- "Keywords"
- "Usage notes"
- "Output fields" on page 659
- "Examples" on page 660

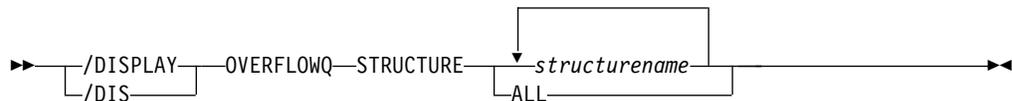
### Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

Table 297. Valid environments for the /DISPLAY OVERFLOWQ command and keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
/DISPLAY	X	X	X
OVERFLOWQ	X		X
STRUCTURE	X		X

### Syntax



### Keywords

The following keywords can be specified for the /DISPLAY OVERFLOWQ command:

#### STRUCTURE

Specifies the primary coupling facility list structures to be displayed.

### Usage notes

If a specified structure is not defined, the overflow structure name is displayed as NOT-DEFINED.

Queue names displayed might not be defined to the IMS subsystem on which this command is issued. If a queue type is not defined, the type is displayed as UNKNOWN, and the queue name is displayed as the 16-byte queue name (with the queue type as the first byte).

This command is valid only in a shared-queues environment.

If there are OTMA output messages that have been moved to the overflow structure, the `/DISPLAY OVERFLOWQ` command displays information about those messages. Instead of displaying the queue name that was used to queue the OTMA output messages, the character string 'TMSTMP-' is displayed. If the messages are queued to a regular member, the IMS ID of the IMS subsystem to which the messages have affinity is displayed following the 'TMSTMP-' string. If the messages are queued to a super member, the field following the 'TMSTMP-' string is left blank because output messages queued to a super member do not have affinity to any IMS subsystem. If the command is entered on a system where the TMEMBER and TPIPE are defined, the TMEMBER and TPIPE names are displayed. If the command is entered on a system where the TMEMBER and TPIPE are not defined, the character string 'UNKNOWN' is placed in the TMEMBER and TPIPE fields.

## Output fields

The following output fields are displayed by the `/DISPLAY OVERFLOWQ` command:

**APPC** The resource type on the overflow structure is an APPC outbound queue. The first eight bytes of the resource name displayed is the unique constant TMSTMP-. The next four bytes are the IMS ID of the IMS subsystem that has the LUNAME and TPNAME defined. If you issue the `/DISPLAY OVERFLOWQ` command on the system on which the LUNAME and TPNAME are defined, then the LUNAME and TPNAME also display.

When LUNAME and TPNAME cannot be located on the system on which they're defined, UNKNOWN appears under the LUNAME and TPNAME headings in the display output.

### BALGRP

The resource type on the overflow structure is a Fast Path PSB name. The resource name displayed is the Fast Path PSB name.

### LTERM

The resource type on the overflow structure is an LTERM. The resource name displayed is the LTERM name.

### OTMA

The resource type on the overflow structure is an OTMA outbound queue. The first eight bytes of the resource name displayed is the unique constant TMSTMP-. The next four bytes are the IMS ID of the IMS subsystem that has the TMEMBER and TPIPE defined. If you issue the `/DISPLAY OVERFLOWQ` command on the system on which the TMEMBER and TPIPE are defined, then the TMEMBER and TPIPE are also displayed.

When LUNAME and TPNAME cannot be located on the system on which they're defined, UNKNOWN appears under the LUNAME and TPNAME headings in the display output.

### REMOTE

The resource type on the overflow structure is a remote LTERM, remote transaction, or an MSNAME. The resource name displayed is the remote LTERM name, remote transaction name, or the msname.

### SUSPENDTRAN

The resource type on the overflow queue is a suspended transaction and the messages are on the suspend queue. The resource name displayed is

the transaction name. If the transaction that is suspended is a serial transaction, or if local affinity has been set for a non-serial transaction, the first eight bytes displayed are the transaction name and the last seven bytes represent the IMS ID of the subsystem that enqueued the message.

### SERIALTRAN

The resource type on the overflow queue is a serial transaction. The first eight bytes of the resource name are the transaction name and the last seven bytes represent the IMS ID of the IMS subsystem that inserted the message for the transaction.

### TRANSACTION

The resource type on the overflow queue is a transaction. The resource name displayed is the transaction name. If local affinity has been set for the transaction, the first eight bytes displayed are the transaction name and the last seven bytes represent the IMS ID of the IMS subsystem that enqueued the message.

### UNKNOWN

The resource type of the resource on the overflow queue is not known to this IMS subsystem. The 16 byte resource name is displayed.

## Examples

The following are examples of the /DISPLAY OVERFLOWQ command:

### *Example 1 for /DISPLAY OVERFLOWQ command*

Entry ET:

```
/DISPLAY OVERFLOWQ STRUCTURE ALL
```

Response ET:

STRUC-RSCTYPE	OFLSTRUC-RSCNAME	LUNAME-TMEMBER	TPNAME-TPIPE
IMMSGQ01	IMMSGQ01OFLW		
TRANSACTION	TRANA		
LTERM	LTERMX		
SUSPENDTRAN	TRANA		
APPC	TMSTMP- IMSA	LUNAME1	TPNAME1
SERIALTRAN	TRANABC IMSA		
APPC	TMSTMP- IMSA		
OTMA	TMSTMP- IMSA	TMEMBERA	TPIPEA
APPC	TMSTMP- IMSA	NETWORK2.LUNAME2	TPNM123456789012+ 3456789012345678+ 901234567890
OTMA	TMSTMP- IMSB		
REMOTE	TRANB		
REMOTE	LTERMA		
UNKNOWN	ASIXTEENCHARNAME		
REMOTE	MSNAMEA		
IMSEMHQ01	OVERFLOW STRUCTURE IS NOT DEFINED		
BALGRP	FPPSB1		
LTERM	LTERMY		
BALGRP	FPPSB2		
*95200/170817*			

### *Example 2 for /DISPLAY OVERFLOWQ command*

Entry ET:

```
/DIS OVERFLOWQ STRUCTURE ALL
```

Response ET:

```
STRUC-RSCTYPE OFLSTRUC-RSCNAME LUNAME-TMEMBER TPNAME-TPIPE
IMSMGQ01      IS NOT IN OVERFLOW MODE
IMSEMQ01      IMSEMQ010FLW
BALGRP        DDLTRN14
*04302/165656*
```

*Example 3 for /DISPLAY OVERFLOWQ command*

Entry ET:

```
/DIS OVERFLOWQ STRUCTURE IMSEMQ01
```

Response ET:

```
STRUC-RSCTYPE OFLSTRUC-RSCNAME LUNAME-TMEMBER TPNAME-TPIPE
IMSEMQ01      IMSEMQ010FLW
BALGRP        DDLTRN14
*04302/165834*
```

*Example 4 for /DISPLAY OVERFLOWQ command*

Entry ET:

```
/DISPLAY OVERFLOWQ STRUCTURE AL
```

Response ET:

```
STRUC-RSCTYPE OFLSTRUC-RSCNAME LUNAME-TMEMBER TPNAME-TPIPE
IMSMGQ01      IMSMSGQ010FLW
OTMA          TMSTMP-          SM01          TPIPE1
OTMA          TMSTMP-          UNKNOWN        UNKNOWN
OTMA          TMSTMP- SYS3     HWS003        TPIPE1
IMSEMQ01      IS NOT IN OVERFLOW MODE
```

Explanation: This example shows that messages queued to TPIPE1 of super member SM01 have been moved to the overflow structure. Messages queued to TPIPE1 of regular member HWS003 have also been moved to the overflow structure. Messages for a TMEMBER or TPIPE that is unknown to the IMS that processed the command have also been moved to the overflow structure.

---

## **/DISPLAY PGM command**

Use the /DISPLAY PGM command to display the status of programs.

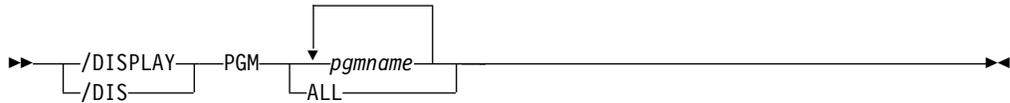
Subsections:

- "Environment"
- "Syntax" on page 662
- "Usage notes" on page 662
- "Equivalent IMS type-2 commands" on page 662
- "Output fields" on page 662
- "Examples" on page 662

### **Environment**

The /DISPLAY PGM command is valid in DB/DC, DBCTL, and DCCTL environments.

## Syntax



## Usage notes

For DBCTL, this command displays the status of BMP programs, DEDB utilities, and CCTL PSBs. Because DBCTL has no knowledge of CICS programs, their status is not displayed.

## Equivalent IMS type-2 commands

The following table shows variations of the /DISPLAY PGM command and the IMS type-2 commands that perform similar functions.

Table 298. Type-2 equivalents for the /DISPLAY PGM command.

Task	/DISPLAY PGM command	Similar IMS type-2 command
Displays information about a program.	/DISPLAY PGM <i>pgmname</i>	QRY PGM NAME( <i>pgmname</i> ) SHOW(ALL)
		QUERY PGM NAME( <i>pgmname</i> ) SHOW(TRAN)

## Output fields

The status displayed can be one of the following:

DB-STOPD  
I/O PREVEN  
LOCK  
NOTINIT  
STOPPED  
TRA

## Examples

Entry ET:

```
/DISPLAY PGM APOL1 DBFSAMP3 BMP255
```

Response ET:

```
PROGRAM      TRAN      TYPE
APOL1        APOL11   TP
              APOL12
              APOL13
              APOL14
              APOL15
              APOL16
              APOL17
              APOL18
DBFSAMP3     FPSAMP1   FPM
BMP255       TRAN255   BMP
              TXCD255
              TXCD255W
*90288/140450*  SYS3
```

Entry ET:

/DISPLAY PGM ALL

Response ET:

PROGRAM	TRAN	TYPE	
AD2TP	TSTAD2A	TP	NOTINIT
	TSTAD2R1		
APOL1	APOL11	TP	NOTINIT
	APOL12		
A3270	A3270	TP	NOTINIT
	3270S		
BMAAJK41	BHE4	TP	NOTINIT
	NQE4		
	SHE4		
BMP255	TRAN255	BMP	NOTINIT
BTAEJK01		BMP	NOTINIT
BTAPJK01		BMP	NOTINIT
BTAPJK11		BMP	NOTINIT
BTAPJK21		BMP	NOTINIT
BTGGJK01		BMP	NOTINIT
BTGGJK11		BMP	NOTINIT
BTGOJK01		BMP	NOTINIT
BTGRJK05		BMP	NOTINIT
CPGM1V0	CONV11V0	TP	NOTINIT
	CONV12V0		
	CONV13V0		
DBF#FPU0		FPU	
DBFSAMP3	FPSAMP1	FPM	
DBFSAMP4	FPSAMP2	TP	NOTINIT
DBFSAM22	TP1	FPM	
DBFSAM99	TP2	TP	NOTINIT
DCLECH0S	DCL	TP	NOTINIT
DDLTP04	TXCDBP04	FPM	
DDLTP05	TXCDBP05	FPM	
DDLTP06	TXCDBP06	FPM	
DDLTP07	TXCDBP07	FPM	NOTINIT
DDLTP08	TXCDBP08	FPM	NOTINIT
DDLTP09	TXCDBP09	FPM	NOTINIT
V2MPPA02	V2MRP01	TP	NOTINIT
	V2MRP02		
V2MPPC01	V2SNS01	TP	
	V2SRS02		
V2MPPC02	V2MRS04	TP	NOTINIT
V2MPPP01	V2SNS02	TP	NOTINIT
	V2SRS03		
	V2SRS04		
V2MPPP02	V2MRP03	TP	NOTINIT
	V2SRP01		
WTCPLI	TESTPLI	TP	NOTINIT

\*89184/142553\*

#### Related reference:

“Status and attributes for the /DISPLAY command” on page 523

 [QUERY PGM command \(Commands\)](#)

---

## /DISPLAY POOL command

Use the /DISPLAY POOL command to display processor storage utilization statistics for the various IMS storage pools.

Subsections:

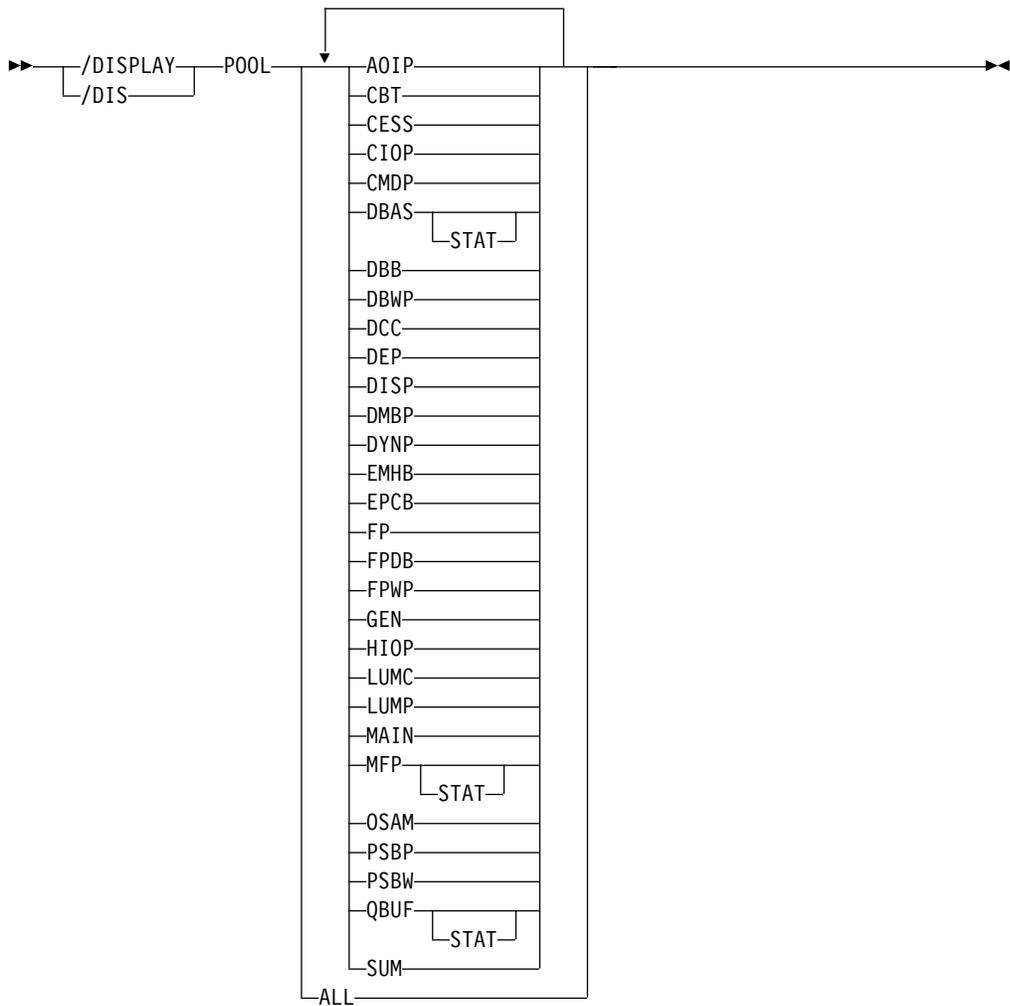
- “Environment” on page 664

- "Syntax"
- "Usage notes"
- "Equivalent IMS type-2 commands" on page 667
- "Abbreviations used in the /DISPLAY POOL command" on page 667
- "Examples" on page 673

## Environment

The /DISPLAY POOL command is valid in DB/DC, DBCTL, and DCCTL environments.

## Syntax



## Usage notes

When the /DISPLAY POOL command is issued from a specific environment (such as DBCTL), only the information that is valid for that environment is displayed.

Certain storage utilization statistics can be obtained with the DL/I STAT call.

Control block table (CBT) pools are defined in DFSCBT00 and are grouped into classes for the purposes of the /DISPLAY POOL command. All CBT pools are valid in all environments.

For a list of all of the IMS CBT pools with a brief description of each, see IMS Control Block Table (CBT) Pools (Diagnosis).

The classes and corresponding parameters are:

- CBT** This displays storage for all CBT pools. Use of this operand generates a large amount of display output.
- DBB** This displays database-related storage.
- DCC** This displays storage associated with DC.
- DEP** This displays storage associated with the dependent region structure. This parameter is not supported for an RSR tracking subsystem.
- DISP** This displays dispatcher-related storage.
- FP** This displays Fast Path-related storage.
- GEN** This displays storage that has no particular functional association.
- OSAM** This displays OSAM-related storage.
- SUM** This displays summary statistics for each of the above storage classes.

A CBT class display shows information for each pool in that class. The amount of storage displayed for these pools might change, because the storage is obtained as needed and can be freed when not in use. In addition, a summary of the total current global and local storage in the class and in all CBT pools is displayed.

If SUM is specified, a summary of the total current local and global storage for each CBT class is displayed.

For the following storage pools, the amount of storage obtained is specified during system definition or by way of EXEC statement overrides in the IMS procedure JCL.

- DBAS** Database buffer pools (includes VSAM buffer pools if VSAM is in the system, and also includes buffer space used by sequential buffering).  
  
Statistics for VSAM local shared resource pools are displayed in the order in which the pools are defined. For each local shared resource pool, the statistics of each subpool are displayed in ascending order by buffer size. The smallest buffers are displayed first and the largest buffers are displayed last. If an index subpool exists in the shared resource pool, its statistics follow the data subpool statistics. Index statistics are also displayed in ascending order by buffer size.  
  
Headings for VSAM subpools include the pool ID of the local shared resource pool. They also indicate whether the subpool is a data subpool (type D) or an index subpool (type I).  
  
The output from issuing the /DIS POOL DBAS command is equivalent to the output from issuing the type-2 QUERY POOL TYPE(DBAS) SHOW(STATISTICS) command.

- DBWP** Database working pool.
- DMBP** Data management block pool.
- EPCB** Fast Path PCB extension pool for MPPs.
- FPDB** Fast Path database buffer pool.
- MAIN** Working storage pool.
- MFP** Message format block pool.
- PSBP** Program specification block pool.  
This parameter is not supported for an RSR tracking subsystem.
- PSBW** Program specification block work area.  
This parameter is not supported for an RSR tracking subsystem.
- QBUF** Message queue buffer pool. Parameter QBUF can also be used to indicate how close the system is to automatic shutdown because of message queue buildup. The statistics displayed for each data set are:
- The maximum number of records available before initiation of automatic shutdown. This value is defined by the SHUTDWN parameter of the MSGQUEUE macro. This value is not the total number of records available in the data set.
  - The number of records currently in use.
  - The percentage of records currently in use to records available before initiation of automatic shutdown. When the value described in this item reaches 100%, an internal checkpoint DUMPQ will automatically be initiated. The number of records in the queue data set might continue to increase during the checkpoint DUMPQ, resulting in percentages greater than 100 being displayed.
  - In a shared-queues environment, two additional lines of output are displayed: the current buffer in-use count and the buffer in-use high water mark.

The following table shows the environments (DB/DC, DBCTL, or DCCTL) in which each storage pool is valid. If a pool is not valid in an environment, it will not appear when /DISPLAY POOL ALL is issued.

*Table 299. Storage pools and their environments*

Storage pools	DB/DC	DBCTL	DCCTL
AOIP	X	X	X
CESS	X	X	X
CIOP	X	X	X
CMDP	X	X	X
DBAS	X	X	
DBWP	X	X	
DMBP	X	X	
DYNP	X		X
EMHB	X	X	X
EPCB	X	X	X

Table 299. Storage pools and their environments (continued)

Storage pools	DB/DC	DBCTL	DCCTL
FPDB	X	X	
FPWP	X	X	
HIOP	X		X
LUMC	X		X
LUMP	X		X
MAIN	X	X	X
MFP	X		X
PSBP	X	X	X
PSBW	X	X	X
QBUF	X		X

**Note:** When the /DISPLAY POOL ALL command is specified for an RSR tracking subsystem, the following pools are not included: DEP, DYNP, LUMC, LUMP, PSBP, PSBW.

### Equivalent IMS type-2 commands

The following table shows variations of the /DISPLAY POOL command and the IMS type-2 commands that perform similar functions.

Table 300. Type-2 equivalents for the /DISPLAY POOL command

Task	/DISPLAY POOL command	Similar IMS type-2 command
Displays processor storage utilization statistics for OSAM and VSAM database buffer pools.	/DISPLAY POOL DBAS	QUERY POOL TYPE(DBAS) SHOW(STATISTICS)

### Abbreviations used in the /DISPLAY POOL command

The following lists indicate the abbreviations used in the /DISPLAY POOL command for each pool and subpool.

#### *CBT pool display*

**NAME**

Storage pool name

**SP**

Associated MVS user number

**CURR**

The current bytes allocated to the pool

**MAX**

Maximum size in bytes that the pool ever reached

**GETS**

Number of GETMAINS issued

**FREES**

Number of FREEMAINS issued

#### *Message queue pool*

**BFRS**

Number of processor storage queue buffers. In a shared queues environment, the number of buffers can be dynamically expanded from the number originally defined. This field displays the number of buffers currently available.

**SIZE** Usable size of one queue buffer, excluding the buffer prefix  
**ENQ** Number of message enqueues  
**DEQ** Number of message dequeues  
**CAN** Number of canceled messages  
**WAIT** Number of I/O waits issued  
**I/O** Number of I/O operations. In a shared-queues environment, it is the sum of CQSPUTs and CQSREADs.  
**ERR** Number of I/O errors

*Message format pool*

**SIZE** Pool size  
**SPACE** Maximum amount of free space  
**DIRS** Space used by MFS Dynamic Directory  
**REQ1** Number of block requests from pool  
**I/O** Number of fetch I/O operations  
**DIR** Number of directory I/O operations  
**WAIT** Number of immediate fetch I/O operations  
**FREE** Amount of free space currently in the pool  
**ERR** Number of I/O errors

*Database buffer pools*

The database buffer pool display consists of the OSAM buffer pool and, if VSAM is in the system, the VSAM buffer subpool. The display also includes information about the buffer space used by sequential buffering within the online IMS subsystem. The information about the sequential buffering buffer space includes:

**MAX** Maximum amount of space available for sequential buffering  
**FREE** Amount of currently free space for sequential buffering  
**CUR** The current bytes allocated to the pools for sequential buffering  
**HIGH** Highest amount of space ever used in this run for sequential buffering

*OSAM buffer subpool*

**This topic contains Product-sensitive Programming Interface information.**

Statistics for each OSAM subpool are displayed separately. The final display provides the statistical summation for all the OSAM subpools in all the shared resource pools.

**SIZE** Buffer pool size  
**REQ1** Number of block requests  
**REQ2** Number of requests satisfied in the pool plus new blocks created  
**READ** Number of read requests issued

**BISAM**

Number of BISAM reads issued plus QISAM SETLs

**WRITES**

Number of OSAM writes issued

**KEYC** Number of retrieve by key calls number

**LCYL** Number of OSAM format logical cylinder requests

**PURG** Number of synchronization calls received

**OWNRR**

Number of release ownership requests

**ERRORS**

Number of permanent errors now in the pool and largest number of permanent errors during this run

*VSAM buffer subpool*

**This topic contains Product-sensitive Programming Interface information.**

Statistics for each VSAM subpool are displayed separately. The final display provides the statistical summation for all the VSAM subpools in all the shared resource pools.

**BSIZE** The size of the buffers in this VSAM subpool

**POOLID**

Local shared resource pool ID

**TYPE** The subpool type: I (Index) or D (Data)

**RRBA** Number of retrieval requests by RBA

**RKEY** Number of retrieval requests by key

**BFALT**

Number of logical records altered

**NREC** Number of new VSAM logical records created

**SYNPTS**

Number of system checkpoint requests

**NMBUFS**

Number of buffers in this VSAM subpool

**VRDS** Number of VSAM control interval reads (OVFLO indicates that the total overflowed a 4-byte field; that is, the total exceeded 4,294,967,295)

**FOUND**

Number of control intervals VSAM found in the subpool through lookaside (OVFLO indicates that the total overflowed a 4-byte field; that is, the total exceeded 4,294,967,295)

**VWTS**

Number of VSAM control interval writes

**ERRORS**

Total number of permanent errors now in the pool and total number of permanent errors in this run

The BSIZE value in the final display is ALL.

### *Enhanced OSAM buffer subpool*

**This topic contains Product-sensitive Programming Interface information.**

Each OSAM subpool is displayed separately. The final display provides the statistical summation for all the OSAM subpools.

**ID** The 4-character POOLID provided at subpool definition time

**BSIZE** The size of the buffers in this OSAM subpool

**NBUF** Number of buffers for this subpool

**FX=** The fix options for this subpool. Y/N indicates whether the DATA BUFFER PREFIX/DATA BUFFERS are fixed.

**LCTREQ**  
Number of LOCATE-type calls for this subpool

**ALTREQ**  
Number of buffer alter calls for this subpool. This count includes NEW BLOCK and BYTALT calls.

**PURGRQ**  
Number of PURGE calls for this subpool

**FNDIPL**  
Number of LOCATE-type calls, for this subpool, where data is already in the OSAM pool

**BFSRCH**  
Number of buffers searched by all LOCATE-type calls for this subpool

**RDREQ**  
Number of READ I/O requests for this subpool

**BFSTLW**  
Number of single block writes initiated by buffer steal routine for this subpool

**PURGWR**  
Number of buffers written by purge

**WBSYID**  
Number of LOCATE calls, for this subpool, that waited because of busy ID

**WBSYWR**  
Number of LOCATE-type calls, for this subpool, that waited because of busy writing

**WBSYRD**  
Number of LOCATE-type calls, for this subpool, that waited due to busy buffer reading

**WRLSEO**  
Number of buffer steal or purge requests, for this subpool, that waited for ownership to be released.

**WNOBFR**  
Number of buffer steal requests, for this subpool, that waited because no buffers were available to be stolen

**ERRORS**  
Total number of I/O errors for this subpool, or the number of the buffers locked in the pool because of write errors for this subpool

**CFREAD**

Number of blocks read from CF.

**CFEXPC**

Number of blocks expected but not read.

**CFWRPR**

Number of blocks written to CF (prime).

**CFWRCH**

Number of blocks written to CF (changed).

**STGCLF**

Number of blocks not written (STG CLS full).

**XVIINV**

Number of XI buffer-invalidate calls.

**XICLCT**

Number of buffers found invalidated by XI on VECTOR call.

**SBSEQR**

Number of immediate (SYNC) sequential reads (SB stat).

**SBANTR**

Number of anticipatory reads (SB stat).

For the summary totals, the POOLID is not shown. BSIZE is set to ALL. NBUF indicates the total number of buffers in the pool. The FX= field is replaced by OSM=, which shows the total size of the OSAM buffer pool.

*Enhanced VSAM buffer subpool***This topic contains Product-sensitive Programming Interface information.**

Each VSAM subpool is displayed separately. The final display provides the statistical summation for all the VSAM subpools.

- ID** The 4-character POOLID provided at subpool definition time
- BSIZE** The size of the buffers in this VSAM subpool
- TYPE** identifies the subpool as containing INDEX or DATA buffer
- FX=** Number of options for this subpool. Y/N is used to indicate whether the INDEX BUFFERS/DATA BUFFER PREFIX/ DATA BUFFERS are fixed.
- RRBA** Number of retrieval requests by RBA
- RKEY** Number of retrieval requests by KEY
- BFALT**  
Number of logical records altered
- NREC** Number of new VSAM logical records created
- SYNC PT**  
Number of system checkpoint (synchronization point) requests
- NBUFS**  
Number of buffers in this VSAM subpool
- VRDS** Number of VSAM control interval reads (OVERFLOW indicates that the total overflowed a 4-byte field; that is, the total exceeded 4,294,967,295)

**FOUND**

Number of control intervals VSAM found in the subpool through lookaside (OVERFLOW indicates that the total overflowed a 4-byte field; that is, the total exceeded 4,294,967,295)

**VWTS**

Number of VSAM control interval writes

**HSR-S**

Number of successful VSAM reads from Hiperspace™ buffers

**HSW-S**

Number of successful VSAM writes to Hiperspace buffers

**HS NBUFS**

Number of Hiperspace buffers defined for this subpool

**HS R/W-FAIL**

Number of failed VSAM reads/writes from or to Hiperspace buffers. This indicates the number of times a VSAM READ/WRITE request from or to Hiperspace resulted in DASD I/O.

**ERRORS**

Number of permanent write errors now in the subpool, or the largest number of errors in this execution

The BSIZE value in the final display is the total size of all the VSAM subpool buffers. BSIZE is set to ALL. For the summary totals, the POOLID is not shown. The TYPE and FX= fields are replaced by VS=, which is the total size of the VSAM subpool in virtual storage, and HS=, which is the total size of the VSAM subpool in Hiperspace.

***Fast Path database buffer pool*****AVAIL**

Number of available page-fixed database buffers

**WRITING**

Number of page-fixed database buffers being written to disk, including the buffers being filled with sequential dependent segments

**PGMUSE**

Number of page-fixed buffers allocated to PSTs

**UNFIXED**

Number of available buffers for starting a new region

**POOLNAME**

Name of the private buffer pool for the area. This name is specified in the VSPEC member.

**CISIZE**

Control interval size of the private pool for the area.

**PBUF** Size allocated for the primary pool (specified in the VSPEC member).

**SBUF** Size allocated for the secondary pool (specified in the VSPEC member).

**MAX** Maximum number of buffers for the private pool (specified in the VSPEC member).

**CURRENT**

Total number of buffers currently in the pool, including both primary and secondary allocations.

**LK** Y indicates that buffer lookaside is active for the pool; N indicates that it is not.

**HITS** Percentage of searches of the pool for which a buffer was found. Only displayed for lookaside pools.

**VALID**

Percentage of times a buffer found in the pool had valid data. An Invalid buffer is read from DASD and the pool copy is replaced. Only displayed for lookaside pools.

You use the percentages displayed for HITS and VALID together.

**Example:** If HITS is 40%, and VALID is 75%, a buffer was found in the pool 40% of the time, and of that 40%, 75% of the buffers found had valid data, that is, 30% of the HITS had valid data. So, IMS had to read data from DASD approximately 70% of the time.

***PSBP, DMBP, PSBW, DBWP, EPCB, and MAIN buffer pools***

**SIZE** pool size

**FREE** amount of currently free space

**HIGH** Highest amount of space ever used in this run

***AOIP, CESS, CIOP, CMDP, DYNP, EMHB, FPWP, HIOP, LUMC, and LUMP buffer pools***

During the execution of IMS, the AOIP, CESS, CIOP, CMDP, DYNP, EMHB, FPWP, HIOP, LUMC, and LUMP buffer pools can dynamically expand and contract.

Internally, each dynamic pool has one or more blocks of storage that are not contiguous, each divided into fixed-length buffers. By obtaining new blocks and releasing unused blocks, a buffer pool can expand or contract during the execution of IMS. The number of blocks needed for each pool depends on several factors such as the number of buffer requests from the pool, buffer size, and block size.

**SIZE** Pool size

**HIGH** Maximum size since last checkpoint

**LIMIT**

Upper expansion limit

**OVERFLOW**

Size of the oversized chain

**Examples**

The following are examples of the /DISPLAY POOL command.

For a list of all of the IMS CBT pools with a brief description of each, see IMS Control Block Table (CBT) Pools (Diagnosis).

***Example 1 for /DISPLAY POOL command***

Entry ET:

/DISPLAY POOL ALL

Response ET:

NAME	SP#	CURR	MAX	GETS	FREES
IOSB	228	28K	28K	3	0
GIOB	228	0K	0K	0	0
OSWA	228	8K	8K	1	0
GOWA	228	0K	0K	0	0
PST	231	52K	56K	14	1
DPST	231	40K	44K	11	1
SAP	231	132K	132K	32	0
GQMW	231	16K	16K	4	0
LQMW	0	324K	324K	81	0
D1WA	228	8K	8K	2	0
DL2W	0	8K	8K	2	0
DG2W	231	4K	4K	1	0
QSAV	231	4K	16K	3	2
VRPL	231	16K	16K	4	0
LSAV	0	808K	808K	201	1
AWE	231	112K	112K	28	0
PDIR	231	36K	36K	1	0
DDIR	231	28K	28K	1	0
LCRE	241	4K	4K	1	0
PCIB	0	0K	0K	0	0
SIDX	241	4K	4K	1	0
RRE	241	4K	4K	1	0
SMB	231	62K	62K	1	0
BCPT	231	4K	4K	1	0
GESE	241	0K	0K	0	0
EPST	231	208K	208K	20	0
IDT	241	0K	0K	0	0
DBRC	231	0K	0K	0	0
TTAB	231	224K	224K	56	0
EZS	241	0K	0K	0	0
FNCB	231	0K	0K	0	0
TCBT	231	4K	4K	1	0
CMWU	231	12K	12K	3	0
FSRB	228	4K	4K	1	0
KLSD	0	0K	0K	0	0
IRLM	231	0K	0K	0	0
STTR	231	52K	52K	13	0
BQEL	231	0K	0K	0	0
USMU	229	0K	0K	0	0
RCTE	231	3K	3K	1	0
SLOG	231	4K	4K	1	0
ADSC	228	0K	0K	0	0
FPCP	231	0K	0K	0	0
AESL	231	0K	0K	0	0
XPST	231	16K	16K	4	0
BXQE	231	16K	16K	4	0
SRBC	228	8K	8K	2	0
MSGP	241	4K	4K	1	0
LPST	251	8K	8K	2	0
CULE	231	4K	4K	1	0
RPST	241	4K	4K	1	0
VTCB	251	18K	18K	0	0
RECA	0	19K	19K	1	0
GSAV	231	0K	0K	0	0
FEIB	231	0K	0K	0	0
IEQE	0	0K	0K	0	0
L56X	231	4K	4K	1	0
X124	241	4K	4K	1	0
STAT	241	4K	4K	1	0
EQEL	241	0K	0K	0	0
TT24	231	24K	24K	6	0
LGWA	231	104K	104K	26	0
VWA	231	8K	8K	2	0
LGWX	0	0K	0K	0	0
LQB	251	104K	104K	0	0
RCNT	251	2K	2K	0	0

```

DDRE 0      0K      0K      0      0
CCB  0      0K      0K      0      0
LGND 251    8K      8K      2      0
USRD 251    4K      4K      1      0
LS24 0      2K      2K      1      0
GS24 231    0K      0K      0      0
CLLE 231    64K     64K     16     1
QMBA 0      8K      8K      1      0
AHDR 251    0K      0K      0      0
XMCI 241    180K    180K    45     1
DBPB 231    0K      0K      0      0
LG24 0      12K     12K     3      0
IAFP 231    0K      0K      0      0
RACW 231    0K      0K      0      0
LUB  251    0K      0K      0      0
TIB  251    28K     28K     1      0
DESC 251    0K      0K      0      0
PF62 251    0K      0K      0      0
QAB  251    0K      0K      0      0
LCLL 0      436K    436K    109    0
CBLK 251    0K      0K      0      0
SVPG 231    24K     24K     6      0
SVPL 0      148K    148K    37     0
SOPB 0      0K      0K      0      0
CBT POOLS      GLBL 1541K LCL 2236K
MESSAGE QUEUE POOL: BFRS/SIZE 20/2112
ENQ  6 DEQ  6 CAN  30 WAIT  0 I/O  15 ERR  0
QBLKS: MAX # RECORDS AVAIL 1332 CUR IN USE 3 = 0 %
MSGQ:  MAX # RECORDS AVAIL 5940 CUR IN USE 4 = 0 %
LMSGQ: MAX # RECORDS AVAIL 1350 CUR IN USE 4 = 0 %
MESSAGE FORMAT POOL: SIZE 49152 SPACE 48576 DIRS 9011
REQ1  2 I/O  2 DIR  4
WAIT  2 FREE 48432 ERR 0
SEQUENTIAL BUFFERING: STATUS = NOT INIT
MAX  N.A. FREE N.A. CURR 0K HIGH 0K
OSAM DB BUFFER POOL: ID 004K BSIZE 4K NBUF 1000 FX=Y/Y
LCTREQ 1765296 NEWBLK 0 ALTREQ 340800
PURGRQ 39371 FNDIPL 1370897 BFSRCH 1987604
RDREQ 378355 BFSTLW 0 PURGWR 150284
WBSYID 1431 WBSYWR 0 WBSYRD 0
WRLSEO 296 WNOBFR 0 ERRORS 00000/00000
OSAM DB BUFFER POOL: ID 008K BSIZE 8K NBUF 100 FX=Y/Y
LCTREQ 228080 NEWBLK 0 ALTREQ 0
PURGRQ 0 FNDIPL 204190 BFSRCH 316566
RDREQ 23891 BFSTLW 0 PURGWR 0
WBSYID 0 WBSYWR 0 WBSYRD 0
WRLSEO 0 WNOBFR 0 ERRORS 00000/00000
OSAM DB BUFFER POOL: ID 012K BSIZE 12K NBUF 100 FX=Y/Y
LCTREQ 83282 NEWBLK 0 ALTREQ 6896
PURGRQ 4384 FNDIPL 70743 BFSRCH 81395
RDREQ 7622 BFSTLW 0 PURGWR 6118
WBSYID 0 WBSYWR 0 WBSYRD 0
WRLSEO 0 WNOBFR 0 ERRORS 00000/00000
OSAM DB BUFFER POOL: BSIZE ALL NBUF 1200 OSM= 6000K
LCTREQ 2076667 NEWBLK 0 ALTREQ 347697
PURGRQ 43755 FNDIPL 1645837 BFSRCH 2385576
RDREQ 409869 BFSTLW 0 PURGWR 156402
WBSYID 1431 WBSYWR 0 WBSYRD 0
WRLSEO 296 WNOBFR 0 ERRORS 00000/00000
VSAM DB BUFFER POOL: ID VLP1 BSIZE 2K TYPE D FX=N/Y/N
RRBA 0 RKEY 0 BFALT 0
NREC 0 SYNC PT 29446 NBUFS 500
VRDS 1253 FOUND 0 VWTS 68
HSR-S 0 HSW-S 0 HS NBUFS 0
HS R/W-FAIL 000000/000000 ERRORS 000000/000000
VSAM DB BUFFER POOL: ID VLP1 BSIZE 4K TYPE D FX=N/Y/N
RRBA 370 RKEY 187583 BFALT 0

```

```

NREC      10750   SYNC PT   29446   NBUFS      1000
VRDS     145632   FOUND      0     VWTS      9771
HSR-S      0     HSW-S      50     HS NBUFS   50
HS R/W-FAIL 000000/000000  ERRORS 000000/000000
VSAM DB BUFFER POOL: ID VPL1 BSIZE 8K TYPE D FX=N/Y/N
RRBA      7375   RKEY      6704   BFALT      0
NREC      0     SYNC PT   29446   NBUFS      100
VRDS      7362   FOUND      0     VWTS      0
HSR-S      0     HSW-S      0     HS NBUFS   0
HS R/W-FAIL 000000/000000  ERRORS 000000/000000
VSAM DB BUFFER POOL: BSIZE ALL VS= 5800K HS= 200K
RRBA      7745   RKEY     194287   BFALT      0
NREC     10752   SYNC PT   29446   NBUFS     1600
VRDS    154247   FOUND      0     VWTS     9839
HSR-S      0     HSW-S      50     HS NBUFS   50
HS R/W-FAIL 000000/000000  ERRORS 000000/000000
DMPB BUFFER POOL:
SIZE 200K FREE 200K HIGH 0K
PSBP BUFFER POOL:
SIZE 80K FREE 80K HIGH 0K
DPSB BUFFER POOL:
SIZE 160K FREE 160K HIGH 0K
CIOP BUFFER POOL:
SIZE 14K HIGH 65K LIMIT NONE OVERFLOW 0K
MAIN BUFFER POOL:
SIZE 40K FREE 39K HIGH 1K
SPAP BUFFER POOL:
SIZE 0K HIGH 0K LIMIT 9999K OVERFLOW 0K
PSBW BUFFER POOL:
SIZE 600K FREE 600K HIGH 0K
DBWP BUFFER POOL:
SIZE 80K FREE 80K HIGH 0K
HIOP BUFFER POOL:
SIZE 214K HIGH 420K LIMIT 9999K OVERFLOW 0K
FPDB BUFFER POOL:
AVAIL = 60 WRITING = 0 PGMUSE = 0 UNFIXED = 90
POOLNAME CFSIZE PBUF SBUF MAX CURRENT LK HITS VALID
1024 01024 00064 00016 00256 00064 N NA NA
2048 02048 00064 00016 00256 00064 N NA NA
CESS BUFFER POOL:
POOL IS NOT AVAILABLE FOR STATISTICS
EMHB BUFFER POOL:
SIZE 0K HIGH 0K LIMIT NONE OVERFLOW 0K
FPWP BUFFER POOL:
SIZE 0K HIGH 0K LIMIT NONE OVERFLOW 0K
EPCB BUFFER POOL:
SIZE 300K FREE 300K HIGH 0K
LUMP BUFFER POOL:
SIZE 68K HIGH 68K LIMIT NONE OVERFLOW 0K
LUMC BUFFER POOL:
SIZE 0K HIGH 32K LIMIT NONE OVERFLOW 0K
AOIP BUFFER POOL:
SIZE 82K HIGH 82K LIMIT NONE OVERFLOW 0K
CMDP BUFFER POOL:
SIZE 0K HIGH 0K LIMIT NONE OVERFLOW 0K
DYNP BUFFER POOL:
SIZE 42K HIGH 42K LIMIT NONE OVERFLOW 0K
*92120/134306*

```

*Example 2 for /DISPLAY POOL command*

Entry ET:

```
/DISPLAY POOL AOIP
```

Response ET:

A0IP BUFFER POOL:  
 SIZE 32K HIGH 32K LIMIT NONE OVERFLOW 0K  
 \*92280/095545\*

*Example 3 for /DISPLAY POOL command*

For a list of all of the IMS CBT pools with a brief description of each, see IMS Control Block Table (CBT) Pools (Diagnosis).

Entry ET:

/DISPLAY POOL CBT

Response ET:

NAME	SP#	CURR	MAX	GETS	FREES
IOSB	228	20K	20K	1	0
GIOB	228	0K	0K	0	0
OSWA	228	8K	8K	1	0
GOWA	228	0K	0K	0	0
PST	231	52K	56K	14	1
DPST	231	40K	44K	11	1
SAP	231	28K	28K	6	0
GQMW	231	12K	12K	3	0
LQMW	0	40K	40K	10	0
D1WA	228	8K	8K	2	0
DL2W	0	8K	8K	2	0
DG2W	231	4K	4K	1	0
QSAV	231	4K	12K	2	1
VRPL	231	16K	16K	4	0
LSAV	0	68K	72K	17	1
AWE	231	12K	12K	3	0
PDIR	231	36K	36K	1	0
DDIR	231	28K	28K	1	0
LCRE	241	4K	4K	1	0
PCIB	0	0K	0K	0	0
SIDX	241	4K	4K	1	0
RRE	241	4K	4K	1	0
SMB	231	61K	61K	1	0
BCPT	231	4K	4K	1	0
GESE	241	0K	0K	0	0
EPST	231	144K	144K	20	0
IDT	241	0K	0K	0	0
DBRC	231	0K	0K	0	0
TTAB	231	184K	184K	46	0
EZS	241	0K	0K	0	0
FNCB	231	0K	0K	0	0
TCBT	231	4K	4K	1	0
CMWU	231	8K	8K	2	0
FSRB	228	4K	4K	1	0
KLSD	0	0K	0K	0	0
IRLM	231	0K	0K	0	0
STTR	231	52K	52K	13	0
BQEL	231	0K	0K	0	0
USMU	229	0K	0K	0	0
RCTE	231	3K	3K	1	0
SLOG	231	4K	4K	1	0
ADSC	228	0K	0K	0	0
FPCP	231	0K	0K	0	0
AESL	231	0K	0K	0	0
XPST	231	16K	16K	4	0
BXQE	231	16K	16K	4	0
SRBC	228	8K	8K	2	0
MSGP	241	4K	4K	1	0
LPST	251	8K	8K	2	0
CULE	231	152K	152K	1	0
RPST	241	4K	4K	1	0

VTCB	251	305K	305K	0	0
RECA	0	19K	19K	1	0
GSAV	231	0K	0K	0	0
FEIB	231	0K	0K	0	0
IEQE	0	0K	0K	0	0
L56X	231	4K	4K	1	0
X124	241	4K	4K	1	0
STAT	241	4K	4K	1	0
EQEL	241	0K	0K	0	0
TT24	231	24K	24K	6	0
LGWA	231	104K	104K	26	0
VWA	231	8K	8K	2	0
LGWX	0	0K	0K	0	0
LQB	251	104K	104K	0	0
RCNT	251	4K	4K	0	0
DDRE	0	0K	0K	0	0
CCB	0	0K	0K	0	0
LGND	251	8K	8K	2	0
USRD	251	4K	4K	1	0
LS24	0	2K	2K	1	0
GS24	231	0K	0K	0	0
CLLE	231	60K	64K	16	1
QMBA	0	8K	8K	1	0
AHDR	251	0K	0K	0	0
XMCI	241	32K	36K	9	1
DBPB	231	0K	0K	0	0
LG24	0	12K	12K	3	0
IAFP	231	0K	0K	0	0
RACW	231	0K	0K	0	0
LUB	251	0K	0K	0	0
TIB	251	28K	28K	1	0
DESC	251	4K	4K	1	0
PF62	251	0K	0K	0	0
QAB	251	0K	0K	0	0
LCLL	0	36K	36K	9	0
CBLK	251	0K	0K	0	0
CBT POOLS		GLBL	1187K	LCL	658K

\*91057/132506\*

*Example 4 for /DISPLAY POOL command*

Entry ET:

/DISPLAY POOL CIOP SPAP HIOP CESS EMHB FPWP LUMP LUMC

Response ET:

```

CIOP BUFFER POOL:
  SIZE      16K HIGH      96K LIMIT      999K OVERFLOW      0K
SPAP BUFFER POOL:
  SIZE       0K HIGH       0K LIMIT     9999K OVERFLOW      0K
HIOP BUFFER POOL:
  SIZE     128K HIGH     128K LIMIT     9999K OVERFLOW      0K
CESS BUFFER POOL:
  POOL IS NOT AVAILABLE FOR STATISTICS
EMHB BUFFER POOL:
  SIZE       0K HIGH       0K LIMIT     9999K OVERFLOW      0K
FPWP BUFFER POOL:
  SIZE       0K HIGH       0K LIMIT     9999K OVERFLOW      0K
LUMP BUFFER POOL:
  SIZE      32K HIGH      32K LIMIT      NONE OVERFLOW      0K
LUMC BUFFER POOL:
  SIZE       0K HIGH       32K LIMIT      NONE OVERFLOW      0K
*91051/161116*

```

*Example 5 for /DISPLAY POOL command*

Entry ET:  
/DISPLAY POOL DBAS

Response ET:

```

SEQUENTIAL BUFFERING:  STATUS = NOT INIT
MAX  N.A.  FREE  N.A.  CURR  0K  HIGH  0K
OSAM DB BUFFER POOL:  ID  BSIZE  2K  NBUF  4  FX=N/N
LCTREQ  0  NEWBLK  0  ALTREQ  0
PURGRQ  0  FNDIPL  0  BFSRCH  0
RDREQ  0  BFSTLW  0  PURGWR  0
WBSYID  0  WBSYWR  0  WBSYRD  0
WRLSEO  0  WNOBFR  0  ERRORS  00000/00000
OSAM DB BUFFER POOL:  ID  BSIZE  6K  NBUF  4  FX=N/N
LCTREQ  0  NEWBLK  0  ALTREQ  0
PURGRQ  0  FNDIPL  0  BFSRCH  0
RDREQ  0  BFSTLW  0  PURGWR  0
WBSYID  0  WBSYWR  0  WBSYRD  0
WRLSEO  0  WNOBFR  0  ERRORS  00000/00000
OSAM DB BUFFER POOL:  ID  BSIZE  8K  NBUF  4  FX=N/N
LCTREQ  0  NEWBLK  0  ALTREQ  0
PURGRQ  0  FNDIPL  0  BFSRCH  0
RDREQ  0  BFSTLW  0  PURGWR  0
WBSYID  0  WBSYWR  0  WBSYRD  0
WRLSEO  0  WNOBFR  0  ERRORS  00000/00000
OSAM DB BUFFER POOL:  BSIZE  ALL  NBUF  12  OSM=  64K
LCTREQ  0  NEWBLK  0  ALTREQ  0
PURGRQ  0  FNDIPL  0  BFSRCH  0
RDREQ  0  BFSTLW  0  PURGWR  0
WBSYID  0  WBSYWR  0  WBSYRD  0
WRLSEO  0  WNOBFR  0  ERRORS  00000/00000
VSAM DB BUFFER POOL:  ID  XXXX  BSIZE  1K  TYPE  D  FX=N/N/N
RRBA  0  RKEY  0  BFALT  0
NREC  0  SYNC  PT  0  NBUFS  8
VRDS  0  FOUND  0  VWTS  0
HSR-S  0  HSW-S  0  HS  NBUFS  0
HS R/W-FAIL  000000/000000  ERRORS  000000/000000
VSAM DB BUFFER POOL:  ID  XXXX  BSIZE  2K  TYPE  D  FX=N/N/N
RRBA  0  RKEY  0  BFALT  0
NREC  0  SYNC  PT  0  NBUFS  4
VRDS  0  FOUND  0  VWTS  0
HSR-S  0  HSW-S  0  HS  NBUFS  0
HS R/W-FAIL  000000/000000  ERRORS  000000/000000
VSAM DB BUFFER POOL:  ID  XXXX  BSIZE  8K  TYPE  D  FX=N/N/N
RRBA  0  RKEY  0  BFALT  0
NREC  0  SYNC  PT  0  NBUFS  4
VRDS  0  FOUND  0  VWTS  0
HSR-S  0  HSW-S  0  HS  NBUFS  0
HS R/W-FAIL  000000/000000  ERRORS  000000/000000
VSAM DB BUFFER POOL:  BSIZE  ALL  VS=  48K  HS=  0K
RRBA  0  RKEY  0  BFALT  0
NREC  0  SYNC  PT  0  NBUFS  16
VRDS  0  FOUND  0  VWTS  0
HSR-S  0  HSW-S  0  HS  NBUFS  0
HS R/W-FAIL  000000/000000  ERRORS  000000/000000
*92120/134346*

```

*Example 6 for /DISPLAY POOL command*

Entry ET:  
/DISPLAY POOL DBB

Response ET:

NAME	SP#	CURR	MAX	GETS	FREES
PDIR	231	36K	36K	1	0
DDIR	231	28K	28K	1	0
SMB	231	61K	61K	1	0
BQEL	231	0K	0K	0	0
IEQE	0	0K	0K	0	0
EQEL	241	0K	0K	0	0
DDRE	0	0K	0K	0	0
CLASS =	DBB	GLBL	125K	LCL	0K
CBT POOLS		GLBL	1187K	LCL	662K

\*91051/161027\*

*Example 7 for /DISPLAY POOL command*

Entry ET:

/DISPLAY POOL DCC

Response ET:

NAME	SP#	CURR	MAX	GETS	FREES
PCIB	0	0K	0K	0	0
VTCB	251	305K	305K	0	0
RECA	0	19K	19K	1	0
LGND	251	16K	16K	4	0
AHDR	251	0K	0K	0	0
IAFP	231	0K	0K	0	0
RACW	231	0K	0K	0	0
LUB	251	0K	0K	0	0
TIB	251	28K	28K	1	0
DESC	251	0K	0K	0	0
PF62	251	0K	0K	0	0
QAB	251	0K	0K	0	0
CBLK	251	0K	0K	0	0
CLASS =	DCC	GLBL	0K	LCL	368K
CBT POOLS		GLBL	1187K	LCL	662K

\*91051/161038\*

*Example 8 for /DISPLAY POOL command*

Entry ET:

/DISPLAY POOL FPDB

Response ET:

POOLNAME	CISIZE	PBUF	SBUF	MAX	CURRENT	LK	HITS	VALID
V\$\$\$CFSTRUCTURE1	04096	00100	00010	00500	00100	N	000%	NA
POOL4	04096	00010	00010	00500	00010	Y	NA	NA

*Example 9 for /DISPLAY POOL command*

Entry ET:

/DISPLAY POOL MFP

Response ET:

MESSAGE	FORMAT	POOL:	SIZE	49152	SPACE	48576	DIRS	9011
REQ1		2	I/O		2	DIR		4
WAIT		2	FREE	48432	ERR			0

\*91051/172502\*

*Example 10 for /DISPLAY POOL command*

Entry ET:

/DISPLAY POOL MFP CIOP MAIN

Response ET:

```
MESSAGE FORMAT POOL:  SIZE      49152 SPACE      48576 DIRS      9011
REQ1           2 I/O           2 DIR           4
WAIT           2 FREE          48432 ERR           0
CIOP BUFFER POOL:
SIZE          96K HIGH          96K LIMIT      999K OVERFLOW      0K
MAIN BUFFER POOL:
SIZE          40K FREE          40K HIGH          5K
*91051/160913*
```

*Example 11 for /DISPLAY POOL command*

Entry ET:

/DISPLAY POOL MFP STAT

Response ET:

```
MFBP BUFFER POOL:
0170 00000000 00000002 00000002 00000000 00000004 00000000 00000000 00000002
0190 00000000 00000000 00000000 00000002 00000000 00000000 00000000 00000000
01B0 00000000 00000001
*91051/172508*
```

*Example 12 for /DISPLAY POOL command*

Entry ET:

/DISPLAY POOL PSBP

Response ET:

```
PSBP BUFFER POOL:
SIZE      80K FREE      80K HIGH      0K
DPSB BUFFER POOL:
SIZE     160K FREE     160K HIGH      0K
*91051/161048*
```

*Example 13 for /DISPLAY POOL command*

Entry ET:

/DISPLAY POOL QBUF

Response ET:

```
MESSAGE QUEUE POOL:  BFRS/SIZE 20/2112          SYS3
ENQ   18 DEQ   18 CAN    52 WAIT    0 I/O   15 ERR  0
QBLKS: MAX # RECORDS AVAIL      5 CUR IN USE      3 = 60 %
MSGQ:  MAX # RECORDS AVAIL 262167 CUR IN USE      4 =  0 %
LMSGQ: MAX # RECORDS AVAIL 524293 CUR IN USE      2 =  0 %
*91051/172432*
```

Explanation: This example shows the /DISPLAY POOL QBUF command output in a non-shared-queues environment.

*Example 14 for /DISPLAY POOL command*

Entry ET:

/DISPLAY POOL QBUF STAT

Response ET:

```

QBUF BUFFER POOL:
0054 00000000 0000005E 00000126 00000001 00000000 00000000 00000007 00000008
0074 00000008 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0094 00000000 00000000 00000000 00000019 00000323 00000000 00000013 00000013
00B4 00000035 00000003 00000004 00000002
*91051/172440*   SYS3

```

*Example 15 for /DISPLAY POOL command*

Entry ET:

```
/DISPLAY POOL SUM
```

Response ET:

```

CBT POOLS      GLBL  1187K LCL    662K
CLASS = OSAM   GLBL   28K LCL     0K
CLASS = GEN    GLBL  724K LCL    54K
CLASS = DEP    GLBL  216K LCL     0K
CLASS = DISP   GLBL   88K LCL   128K
CLASS = DBB    GLBL  125K LCL     0K
CLASS = DCC    GLBL    0K LCL   368K
CLASS = FP     GLBL    7K LCL     0K
*91051/161003*

```

*Example 16 for /DISPLAY POOL command*

Entry ET:

```
/DISPLAY POOL QBUF
```

Response ET:

```

MESSAGE QUEUE POOL:  BFRS/SIZE 480/2520
ENQ  255 DEQ  2 CAN  82 WAIT  0 I/O  28 ERR  0
QBLKS:  MAX # RECORDS AVAIL  50176 CUR IN USE  3 =  0 %
SMSGQ:  MAX # RECORDS AVAIL  52416 CUR IN USE  0 =  0 %
LMSGQ:  MAX # RECORDS AVAIL  20160 CUR IN USE 254 =  1 %
CUR BUFFER INUSE COUNT  379
BUFFER INUSE HIGH WATER MARK  379 I
*13091/160847*

```

Explanation: This example shows the /DIS POOL QBUF command output in a shared-queues environment.

**Related concepts:**

 [Monitoring VSAM buffers \(Database Administration\)](#)

**Related reference:**

 [QUERY POOL command \(Commands\)](#)

## **/DISPLAY PSB command**

Use the /DISPLAY PSB command to display which transactions the PSBs are processing, any routing code associated with the transactions, the databases being accessed, and the type of access.

Subsections:

- “Environment” on page 683
- “Syntax” on page 683
- “Usage notes” on page 683
- “Equivalent IMS type-2 commands” on page 683

- “Examples”

## Environment

The /DISPLAY PSB command is valid in DB/DC, DBCTL, and DCCTL environments.

## Syntax



## Usage notes

For HSSP PSBs, the letter H is appended to the access intent under the ACCESS heading.

To display the status of PSBs, use the /DISPLAY PGM command.

## Equivalent IMS type-2 commands

The following table shows variations of the /DISPLAY PSB command and the IMS type-2 commands that perform similar functions.

Table 301. Type-2 equivalents for the /DISPLAY PSB command.

Task	/DISPLAY PSB command	Similar IMS type-2 command
Displays transactions, routing codes, and databases associated with a PSB.	/DISPLAY PSB <i>psbname</i>	QUERY PGM NAME( <i>pgmname</i> ) SHOW(DB)
		QUERY PGM NAME( <i>pgmname</i> ) SHOW(RTC)
		QUERY PGM NAME( <i>pgmname</i> ) SHOW(TRAN)

## Examples

Entry ET:

```
/DISPLAY PSB DDLTLM06 DDLTLM07
```

Response ET:

PSB-NAME	TRANCODE	RTCODE	DBD-NAME	ACCESS			
DDLTLM06	TXCDLM06	TXCDLM06	MSDBLM01	R/W			
			MSDBLM02	R/W			
			MSDBLM03	R/W			
			MSDBLM01	R/W			
			MSDBLM04	R/W			
			MSDBLM04	R/W			
			MSDBLM05	R/W			
			MSDBLM05	R/W			
			MSDBLM06	R/W			
			MSDBLM06	R/W			
			MSDBLM06	R/W			
			MSDBLM06	R/O			
			DDLTLM07	TXCDLM07	TXCDLM07	MSDBLM01	R/W
						MSDBLM02	R/W

MSDBLM03	R/W
MSDBLM01	R/O
MSDBLM04	R/W
MSDBLM04	R/W
MSDBLM05	R/W
MSDBLM05	R/W
MSDBLM06	R/O
MSDBLM07	R/W

\*91068/111601\*

**Related reference:**

 [QUERY PGM command \(Commands\)](#)

---

## **/DISPLAY PTERM command**

Use the /DISPLAY PTERM command to display physical terminals with specific attributes. The /DISPLAY PTERM command can be specified without an associated LINE keyword on /DISPLAY when PTERM is used with the special "attribute" parameter.

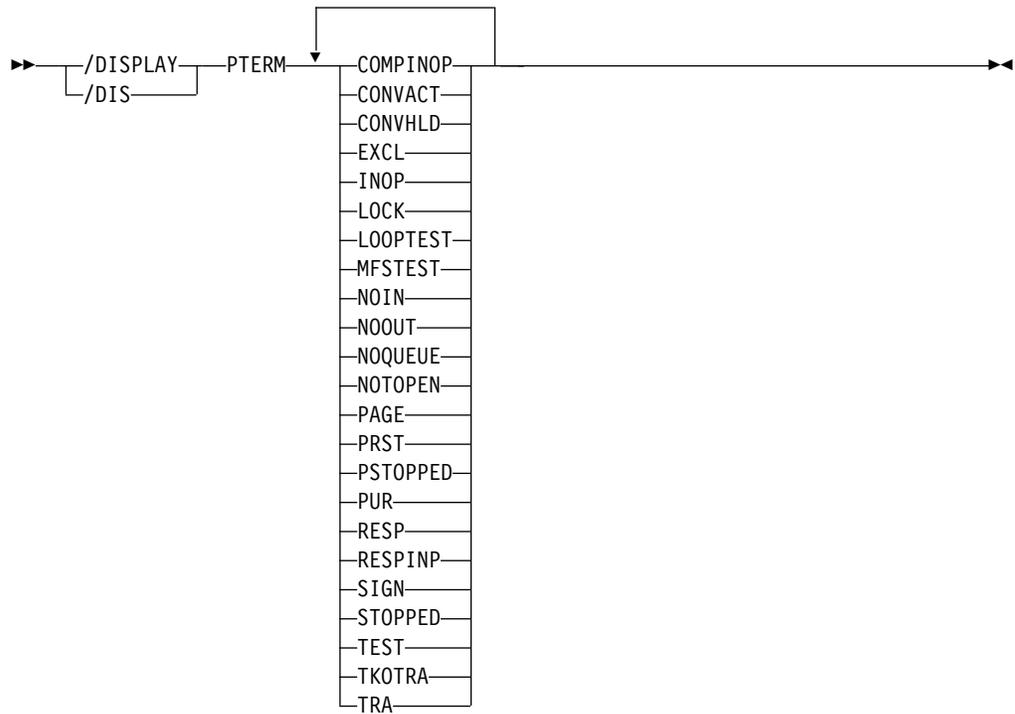
Subsections:

- "Environment"
- "Syntax" on page 685
- "Usage notes" on page 685
- "Examples" on page 686

### **Environment**

The /DISPLAY PTERM command is valid in DB/DC and DCCTL environments.

## Syntax



## Usage notes

Physical terminals are selected for display based on the attribute or attributes specified. The attribute display format is the same as the standard LINE/PTERM display. Attributes usually correspond to the conditions displayed for line/physical terminals.

The attributes that can be specified with the PTERM keyword are listed in the following table. For example, /DISPLAY PTERM TRACE displays all physical terminals that are currently being traced.

*Table 302. /DISPLAY PTERM command attributes*

COMPINOP	CONVACT (Corresponds to the NODE condition CONV-ACT)
CONVHLD (Corresponds to the NODE condition CONV-HLD)	EXCL
INOP	LOCK
LOOPTEST	MFSTEST
NOIN	NOOUT
NOQUEUE	NOTOPEN
PAGE	PRST
PSTOPPED	PUR
RESP	RESPINP
SIGN	STOPPED
TEST	TKOTRA
TRA	

## Examples

This example shows the display of all physical terminals that are being traced.

Entry ET:

```
/DISPLAY PTERM TRA
```

Response ET:

```
LIN/PTE TYPE ADDR RECD ENQCT DEQCT QCT SENT IMS1
2- 1 RDR/PTR **** 0 0 0 0 0 STOPPED INOP NOTOPEN TRA
*13128/111659*
```

---

## /DISPLAY Q command

Use the /DISPLAY Q command to display the message queues according to classes and priority levels.

Subsections:

- "Environment"
- "Syntax"
- "Keywords"
- "Examples" on page 687

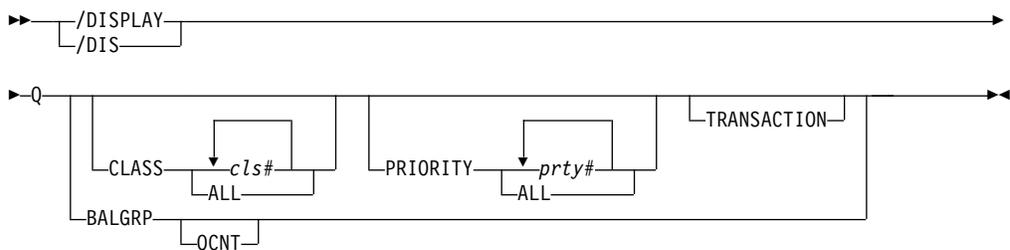
### Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

Table 303. Valid environments for the /DISPLAY Q command and keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
/DISPLAY	X	X	X
BALGRP	X		X
CLASS	X		X
Q	X		X
PRIORITY	X	X	X
TRANSACTION	X		X

### Syntax



### Keywords

The following keywords can be specified for the /DISPLAY Q command:

**BALGRP**

Displays queue counts of the Fast Path load balancing group queues.

**CLASS**

Displays queues for specified classes.

**PRIORITY**

Displays queues for specified priorities.

**QCNT**

Specifies that global queue count information is to be displayed for Fast Path load balancing groups. If QCNT is not specified, only local queue counts are displayed; when it is specified, no local queue counts are displayed.

This keyword is valid only in a shared-queues environment.

**TRANSACTION**

Displays queues for transactions that are ready to run. DFSCPIC is displayed as psbname for CPI Communications driven transactions that have not issued a DL/I call to allocate a PSB.

Combinations of the CLASS, PRIORITY, and TRANSACTION keywords enable classes, priority levels within classes, transactions within priority levels, and message counts to be displayed on an as-needed basis.

**Examples**

The following are examples of the /DISPLAY Q command.

*Example 1 for /DISPLAY Q command*

Entry ET:

```
/DISPLAY Q
```

Response ET:

```
CLS CT  PTY CT  MSG CT  TRAN CT
   3      5     28      8
*90253/103811*
```

Entry ET:

```
/DISPLAY Q BALGRP
```

Response ET:

```
BALGRP  NO.RGNS  MSG CT  ENQ COUNT  DEQ COUNT
DDLTM06      1      1      2      1
DDLTM07      1      0      0      0
*91068/112000*
```

**Explanation:** In a shared-queues environment, the DEQ COUNT column shows the number of messages processed by the local IMS. This can be a message that is entered by the local IMS system or a sharing IMS system in the IMSplex.

Entry ET:

```
/DISPLAY Q BALGRP QCNT
```

Response ET:

```
BALGRP          GBLQCT
SMQFP1          2
SMQFP2          3
*96197/174618*
```

Entry ET:

```
/DISPLAY Q CLASS ALL
```

Response ET:

```
CLS      PTY CT  MSG CT  TRAN CT
  1         1    2      1
  2         1    2      1
  3         1    4      1
*90253/103855*
```

Entry ET:

```
/DISPLAY Q CLASS 1 2
```

Response ET:

```
CLS      PTY CT  MSG CT  TRAN CT
  1         1    2      1
  2         1    2      1
*90253/103913*
```

Entry ET:

```
/DISPLAY Q PRIORITY ALL
```

Response ET:

```
CLS      PTY  MSG CT  TRAN CT
  1         9    2      1
  2         1    2      1
  3         5    4      1
*90253/103945*
```

Entry ET:

```
/DISPLAY Q PRIORITY 10
```

Response ET:

```
CLS      PTY  MSG CT  TRAN CT
  4        10   18      3
*90253/1104016
```

Entry ET:

```
/DISPLAY Q CLASS 4 PRIORITY 10
```

Response ET:

```
CLS      PTY  MSG CT  TRAN CT
  4        10   18      3
*90309/122418*
```

Entry ET:

```
/DISPLAY Q TRANSACTION
```

Response ET:

```
CLS      PTY  MSG CT  TRAN  PSBNAME
  3        10    3  CLOSE  DFSSAM05
  4        10    2  ADDPART DFSSAM04
```

```

4      10      4 PART    DFSSAM02
5      7       1 ADDINV   DFSSAM04
5      10      5 DLETINV  DFSSAM04
*90243/110324*

```

Entry ET:

```
/DISPLAY Q CLASS 3 5 TRANSACTION
```

Response ET:

```

CLS    PTY    MSG CT  TRAN    PSBNAME
3      10     3  CLOSE  DFSSAM05
5      7      1  ADDINV  DFSSAM04
5      10     5  DLETINV DFSSAM04
*90243/110332*

```

Entry ET:

```
/DISPLAY Q CLASS 4 PRIORITY 10 11 TRANSACTION
```

Response ET:

```

CLS    PTY    MSG CT  TRAN    PSBNAME
4      10     2  ADDBPART DFSSAM04
4      10     4  PART    DFSSAM02
*90243/110343*

```

Entry ET:

```
/DISPLAY Q CLASS 1 2 3 PRIORITY 1
```

Response ET:

```

CLS    PTY    MSG CT  TRAN CT
1      1      2      2      1
2      1      2      2      1
3      1      4      4      1
*90253/104321*

```

### *Example 2 for /DISPLAY Q command*

Entry ET:

```
/DISPLAY QUEUE TRANSACTION
```

Response ET:

```

CLS    PTY    MSG CT  TRAN    PSBNAME
20     5      2      2      TPN1    DFSCPIC
20     2      5      5      TPN2    SAAPSB1
22     3      7      7      TPN3    SAAPSB2
*90332/114253*

```

---

## **/DISPLAY QCNT command**

Use the /DISPLAY QCNT command to display global queue information for the specified resource type.

The resource type can be APPC, BALGRP, LTERM, OTMA, REMOTE, or TRANSACTION. This command displays all the queues for the resource type with at least one message whose message age is greater than the message age value specified.

The /DISPLAY QCNT OTMA command displays the global queue counts for all OTMA outbound queues, including those messages that are queued on a

transaction pipe that is defined to a super member. Instead of displaying the queue name that was used to queue the OTMA output messages, the character string 'TMSTMP-' is displayed. If the member is a regular member, the IMS ID of the IMS to which the messages have affinity is displayed following the 'TMSTMP-' character string. If the member is a super member, the IMS ID field is left blank since messages queued to the super member do not have affinity to any IMS.

This count represents the message counts of messages that can be processed by the IMS system where the command is issued. This count includes messages that can be processed by any IMS system (messages with no affinity), plus messages that can be processed by the IMS system where the command is issued (messages with an affinity to the IMS system that issued the command).

Subsections:

- "Environment"
- "Syntax"
- "Keywords" on page 691
- "Usage notes" on page 691
- "Equivalent IMS type-2 commands" on page 691
- "Output fields" on page 692
- "Examples" on page 692

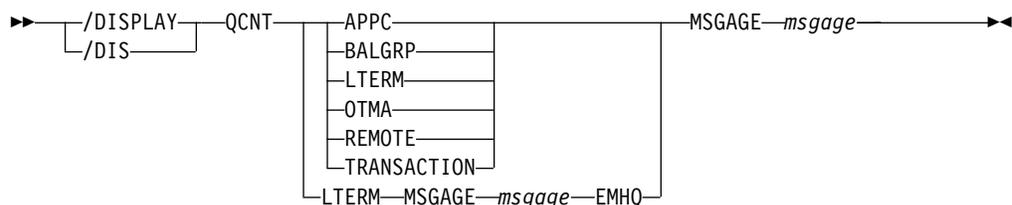
## Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

*Table 304. Valid environments for the /DISPLAY QCNT command and keywords*

Command / Keywords	DB/DC	DBCTL	DCCTL
/DISPLAY	X	X	X
APPC	X		X
BALGRP	X		X
LTERM	X		X
MSGAGE	X		X
OTMA	X		X
QCNT	X		X
REMOTE	X		X
TRANSACTION	X		X

## Syntax



## Keywords

The following list describes the resource types allowed:

**APPC** Specifies that global information for all APPC outbound queues is to be displayed.

For messages that are placed on the shared queues using a side information entry name, the side information entry name is returned in the LUNAME field and the character string, DFSSIDE, is returned in the TPNAME field.

### **BALGRP**

Specifies that global information for LTERMs and MSNAMEs is to be displayed.

### **LTERM**

Specifies that global information for LTERMs and MSNAMEs is to be displayed.

### **MSGAGE**

Specifies the message age in number of days (0-365). Only those queues with messages older than or equal to the value are displayed. The resource name and the time when the message was placed on the shared queues are displayed.

If you specify MSGAGE 0, all resources (queues) are displayed.

### **OTMA**

Specifies that global information for all OTMA outbound queues is to be displayed.

### **REMOTE**

Specifies that global information for remote transactions and remote LTERMs is to be displayed.

### **TRANSACTION**

Specifies that global information for transactions is to be displayed, not including transactions that are suspended.

## Usage notes

This command is valid only in a shared-queues environment.

**Recommendation:** When you issue this command, IMS reads every message for the resource type. To minimize the performance impact, issue this command only when necessary.

## Equivalent IMS type-2 commands

The following table shows variations of the /DISPLAY QCNT command and the IMS type-2 commands that perform similar functions.

Table 305. Type-2 equivalents for the /DISPLAY QCNT command

Task	/DISPLAY QCNT command	Similar IMS type-2 command
Displays lterms with messages that are older than the specified age (shared queues).	/DISPLAY QCNT LTERM MSGAGE <i>x</i>	QUERY LTERM MSGAGE( <i>x</i> )

## Output fields

The output fields are displayed by the /DISPLAY QCNT command:

### QUEUENAME

1-8 byte queue name.

If the output is the result of a /DISPLAY QCNT APPC MSGAGE or a /DISPLAY QCNT OTMA MSGAGE command, the value displayed for the queue name is the character string 'TMSTMP' followed by the IMSID (or the first seven bytes of the RSENAME if XRF capable) of the IMS to which the output messages have affinity.

### QCNT-TOTAL

Total count of messages on the queue.

### QCNT-AGED

Count of messages with a message age greater than or equal to the message age specified. This count does not include messages whose age is less than the message age specified.

### TSTMP-OLD

The time stamp of the oldest message for the queue name on the shared queue.

### TSTMP-NEW

The time stamp of the newest message for the queue name on the shared queue.

**Note:** The time stamp is the time when the message is put on the shared queues by CQS. These time stamps can be used to select records from the CQS log or the IMS log. Because this is a CQS generated time stamp, it might not correspond to the time stamp for the same message on the IMS log as logging a message in IMS and logging the same message in CQS are separate events.

## Examples

The following are examples of the /DISPLAY QCNT command.

### *Example 1 for the /DISPLAY QCNT command*

Entry ET:

```
/DISPLAY QCNT LTERM MSGAGE 5
```

Response ET:

QUEUENAME	QCNT-TOTAL	QCNT-AGED	TSTMP-OLD	TSTMP-NEW
TESTLTEA	1	1	95280/132006	95280/132006
LTERMABC	4	4	95274/083000	95275/091836
LTERMA	100	100	95275/080000	95281/212224
*95290/132006*				

### *Example 2 for the /DISPLAY QCNT command*

Entry ET:

```
/DIS QCNT APPC MSGAGE 0
```

Response ET:

QUEUENAME	QCNT-TOTAL	QCNT-AGED	TSTMP-OLD	TSTMP-NEW
TMSTMP- SYS3	16	16	02192/080833	02192/080928
LUNAME: L62MVS1				
TPNAME: TPNAME123				
TMSTMP- IMS2	12	12	02192/080620	02192/080658
LUNAME: NETWRKID.L62IMS1				
TPNAME: ABCDEFGHIJKLMNOPQRSTUVWXYZ				

\*02192/081543\*

*Example 3 for the IDISPLAY QCNT command*

Entry ET:

/DISPLAY QCNT TRANSACTION MSGAGE 5

Response ET:

QUEUENAME	QCNT-TOTAL	QCNT-AGED	TSTMP-OLD	TSTMP-NEW
TRANA	20	20	95200/132006	95280/132006
TRANBBB	4	4	95274/083000	95275/091836
TRANSACA	220	220	95275/080000	95281/212224
TRANSDDD	13	0	95290/101455	95290/101456
TRANSEEE	55	13	95277/152118	95290/114317

\*95290/132006\*

*Example 4 for the IDISPLAY QCNT command*

Entry ET:

/DIS QCNT OTMA MSGAGE 0

Response ET:

QUEUENAME	QCNT-TOTAL	QCNT-AGED	TSTMP-OLD	TSTMP-NEW
TMSTMP- SYS3	24	24	05038/083745	05038/083745
TMEMBER: HWS003				
TPIPE: TPIPE1				
TMSTMP-	2	2	05038/084022	05038/084022
TMEMBER: SM01				
TPIPE: TPIPE1				
TMSTMP-	5	5	05038/092537	05038/092537
TMEMBER: SM01				
TPIPE: TPIPE2				
TMSTMP-	16	16	05038/072311	05038/072311
TMEMBER: SM01				
TPIPE: TPIPE3				

\*02192/083757\*

Explanation: In this example, 24 messages are queued on TPIPE1 for member HWS003. For super member SM01, two messages are queued on TPIPE1, five messages are queued on TPIPE, and sixteen messages are queued on TPIPE3.

*Example 5 for the IDISPLAY QCNT command*

Entry ET:

/DIS QCNT LTERM MSGAGE 0 EMHQ

Response ET:

QUEUENAME	QCNT-TOTAL	QCNT-AGED	TSTMP-OLD
TSTMP-NEW	SYS3		
IMSUS01	1	1	04309/103912

```

04309/103912   SYS3
IMSUS02                1          1   04309/105127
04309/105127   SYS3
*04310/132127*  SYS3

```

**Related reference:**

 [QUERY LTERM command \(Commands\)](#)

## **/DISPLAY RECOVERY command**

Use the /DISPLAY RECOVERY command to display the recovery process in progress.

It also displays the list of database data sets and areas being recovered by an online database recovery service. The database data sets and areas are displayed as part of a recovery list, which is a set of database data sets, areas, or both processed by an online database recovery service in a single recovery instance. The /DISPLAY RECOVERY command also shows the status of one or all of the recovery lists that exist.

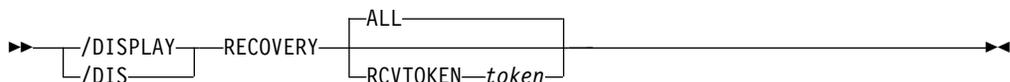
Subsections:

- “Environment”
- “Syntax”
- “Keywords”
- “Usage notes” on page 695
- “Recovery list information” on page 695
- “Recovery progress information” on page 696
- “Recovery list entry information” on page 696
- “Output fields” on page 697
- “Examples” on page 697

### **Environment**

The /DISPLAY RECOVERY command is valid in DB/DC and DBCTL environments.

### **Syntax**



### **Keywords**

The following keywords can be specified for the /DISPLAY RECOVERY command

**ALL**

An optional parameter that displays all the recovery list information and the recovery progress information. No recovery list entry information is displayed.

**RCVTOKEN**

An optional parameter that displays the recovery token associated with the specific recovery list. The recovery list information and recovery progress

information for the specific list display with the recovery list entry information, including the database data set and areas contained in the recovery list.

*token*

Specifies the unique recovery token associated with the recovery list. This token can be up to eight characters in length. If RCVTOKEN is specified, *token* must be supplied.

## Usage notes

If an online database recovery service is executing in an IMS DB/DC control region, the /DISPLAY RECOVERY command can be issued from LTERMs, automated operator (AO) application programs, the IMS Master Terminal (MTO), or the z/OS master console and secondary console.

If an online database recovery service is running in an IMS DBCTL region, the /DISPLAY RECOVERY command can be issued by programs using the IMS DBCTL AOI or the z/OS master console and secondary console.

## Recovery list information

This section displays the status of one or more recovery lists. The status includes the following information:

### TOKEN

This shows the recovery list token. If /DIS RECOVERY ALL is issued and no recovery lists exist, NO LISTS is displayed. If IMS is unable to obtain recovery list information from the recovery facility, UNAVAIL is displayed.

### STATUS

The current status of the recovery list. The status is one of the following:

#### FORMING

A /RECOVER START command has not been issued.

#### STARTED

A /RECOVER START command has been issued. Recovery processing started but has not completed.

#### UNKNOWN

No list with the given token can be found.

#### STOPPING

A /REC STOP ALLENT command was issued.

### ERROR

The choices are ABORT or CONT. These are the action options in case an error is encountered.

### REC TYPE

Indicates the type of recovery being processed. The type is one of the following:

**FULL** A full recovery is processing.

**TSR** A time stamp recovery is processing. This occurs when RCVTIME was specified on the /RECOVER START command, but PITR was not specified.

**PITR** A time-stamp recovery is processing with the point-in-time recovery (PITR) option. This occurs when RCVTIME is specified on the /RECOVER START command with PITR.

N/A Recovery has not been started. The type is undefined.

**PROC** The values that may appear reflect the recovery progress in two phases of processing: reading log data sets and restoring image copy data. The format of the entries are as follows:

- nnnn of mmmm LOGS READ
- nnnn of mmmm RESTORED
- COMPLETE - this state will only appear for a few seconds following the completion of restoration of the last DBDS and termination of the recovery list.
- NOT STARTED - The recovery list is ready for processing or for additional updates.
- routing time - if an online database recovery service is being used.

**IC#** Indicates the primary image copy will be used if an image copy is to be used during recovery.

**SOURCE**

The primary copy as contained in the RECON is used for recovery.

## Recovery progress information

This section displays status of the recovery in progress, if there is one. The section is omitted if no recovery is in progress. The status includes the following information:

**TOKEN**

The recovery list token.

**LAST PROCESSED**

The time stamp of the last log record processed by the recovery. The time is displayed in local time.

**RCVTIME**

The RCVTIME value specified on the /RECOVER START command or N/A if none was specified. The time is displayed in local time.

## Recovery list entry information

This section displays the list of database data sets and areas in the given recovery list. The status includes the following information:

**DATABASE DATA SET**

For full function, this is the DB name and DD name of the database data set. For Fast Path, it is the area name. If there are multiple area data sets for the area, the string '(MADS)' will also be displayed.

**START OPTION**

This shows the start option that was specified on the /RECOVER ADD or /RECOVER START command or taken from the system default. The values can be **STALOCAL**, **STAGLOBAL**, or **OFFLINE**.

**STATUS**

This indicates the status of the database data set or area. The status can be one of the following:

**FAILED**

An error occurred for this database data set or area during recovery.

### STOPPED

A /RECOVER STOP command was entered to stop recovery processing for the database data set or area.

### NORMAL

Recovery is in progress or pending for the database data set or area.

### INVALID

DBRC no longer contains information about this database data set or area.

### AUTH SSID

A list of SSIDs that are still authorized to the database data set or area. **NONE** displays if no IMS is authorized to the database data set or area. If recovery is in progress, **N/A** displays.

## Output fields

The /DISPLAY RECOVERY command output is divided into three sections.

- Recovery List Information
- Recovery Progress Information
- Recovery List Entry Information

## Examples

The following are examples of the /DISPLAY RECOVERY command.

### *Example 1 for /DISPLAY RECOVERY command*

In this example, a /DISPLAY RECOVERY command is issued before a /RECOVER START for RCVTOKEN RECOV1 is issued.

Entry ET:

```
/DISPLAY RECOVERY RCVTOKEN RECOV1
```

Response ET:

```
**** RECOVERY LIST INFORMATION ****
TOKEN  STATUS  ERROR- REC TYPE  PROC  IC#  SOURCE
RECOV1 FORMING N/A    N/A    N/A    N/A
**** RECOVERY LIST ENTRY INFORMATION ****
DATABASE DATA SET  START OPTION  STATUS  AUTH SSID
DB23AR1              STAGLOBAL    NORMAL  NONE
DB23AR2              OFFLINE      NORMAL  NONE
DB23AR3              STAGLOBAL    NORMAL  NONE
DB23AR4              STALOCAL     NORMAL  NONE
DB23AR5              STALOCAL     NORMAL  NONE
DHVNTZ02 HIDAM      STAGLOBAL    NORMAL  NONE
DIVNTZ02 DBHVSAM1  OFFLINE      NORMAL  NONE
DXVNTZ02 XDLBT04I  STALOCAL     NORMAL  NONE
*03127/153515*
```

### *Example 2 for /DISPLAY RECOVERY command*

In Example 2, a /DISPLAY RECOVERY command is issued before a /RECOVER START for RCVTOKEN RECOV1 is issued. Some of the database data sets in the recovery list are still authorized to two IMS systems.

Entry ET:

/DISPLAY RECOVERY RCVTOKEN RECOV1

Response ET:

```
**** RECOVERY LIST INFORMATION ****
TOKEN   STATUS   ERROR- REC TYPE  PROC  IC#  SOURCE
RECOV1  FORMING  N/A    N/A      N/A   N/A
**** RECOVERY LIST ENTRY INFORMATION ****
DATABASE DATA SET  START OPTION  STATUS  AUTH SSID
DB23AR1             STAGLOBAL    NORMAL  NONE
DB23AR2             OFFLINE      NORMAL  NONE
DB23AR3             STAGLOBAL    NORMAL  NONE
DB23AR4             STALOCAL     NORMAL  NONE
DB23AR5             STALOCAL     NORMAL  NONE
DHVNTZ02 HIDAM     STAGLOBAL    NORMAL  IMS1
                               IMS2
DIVNTZ02 DBHVSAM1  OFFLINE      NORMAL  IMS1
                               IMS2
DXVNTZ02 XDLBT04I  STALOCAL     NORMAL  IMS1
                               IMS2      *03127/153515*
```

*Example 3 for /DISPLAY RECOVERY command*

In this example, a /DISPLAY RECOVERY command is issued after a /RECOVER START RCVTOKEN RECOV1 ERRORCONT command was issued when the IMS Database Recovery Facility is the recovery product being used. In this case, all logs have been processed and three of the eight data sets have been restored.

Entry ET:

/DISPLAY RECOVERY RCVTOKEN RECOV1

Response ET:

```
**** RECOVERY LIST INFORMATION ****
TOKEN   STATUS   ERROR- REC TYPE  PROC  IC#  SOURCE
RECOV1  STARTED  CONT  FULL    RTDB  0    PRI
**** RECOVERY PROGRESS INFORMATION ****
TOKEN   PROGRESS INFORMATION  RCVTIME
RECOV1  0003 OF 0008 RESTORED  N/A
**** RECOVERY LIST ENTRY INFORMATION ****
DATABASE DATA SET  START OPTION  STATUS  AUTH SSID
DB23AR1             STAGLOBAL    NORMAL  N/A
DB23AR2             OFFLINE      NORMAL  N/A
DB23AR3             STAGLOBAL    NORMAL  N/A
DB23AR4             STALOCAL     NORMAL  N/A
DB23AR5             STALOCAL     NORMAL  N/A
DHVNTZ02 HIDAM     STAGLOBAL    NORMAL  N/A
DIVNTZ02 DBHVSAM1  OFFLINE      NORMAL  N/A
DXVNTZ02 XDLBT04I  STALOCAL     NORMAL  N/A  *03127/153515*
```

*Example 4 for /DISPLAY RECOVERY command*

In the following example, a /DISPLAY RECOVERY command is issued after a /RECOVER START ERRORCONT command was issued. Also a /RECOVER STOP command was issued for DHVNTZ02, and DB23AR2 failed during a recovery.

Entry ET:

/DISPLAY RECOVERY RCVTOKEN RECOV1

Response ET:

```
**** RECOVERY LIST INFORMATION ****
TOKEN   STATUS   ERROR- REC TYPE  PROC  IC#  SOURCE
RECOV1  STARTED  CONT  FULL    RTDB  0    PRI
```

```

**** RECOVERY PROGRESS INFORMATION *****
TOKEN   PROGRESS INFORMATION   RCVTIME
RECOV1  2000.251 07:57:00.3    N/A
**** RECOVERY LIST ENTRY INFORMATION *****
DATABASE DATA SET  START OPTION  STATUS  AUTH SSID
DB23AR1             STAGLOBAL   NORMAL  N/A
DB23AR2             OFFLINE     FAILED  N/A
DB23AR3             STAGLOBAL   NORMAL  N/A
DB23AR4             STALOCAL   NORMAL  N/A
DB23AR5             STALOCAL   NORMAL  N/A
DHSVNTZ02 HIDAM    STAGLOBAL   STOPPED N/A
DIVVNTZ02 DBHVSAM1 OFFLINE     NORMAL  N/A
DXVNTZ02 XDLBT04I STALOCAL   NORMAL  N/A *03127/153515*

```

**Example 5 for /DISPLAY RECOVERY command**

In this example, a /DISPLAY RECOVERY ALL command is issued after three lists are created using the IMS Database Recovery Facility recovery product. In this case, the IMS Database Recovery Facility has processed 76 of the 126 logs that will need to be read. One of the lists (RECOV2) is undergoing TSR.

Entry ET:

```
/DISPLAY RECOVERY ALL
```

Response ET:

```

**** RECOVERY LIST INFORMATION *****
TOKEN   STATUS  ERROR- REC TYPE  PROC  IC#  SOURCE
RECOV1  FORMING N/A   N/A    N/A   N/A  N/A
RECOV2  STARTED CONT  TSR    RTDB  0    PRI
RECOV3  FORMING N/A   N/A    N/A   N/A  N/A
**** RECOVERY PROGRESS INFORMATION *****
TOKEN   PROGRESS INFORMATION   RCVTIME
RECOV2  0076 OF 0125 LOGS READ  2003.127 08:30:00.0
**** RECOVERY LIST ENTRY INFORMATION *****
DATABASE DATA SET  START OPTION  STATUS  AUTH SSID
DB23AR1             STAGLOBAL   NORMAL  N/A
DB23AR2             OFFLINE     NORMAL  N/A
DB23AR3             STAGLOBAL   NORMAL  N/A
DB23AR4             STALOCAL   NORMAL  N/A
DB23AR5             STALOCAL   NORMAL  N/A
DHSVNTZ02 HIDAM    STAGLOBAL   NORMAL  N/A
DIVVNTZ02 DBHVSAM1 OFFLINE     NORMAL  N/A
DXVNTZ02 XDLBT04I STALOCAL   NORMAL  N/A *03127/153515*

```

**Example 6 for /DISPLAY RECOVERY command**

In this example, a /DISPLAY RECOVERY command is issued after three lists are created. No other parameters are specified.

Entry ET:

```
/DISPLAY RECOVERY ALL
```

Response ET:

```

**** RECOVERY LIST INFORMATION *****
TOKEN   STATUS  ERROR- REC TYPE  PROC  IC#  SOURCE
RECOV1  FORMING N/A   N/A    N/A   N/A  N/A
RECOV2  FORMING N/A   N/A    N/A   N/A  N/A
RECOV3  FORMING N/A   N/A    N/A   N/A  N/A
*03127/153515*

```

**Example 7 for /DISPLAY RECOVERY command**

In this example, a /DISPLAY RECOVERY ALL command is issued when no recovery lists exist.

Entry ET:

```
/DISPLAY RECOVERY ALL
```

Response ET:

```
**** RECOVERY LIST INFORMATION *****
TOKEN  STATUS  ERROR- REC TYPE  PROC  IC#  SOURCE
**** RECOVERY LIST INFORMATION *****
TOKEN  STATUS  ERROR- REC TYPE  PROC  IC#  SOURCE
NO LISTS
*03127/153515*
```

#### *Example 8 for /DISPLAY RECOVERY command*

In this example, a /DISPLAY RECOVERY RCVTOKEN RECOV3 is issued when the RCVTOKEN does not exist.

Entry ET:

```
/DISPLAY RECOVERY RCVTOKEN RECOV3
```

Response ET:

```
**** RECOVERY LIST INFORMATION *****
TOKEN  STATUS  ERROR- REC TYPE  PROC  IC#  SOURCE
**** RECOVERY LIST INFORMATION *****
TOKEN  STATUS  ERROR- REC TYPE  PROC  IC#  SOURCE
RECOV3  UNKNOWN  N/A    N/A    *03127/153515*
```

---

## **/DISPLAY RTCODE command**

Use the /DISPLAY RTCODE command to display the status of Fast Path routing codes, the PSB using the routing code, and the region, if any.

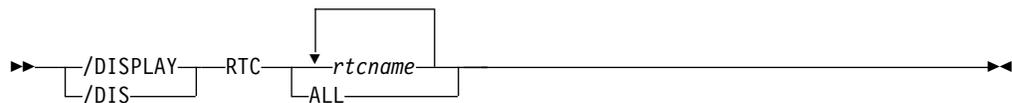
Subsections:

- "Environment"
- "Syntax"
- "Equivalent IMS type-2 commands" on page 701
- "Output fields" on page 701
- "Examples" on page 701

### **Environment**

The /DISPLAY RTCODE command is valid in DB/DC and DCCTL environments.

### **Syntax**



## Equivalent IMS type-2 commands

The following table shows variations of the /DISPLAY RTC command and the IMS type-2 commands that perform similar functions.

Table 306. Type-2 equivalents for the /DISPLAY RTC command

Task	/DISPLAY RTC command	Similar IMS type-2 command
Displays information about one or more Fast Path routing codes.	/DISPLAY RTC <i>rtcname1...rtcnamen</i>   ALL	QUERY RTC NAME( <i>rtcname1...rtcnamen</i>   *) SHOW(ALL)

## Output fields

The status displayed can be one of the following:

NOTINIT  
NOTSCHED  
STOPPED

## Examples

Entry ET:

```
/DISPLAY RTCODE ALL
```

Response ET:

```
RTCODE    PROGRAM    STATUS
DCL       EMHPSB     NOTSCHED,STOPPED
EMHCOBOL  EMHCOBOL  NOTSCHED,STOPPED
EMHPLI    EMHPLI     NOTSCHED,STOPPED
EMHTX     EMHPSB     NOTSCHED,STOPPED
TXCDLM06  DDLTLM06  STOPPED
TXCDLM07  DDLTLM07  STOPPED
*91068/114840*
```

**Related reference:**

 [QUERY RTC command \(Commands\)](#)

---

## /DISPLAY SHUTDOWN STATUS command

Use the /DISPLAY SHUTDOWN STATUS command to display system activity during a shutdown type of checkpoint.

Subsections:

- “Environment”
- “Syntax” on page 702
- “Usage notes” on page 702
- “Output fields” on page 703
- “Examples” on page 704

## Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

Table 307. Valid environments for the /DISPLAY SHUTDOWN command and keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
/DISPLAY	X	X	X
SHUTDOWN	X	X	X
STATUS	X	X	X

## Syntax

→ /DISPLAY SHUTDOWN STATUS →  
 ↳ /DIS

## Usage notes

When issuing the /DISPLAY SHUTDOWN STATUS command from a specific environment (such as DBCTL), only the information that is valid for that environment is displayed. For example, in a DBCTL environment, only the number of IMS threads is displayed.

In an LU 6.2 environment, the /DISPLAY SHUTDOWN STATUS command indicates the dependent regions that have active CPI Communications driven transaction programs. These regions must be terminated before shutdown is completed. In addition, the command also displays LU 6.2 conversations that hang the shutdown processing.

In a shared-queues environment, the /DISPLAY SHUTDOWN STATUS command displays the CQS job name that needs to be restarted if IMS shutdown hangs because CQS is inactive.

If a /DISPLAY SHUTDOWN STATUS command is issued while a database is being recovered with the Online Recovery Services, the response DB RECOVERY IN PROGRESS is returned in addition to the other responses. The /DISPLAY SHUTDOWN STATUS command displays a line with the count of active and indoubt ODBA threads that are active.

One of the following commands can be used to close the VTAM node:

- /CHECKPOINT (except simple checkpoint)
- /CLSDST
- /IDLE NODE (only after a /CLDST command has been tried first)
- /QUIESCE (for session type 6 nodes only)
- /STOP NODE

The /STOP DC command or the /CHECKPOINT (other than simple checkpoint) command can be issued to close the VTAM ACB.

If a checkpoint purge is in progress and messages are in the output queues, an orderly shutdown can be initiated by entering the /CLSDST command or the /STOP command. This allows termination after the next output message is sent, but might be insufficient for display terminals or terminals in input mode. If a node is hung, the following can be done:

1. A /CLSDST FORCE command can be issued.
2. If that fails, then the /IDLE command can be issued.

If any messages are queued for PTERM1 (system console) or the master terminal, they must be requested and received before the purge is completed. The /STOP and /IDLE commands are ineffective against these terminals. When using ISC parallel sessions, one line of information is displayed for each applicable session of the node. The user name is also displayed.

In addition to /DISPLAY SHUTDOWN STATUS command, use the following commands to get more information during shutdown:

- Use /DISPLAY CCTL command for all coordinator controllers.
- Use /DISPLAY UOR command for displaying status information about units of work for protected resources that are managed by z/OS Resource Recovery Services (RRS).

## Output fields

The status displayed when the /DISPLAY SHUTDOWN STATUS command is issued can include the following information:

### COMMIT *x* TMEMBER=*membername* TPIPE=*tpipename*

The commit mode for the transaction and which OTMA client (member) and TPIPE is currently in progress and thus preventing completion of shutdown. COMMIT *x* can be either of the following:

- 0 Commit-Then-Send
- 1 Send-Then-Commit

### Master Terminal Status

The following text is displayed for the primary master, secondary master, or system console:

```
PRIMARY MSTR
SECOND MSTR      ACTIVE , MESSAGES WAITING
SYS CONSOLE
```

ACTIVE indicates the terminal is in the following states:

```
OPERABLE
ALLOWING INPUT
ALLOWING OUTPUT
NOT STOPPED FOR QUEUEING
NOT LOCKED
NOT IN TEST MODE
```

MESSAGES WAITING indicates that there are system messages (Q3) waiting to be sent, thus preventing shutdown from completing.

### MSG-IN *x* MSG-OUT *y*

The message totals, where *x* is the total number of input messages and *y* is the total number of output messages.

### ONLINE CHANGE IN PROGRESS

Online change was initiated with the /MODIFY PREPARE command or the INITIATE OLC PHASE(PREPARE) command.

### OTMA PHASE=*x*

The current phase of IMS Open Transaction Manager Access (OTMA) processing. The shutdown phases are:

- 1 OTMA shutdown processing has begun.

- 2 Dependent regions have terminated. DFS1970 messages have been sent to OTMA clients for all inflight Send-then-Commit (commit mode 1) transactions.
- 3 OTMA is waiting for termination of all client-processing-related activities.
- 4 OTMA shutdown is complete.

### SYSTEM PURGING

The checkpoint is the result of a /CHECKPOINT PURGE command.

### TERMINAL USER STATUS

The header for the portion of the display output that shows active terminals, whose status can be one of the following:

```
INPUT IN PROCESS
OUTPUT IN PROCESS
AWAITING RESPONSE
XX MSGS IN QUEUE
```

If there is no terminal activity, the status can be one of the following:

```
NO INPUTTING LINES
NO OUTPUTTING LINES
```

If the system-defined MSC LNB is active, the following message is displayed:

```
LINK SYS      OUTPUT IN PROCESS
```

### *xx* MSG PROCESSING REGION(S) ACTIVE

Where *xx* is the number of BMP, TP, and FP regions that are active. If the checkpoint has already been posted, this is the only status displayed.

For DBCTL, a count of active CCTL threads is included.

### *xxx* ODBA THREADS, *yyy* INDOUBT ODBA THREADS ACTIVE

Where *xxx* is the number of active ODBA threads and *yyy* is number of indoubt ODBA threads active in the system

### *xxxxxxx* PROCESSING REGION(S) ACTIVE

Active regions, where *xxxxxxx* is one of the following types:

```
BMP MSG
TP MSG
FP
```

If status of a line or link shows input or output in progress and immediate shutdown is desired, the /IDLE command can be used.

## Examples

The following are examples of the /DISPLAY SHUTDOWN STATUS command.

### *Example 1 for /DISPLAY SHUTDOWN STATUS command*

Entry ET:

```
/DISPLAY SHUTDOWN STATUS
```

Response ET:

```
DFS134 SHUTDOWN CHECKPOINT NOT IN PROGRESS,
CANNOT PROCESS COMMAND
```

Entry ET:

/DISPLAY SHUTDOWN STATUS

Response ET:

SYSTEM PURGING

```
TERMINAL USER STATUS
  5- 1      INPUT IN PROCESS
  5- 2      INPUT IN PROCESS
 11- 1      OUTPUT IN PROCESS
TERMINAL USER STATUS
LINK 10     OUTPUT IN PROCESS
  MSG-IN 2  MSG-OUT 2
MASTER ACTIVE
*91111/222226*
```

Entry ET:

/DISPLAY SHUTDOWN STATUS

Response ET:

```
TERMINAL USER STATUS
NSLUTP6 SPOOLA AWAITING RESPONSE
NSLUTP6 SPOOLB 5 MSGS IN QUEUE
NSLUTP6 SPOOLC INPUT IN PROCESS
NSLUTP6 SPOOLD OUTPUT IN PROCESS
*91111/222226*
```

*Example 2 for /DISPLAY SHUTDOWN STATUS command*

Entry ET:

/DISPLAY SHUTDOWN STATUS

Response ET:

```
TERMINAL USER STATUS
  5- 1      INPUT IN PROCESS
  5- 2      INPUT IN PROCESS
 11- 1      OUTPUT IN PROCESS
TERMINAL USER STATUS
LINK 10     OUTPUT IN PROCESS
  MSG-IN 2  MSG-OUT 2
MASTER ACTIVE
CPI TRAN TRNCODE1 ACTIVE IN REGID 1
CPI TRAN TRNCODE2 ACTIVE IN REGID 4
IMSLU=L62IMS #APPC-CONV= 8 ENABLED
LUNAME STATUS
LUNAME1 3 CONVERSATIONS(S) IN PROCESS
LUNAME1 1 OUTPUT IN PROCESS
LUNAME2 4 CONVERSATIONS(S) IN PROCESS
*90332/114253*
```

Entry ET:

/DISPLAY SHUTDOWN STATUS

Response ET:

```
TERMINAL SUBPOOL STATUS
  5- 1      INPUT IN PROCESS
  5- 2      INPUT IN PROCESS
 11- 1      OUTPUT IN PROCESS
TERMINAL SUBPOOL STATUS
LINK 10     OUTPUT IN PROCESS
  MSG-IN 2  MSG-OUT 2
MASTER ACTIVE
```

```

CPI TRAN TRNCODE1 ACTIVE IN REGID1
CPI TRAN TRNCODE2 ACTIVE IN REGID4
IMSLU=SYS1.IMSLUNME      #APPC-CONV=      8  ENABLED
LUNAME                    STATUS
NETWORK1.LUNAME1        3 CONVERSATION(S) IN PROCESS
NETWORK1.LUNAME1        1 OUTPUT IN PROCESS
NID2.LUNAME2            4 CONVERSATION(S) IN PROCESS
*92232/114253*

```

*Example 3 for /DISPLAY SHUTDOWN STATUS command*

Entry ET:

```
/DISPLAY SHUTDOWN STATUS
```

Response ET:

```

TERMINAL USER      STATUS
1- 1              INPUT IN PROCESS
TERMINAL USER      STATUS
NO OUTPUTTING LINES
MSG-IN 1          MSG-OUT 0
MASTER ACTIVE
OTMA PHASE=2
COMMIT 0 TMBER=CLIENT1          TPIPE=TPIPE1
*94298/174604*

```

*Example 4 for /DISPLAY SHUTDOWN STATUS command*

Entry ET:

```
/DISPLAY SHUTDOWN STATUS
```

Response ET:

```

TERMINAL USER      STATUS
1- 1              INPUT IN PROCESS
TERMINAL USER      STATUS
NO OUTPUTTING LINES
MSG-IN 1          MSG-OUT 0
PRIMARY MSTR SECOND MSTR ACTIVE, MESSAGES WAITING
IMSLU=N/A.N/A      #APPC-CONV=      0  DISABLED
OTMA PHASE=0
0 ODBA THREADS,   1 INDOUBT ODBA THREADS ACTIVE
*06031/135629*

```

*Example 5 for /DISPLAY SHUTDOWN STATUS command*

Entry ET:

```
/DISPLAY SHUTDOWN STATUS
```

Response ET:

```

TERMINAL USER      STATUS
1- 1              INPUT IN PROCESS
TERMINAL USER      STATUS
LINK SYS           OUTPUT IN PROCESS
MSG-IN 1          MSG-OUT 1
MASTER ACTIVE
IMSLU=L62IMS      #APPC-CONV=      3  PURGING
LUNAME            STATUS
IMSNET.L62MVS1    3 CONVERSATION(S) IN PROCESS
OTMA PHASE=0
*04292/112920*

```

---

## /DISPLAY STATUS command

Use the /DISPLAY STATUS command to display the output for all the IMS resources.

Subsections:

- “Environment”
- “Syntax”
- “Usage notes” on page 708
- “Equivalent IMS type-2 commands” on page 709
- “Output fields” on page 709
- “Examples” on page 710

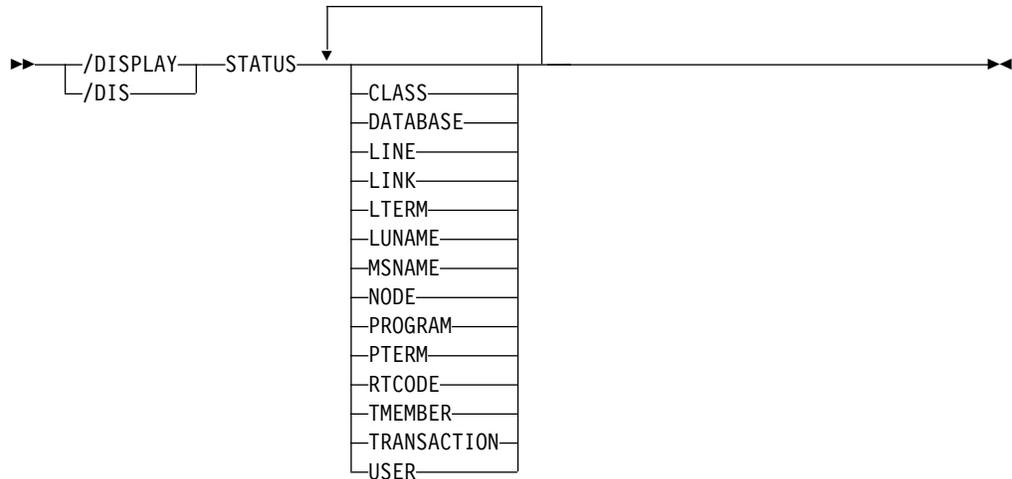
### Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

*Table 308. Valid environments for the /DISPLAY STATUS command and keywords*

Command / Keywords	DB/DC	DBCTL	DCCTL
/DISPLAY	X	X	X
CLASS	X		X
DATABASE	X	X	
LINE	X		X
LINK	X		X
LTERM	X		X
LUNAME	X		X
MSNAME	X		X
NODE	X		X
PROGRAM	X	X	X
PTERM	X		X
RTCODE	X		X
STATUS	X	X	X
TMEMBER	X		X
TRANSACTION	X		X
USER	X		X

### Syntax



## Usage notes

Any combination of the indicated keywords can be used to display the associated resources. If no exceptional conditions are found for any of the resources specified by the individual keywords, STATUS UNRESTRICTED is displayed.

For Fast Path databases, the `/DISPLAY STATUS DATABASE` command shows the area and ADS exceptions as well as the database status. Each area with exceptions is displayed on a separate line after the database line. Each ADS with exceptions is displayed on a separate line after the area line.

The `/DISPLAY STATUS USER` command provides the status of a user. The ALLOC status is followed by the node name to which the user is allocated or signed onto in parenthesis.

**Attention:** In an IMSplex, if NODE, LTERM, or USER is specified with the `/DISPLAY STATUS` command, it will result in extensive accesses to the Resource Manager for global information, and their use should be carefully considered.

In a DBCTL environment, `/DISPLAY STATUS` displays only database and program information.

In a DCCTL environment, `/DISPLAY STATUS` does not display database information.

For `/DISPLAY STATUS TRANSACTION`, DFSCPIC is displayed as the PSB name for CPI Communications driven transactions that have not issued a DL/I APSB call to allocate a PSB.

A status of NOINIT will appear in the output of a `/DISPLAY STATUS TRANSACTION` command when IMS is unable to initialize a transaction resource during restart processing.

For a DB/DC RSR tracking subsystem, `/DISPLAY STATUS` displays only DATABASE, LINE, LTERM, NODE, PTERM, and USER information.

The /DISPLAY STATUS DATABASE and /DISPLAY STATUS commands show the status of online forward recovery (OFR) for those databases in an RSR tracking subsystem that have OFR in progress.

The /DISPLAY STATUS LUNAME command can indicate that a command such as /STO LUNAME ALL INPUT or /STO LUNAME ALL OUTPUT was entered before the /DISPLAY that applies to all future LU 6.2 inbound or outbound conversations.

The /DISPLAY STATUS TMEMBER command displays the IMS Open Transaction Manager Access (OTMA) transaction pipes that are stopped. If a transaction pipe is not stopped, its status is UNRESTRICTED. The command output includes information about stopped transaction pipes that are defined to super members and the super member name if a super member is used to manage the hold queue of a regular member.

## Equivalent IMS type-2 commands

The following table shows variations of the /DISPLAY STATUS command and the IMS type-2 commands that perform similar functions.

Table 309. Type-2 equivalents for the /DISPLAY STATUS command

Task	/DISPLAY STATUS command	Similar IMS type-2 command
Displays information about databases with the specified status.	/DISPLAY STATUS DB	QUERY DB STATUS(ALLOCF, ALLOCS, BACKOUT, EEQE, LOCK, NOTINIT, NOTOPEN, OFR, OLR, OPEN, RECALL, RECOV, RNL, STOSCHD, STOUPDS)
Displays status.	/DISPLAY STATUS LTERM	QUERY LTERM SHOW(STATUS)
Displays logical terminals with specified status.	/DISPLAY STATUS LTERM	QUERY LTERM STATUS(status) SHOW(STATUS)
Displays nodes with specified status.	/DISPLAY STATUS NODE	QUERY NODE STATUS(status)
Displays all programs that have status and what that status is.	/DISPLAY STATUS PGM	QUERY PGM STATUS(DB-NOTAVL, IOPREV, LOCK, NOTINIT, STOSCHD, TRACE)
Displays all Fast Path routing codes that have status and what that status is.	/DISPLAY STATUS RTC	QUERY RTC STATUS(ACTIVE, NOTINIT, NOTSCHD, STOQ)
Displays information about transactions with the specified status.	/DISPLAY STATUS TRANSACTION	QUERY TRAN NAME(tranname) STATUS (IOPREV, LCK, QERR, SUSPEND, STOQ, STOSCHD, USTO)
Displays users with specified status.	/DISPLAY STATUS USER	QUERY USER STATUS(status)

## Output fields

IMS resources that can be listed by the /DISPLAY STATUS command include resources in the states listed in the following table. The /DISPLAY STATUS command displays only conditions that require operator intervention.

Table 310. /DISPLAY STATUS resource states displayed

Resource	State
CLASS	STOPPED

Table 310. /DISPLAY STATUS resource states displayed (continued)

Resource	State
DATABASE	ALLOCF, BACKOUT, EEQE, INQONLY, LOCK, NOTINIT, NOTOPEN
DATABASE ADS	PRE-OPEN FAILED, SEVERE-ERROR, UNAVAILABLE, COPY-PHASE, FORMAT-PHASE
DATABASE AREA	RECOVERY NEEDED, STOPPED
LINE	IDLE, NOIN, NOOUT, NOQUEUE, NOTOPEN, PSTOPPED, PUR, RESP, RESP-INP, STOPPED
LINK	ACTV, COLD, ERE, IDLE, NOTIDLE-A, NOTIDLE-B, NOTIDLE-Cxx, NOTOPEN, NRE, N/A, PSTOPPED
LTERM/MSNAME	LOCK, PSTOPPED, PUR, QERROR, QLOCK, STOPPED
LUNAME	STO, STO-INP, STO-OUTP
NODE	CLSDST, CONV-HLD, C1INOP, C2INOP, C3INOP, C4INOP, DEACTIVATED, DISCONNECTED, FORCE, INOP, LOCK, LOST, OPNDST, QUIESCED, RESP, RESP-INP, RESP-INP-FP, RESYNC, SHUT, STOPPED
PROGRAM	DB-STOPD, I/O PREVENT, LOCK, NOTINIT, STOPPED
PTERM	COMPINOP, CONV-HLD, EXCL, INOP, LOCK, LOOPTEST, NOIN, NOTOPEN, NOOUT, NOQUEUE, PSTOPPED, PUR, RESP, RESP-INP, STOPPED, TEST
RTCODE	NOTINIT, STOPPED
TMEMBER	STO
TRAN	LOCK, NOTINIT, PSTOPPED, PUR, QERROR, QSTP, SPND, STOPPED, USTOPPED
USER	CONV-HLD, DEADQ, RESP, RESP-INP, RESP-INP-FP, STOPPED

## Examples

The following are examples of the /DISPLAY STATUS command.

### Example 1 for /DISPLAY STATUS command

Entry ET:

```
/DISPLAY STATUS
```

Response ET:

```
**TRAN****PSBNAME
STATUS UNRESTRICTED
**DATABASE**
BMACC1   NOTOPEN,NOTINIT,STOPPED
DEDB01
-AREA01   RECOVERY-NEEDED
DEDEB03
-AREA02
  ADS01   PRE-OPEN FAILED
DEDB04   NOTOPEN,STOPPED
-AREA01   RECOVERY-NEEDED
  ADS01   UNAVAILABLE
**PROGRAM***
HIMASN01  STOPPED
SWITCH    STOPPED
```

```

**NODE****
CT3275   DISCONNECTED
CT3277A  DISCONNECTED
CT3277B  SHUT
W612     DISCONNECTED
**LINE*****
4        STOPPED, IDLE
5        STOPPED, IDLE, NOTOPEN
8        STOPPED, IDLE
10       STOPPED, IDLE, NOTOPEN
**PTERM*****
3        1 INOP, STOPPED
3        2 INOP, STOPPED
4        2 INOP, PSTOPPED
5        1 INOP, STOPPED
5        2 INOP, STOPPED
6        1 INOP, STOPPED
**LTERM*****
LU6A     STOPPED
VA01     QLOCK
**CLASS*****
2        STOPPED
**RTCODE****
STATUS UNRESTRICTED
**LINK*****
1        PSTOPPED, IDLE, NOTOPEN
3        PSTOPPED, IDLE, COLD
4        PSTOPPED, IDLE, N/A
5        PSTOPPED, IDLE, NOTOPEN
**MSNAME****
STATUS UNRESTRICTED
**USER*****
STATUS UNRESTRICTED
**LUNAME**TPNAME****
STATUS UNRESTRICTED
*92198/132223*

```

*Example 2 for /DISPLAY STATUS command*

Entry ET:

```
/DISPLAY STATUS DATABASE
```

Response ET:

```

**DATABASE**
DEDB01
-AREA01   STOPPED
DEDB02
-AREA02   NOTOPEN
-AREA03
  ADS03   PRE-OPEN FAILED
DEDB03
-AREA05   STOPPED, NOTOPEN
  ADS03   PRE-OPEN FAILED
  ADS08   UNAVAILABLE
DEDB04   STOPPED
DEDB05   NOTOPEN, NOTINIT
-AREA01   STOPPED
-AREA08   STOPPED, NOTOPEN
DEDB06   STOPPED
-AREA03
  ADS03   UNAVAILABLE
DEDB0777 NOTOPEN, LOCK

```

```
-AREA0555  STOPPED,NOTOPEN
ADS03333  SEVERE ERROR
ADS08888  UNAVAILABLE
*90263/092128*
```

*Example 3 for /DISPLAY STATUS command*

Entry ET:

```
/DISPLAY STATUS LUNAME
```

Response ET:

```
**LUNAME/TPNAME**
LU2
-TPN2          STO
LU5
-TPNAME1234567890+  STO
1234567890
LU3          STO-INP
LU4          STO-OUTP
LU1          STO-INP, STO-OUTP
*95229/155100*
```

Entry ET:

```
/DISPLAY STATUS LUNAME
```

Response ET:

```
**LUNAME/TPNAME**
LU2          STO-INP, STO-OUTP
-TPN2          STO
LU5          STO-INP, STO-OUTP
-TPN5          STO
-TPNAME1234567890+  STO
1234567890
LU3          STO-INP, STO-OUTP
LU4          STO-INP, STO-OUTP
LU1          STO-INP, STO-OUTP
FUTURE INPUT:  STO
FUTURE OUTPUT: STO
*95229/155146*
```

*Example 4 for /DISPLAY STATUS command*

Entry ET:

```
/DISPLAY STATUS NODE
```

Response ET:

```
**NODE*****
NLUTP6S  DISCONNECTED
NLU1     STOPPED, DISCONNECTED
NLUTP6P
-A       HELD
-N/A     STOPPED, DISCONNECTED
-E       FORCE
-C
*90231/040059*
```

Entry ET:

```
/DISPLAY STATUS RTCODE
```

Response ET:

```

**RTCODE****
DCL          STOPPED
EMHCOBOL    STOPPED
EMHPLI      STOPPED
EMHTX       STOPPED
TXCDBP04    STOPPED
TXCDBP05    STOPPED
TXCDBP06    STOPPED
*91068/114900*

```

*Example 5 for /DISPLAY STATUS command*

Entry ET:  
 /DISPLAY STATUS TMEMBER

Response ET:  
 \*\*TMEMBER/TPIPE\*\*  
 CLIENT1  
 -TPIPE1 STO  
 \*94168/095431\*

*Example 6 for /DISPLAY STATUS command*

Entry ET:  
 /DISPLAY STATUS TRANSACTION

Response ET:  
 \*\*TRAN\*\*\*\*PSBNAME  
 TPN4 DFSCPIC PUR  
 TPN5 DFSCPIC STOPPED  
 TPN6 DFSCPIC I/O PREVEN  
 TPN7 DFSCPIC USTOPPED  
 TRAN1 LU2PGM1 PSTOPPED  
 TRAN2 LU2PGM2 PSTOPPED  
 \*90332/114253\*

*Example 7 for /DISPLAY STATUS command*

Entry ET:  
 /DISPLAY STATUS TMEMBER

Response ET:  
 \*\*TMEMBER/TPIPE\*\*  
 HWS001 SM01  
 -TPIPE1 STO  
 -TPIPE3 STO  
 SM01 SM01  
 -TPIPE1 STO  
 -TPIPE2 STO  
 -TPIPE3 STO  
 HWS003  
 -TPIPE1 STO  
 \*05049/095431\*

*Example 8 for /DISPLAY STATUS command*

Entry ET:  
 /DISPLAY STATUS CLASS

Response ET:



## Output fields

The following output fields are displayed for the /DISPLAY STRUCTURE command:

### STATUS

The status conditions that can be displayed. Possible status values are:

#### AVAILABLE

The structure is available to be used by IMS.

#### CONNECTED

IMS is connected to the structure.

#### DISCONNECTED

IMS is not connected to the structure.

#### IN-OVERFLOW

The structure is in overflow mode.

#### REBLD-INPROG

A rebuild is in progress for the structure.

#### SHUTDOWN-STRCHKPT

A structure checkpoint will be taken by CQS for the structure during a CQS normal shutdown.

#### STRCHKPT-INPROG

A structure checkpoint is in progress for the structure.

#### UNAVAILABLE

The structure is not available to be used by IMS.

### STRUCTURE NAME

Indicates the name of the list structure.

**TYPE** Returns the list structure type, which can be either of the following:

#### MSGQ

Indicates a MSGQ primary or overflow structure.

#### EMHQ

Indicates an EMHQ primary or overflow structure.

## Examples

Entry ET:

```
/DISPLAY STRUCTURE ALL
```

Response ET:

```
STRUCTURE NAME  TYPE  STATUS
MSGQ1           MSGQ  CONNECTED, AVAILABLE, SHUTDOWN-STRCHKPT
EMHQ1           EMHQ  CONNECTED, AVAILABLE, IN-OVERFLOW
*95200/170817*
```

Entry ET:

```
/DISPLAY STRUCTURE MSGQ111 EMHQ1
```

Response ET:

```
STRUCTURE NAME  TYPE  STATUS
MSGQ111 IS INVALID
EMHQ1           EMHQ  DISCONNECTED
*95201/121800
```

---

## /DISPLAY SUBSYS command

Use the /DISPLAY SUBSYS command to display information about an external subsystem that is not a coordinator controller (CCTL) subsystem. ESS TYPE is now displayed on the second line of the /DIS SUBSYS command output.

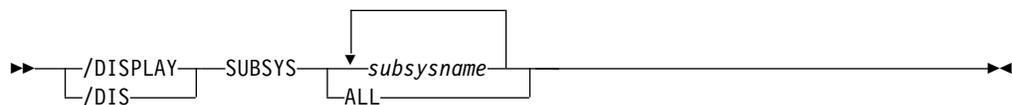
Subsections:

- "Environment"
- "Syntax"
- "Usage notes"
- "Output fields"
- "Examples" on page 718

### Environment

The /DISPLAY SUBSYS command is valid in DB/DC, DBCTL, and DCCTL environments.

### Syntax



### Usage notes

When used without the OASN keyword, the SUBSYS keyword displays the status of the connection between IMS and the external subsystem (not CCTL), as well as all application programs communicating with the external subsystem (not CCTL). The OASN keyword is used with the SUBSYS keyword to display all OASNs associated with the specified external subsystem. The external subsystem is not a CCTL subsystem. See "/DISPLAY OASN SUBSYS command" on page 648 for more information.

### Output fields

Subsystem status terms can be generated by the /DISPLAY SUBSYS command. Dependent region status terms can be generated by the /DISPLAY SUBSYS or /DISPLAY ACTIVE command.

#### *Subsystem status terms*

##### CONN

The IMS control region has successfully completed a host system IDENTIFY request to the external subsystem, making the two subsystems aware of each other's existence so they can begin a normal dialog.

##### NOT CONN

The external subsystem is in an idle state; that is, the external subsystem has not been the object of the /STOP SUBSYS command, or the external subsystem initialization exit indicated not to issue the IDENTIFY REQUEST (connect)

**CONN IN PROGRESS**

The connection process for the specified subsystem is in progress

**STOPPED**

The specified subsystem has been successfully stopped using the /STOP SUBSYS command and all region connections to the specified external subsystem have been terminated

**STOP IN PROGRESS**

The /STOP SUBSYS command is in progress. Before it successfully completes, all active connections to the specified subsystem from all IMS regions must be quiesced.

**TERM IN PROGRESS**

An internal termination of the subsystem connection is underway. This type of termination might be instigated by one of the following conditions:

- The external subsystem termination service exit was called to initiate subsystem disconnect processing.
- IMS drives the external subsystem termination in a quiesce fashion.
- IMS drives the external subsystem termination in a quick or catastrophic fashion.

A X'5609' log record for subsystem disconnect is created for the termination, which indicates the type of external subsystem termination that is in progress.

An IMS shutdown does not result in this condition.

**INVALID SUBSYSTEM NAME = XXXX**

The specified subsystem name has not been defined in the IMS subsystem PROCLIB member

**SUBSYSTEM XXXX NOT DEFINED BUT RECOVERY OUTSTANDING**

The specified subsystem name has not been defined to IMS in the external subsystem PROCLIB member, but IMS still has outstanding recovery elements from a previous execution when the indicated subsystem was known

The command recognition character will also be displayed for the external subsystem.

**START-AF**

The specified subsystem that is attached by the DB2 z/OS Resource Recovery Services (RRS) attachment facility has been successfully started by using the /START command, and dependent region connections to this system will be permitted.

**STOP-AF**

The specified subsystem which was attached by the DB2 z/OS Resource Recovery Services (RRS) attachment facility has been successfully stopped by using the /STOP command, and no more dependent region connections to this system will be permitted.

*Dependent region status terms***CONN**

An IMS dependent region has successfully completed a host system IDENTIFY request to the external subsystem as a result of an application having been scheduled into the dependent region. In an MPP, the application does not have to issue an external subsystem call to cause a connection.

## CONN, ACTIVE

An IMS application program has established communication with an external subsystem (for example, has issued at least one external subsystem call). At this point a thread exists between the IMS application program and the external subsystem.

The absence of a PSB name for a thread indicates that a connection to the external subsystem exists but an application program is not currently occupying the region. The presence or absence of an LTERM name indicates whether a region is message driven.

LTERM is the name of the terminal. LUNAME is displayed in place of LTERM when the originating terminal is LU6.2 or OTMA unless the LUNAME information is in transition, in which case, the literal UNKNOWN will be displayed.

## Examples

The following are examples of the /DISPLAY SUBSYS command.

### *Example 1 for /DISPLAY SUBSYS command*

The external subsystem that is being displayed, V10A, is a DB2 subsystem. It has the command recognition character "@" and is currently active.

Entry ET:

```
/DISPLAY SUBSYS ALL
```

Response ET:

SUBSYS	CRC	REGID	PROGRAM	LTERM	STATUS
ESSTYPE					
V10A	@				CONN
DB2			1	DSN8IC0	FPE0001 CONN, ACTIVE
			2	DSN8IC2	FPE0002 CONN, ACTIVE
V10B	-				CONN
DB2			3	DSN8IC3	FPE0003 CONN, ACTIVE
*13242/145525*		SYS3			

### *Example 2 for /DISPLAY SUBSYS command*

The external subsystem that is being displayed, V10A, is a DB2 subsystem. It has the command recognition character "@" and is currently stopped.

TSO SPOC input:

```
DIS SUBSYS ALL
```

TSO SPOC output:

```
IMSp1ex . . . . . : PLEX1
Routing . . . . . : SYS3
Start time. . . . . : 2013.256 16:01:17.63
Stop time . . . . . : 2013.256 16:01:17.66
Return code . . . . : 00000000
Reason code . . . . : 00000000
Reason text . . . . :
Command master. . . : SYS3
```

MbrName Messages

```

-----
SYS3      SUBSYS  CRC REGID PROGRAM  LTERM   STATUS
SYS3      ESSTYPE
SYS3      V10A    @
SYS3      DB2
SYS3      *13256/160117*

```

## /DISPLAY SYSID TRANSACTION command

Use the /DISPLAY SYSID TRANSACTION command to display the IDs of the local and remote systems associated with the transaction. Values displayed for the local and remote IDs can range from 1 to 2036.

Subsections:

- "Environment"
- "Syntax"
- "Examples"

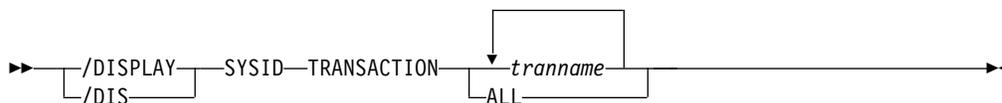
### Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

Table 311. Valid environments for the /DISPLAY SYSID TRANSACTION command and keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
/DISPLAY	X	X	X
SYSID	X		X
TRANSACTION	X		X

### Syntax



### Examples

Entry ET:

```
/DISPLAY SYSID TRANSACTION SKS1 SKS2 SKS3 SKS4
```

Response ET:

```

TRAN      PSBNAME  RID   LID
SKS1      DFSDDL1  21    11
SKS2      DFSDDL2  NA     11
SKS3      DFSDDL3  NA     11
SKS4      DFSDDL4  32    12
*89184/142436*

```

---

## /DISPLAY TIMEOVER command

Use the /DISPLAY TIMEOVER command to display all nodes that have been waiting for a response for a longer time than specified in the /TRACE SET ON TIMEOUT command.

Subsections:

- "Environment"
- "Syntax"
- "Usage notes"
- "Examples"

### Environment

The /DISPLAY TIMEOVER command is valid in DB/DC and DCCTL environments.

### Syntax

→ `/DISPLAY` `TIMEOVER` `time#` →  
└── `/DIS` ──┘

### Keywords

The following keywords can be specified for the /DISPLAY TIMEOVER command:

#### *time#*

Specifies the time period in minutes. This parameter is required. The number of minutes must be between 1 and 60.

### Usage notes

If no nodes are found that have been waiting longer than *time#* minutes, then the message NO NODE WITH TIMEOVER *time#* FOUND is displayed. An error message is displayed and the command is rejected if the timeout trace facility failed during IMS initialization. User names are displayed for ISC nodes, but non-ISC nodes show N/A.

### Examples

Entry ET:

```
/DISPLAY TIMEOVER 10
```

Response ET:

NODE	USER	TYPE	LAST OUTPUT	START TIME
SLU1A	N/A	SLU1	88118/095914	
LU6NDPC	LU6SPC	LUT6	88118/095921	
SLU1C	N/A	SLU1	88118/095929	
SLU1B	N/A	SLU1	88118/095937	
SLU1D	N/A	SLU1	88118/095943	
LU6NDPA	LU6SPA	LUT6	88118/095949	
*88118/101150*				

## Example for ISC TCP/IP terminals

Entry ET:

```
/DISPLAY TIMEOVER 10
```

Response ET:

```
NODE      USER      LINK#  TYPE  LAST OUTPUT START TIME
CICS      IMSUS01   N/A    TCP   12088/171914
*12088/101948*
```

---

## /DISPLAY TMEMBER command

Use the /DISPLAY TMEMBER command to display the current target member status for IMS Open Transaction Manager Access (OTMA) clients and servers.

Subsections:

- “Environment”
- “Syntax”
- “Keywords” on page 722
- “Usage notes” on page 723
- “Output fields” on page 724
- “Examples” on page 727

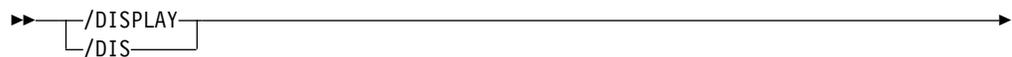
### Environment

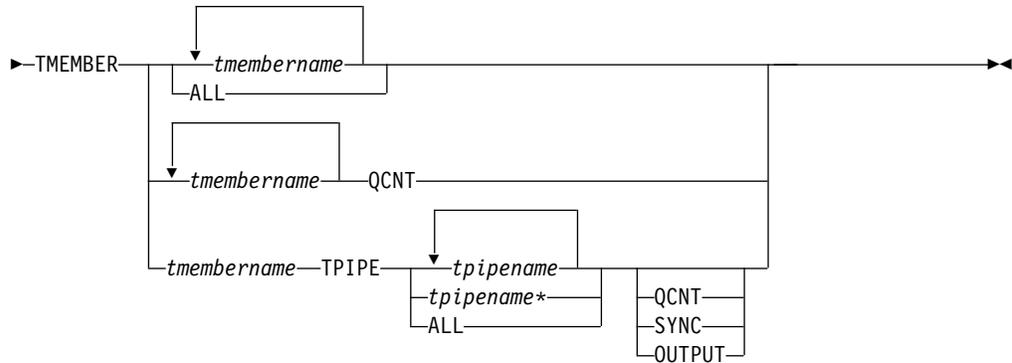
The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

Table 312. Valid environments for the /DISPLAY TMEMBER command and keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
/DISPLAY	X	X	X
QCNT	X		X
TMEMBER	X		X
TPIPE	X		X

### Syntax

►► 



## Keywords

The following keywords can be specified for the /DISPLAY TMEMBER command.

### QCNT

The /DISPLAY TMEMBER command with the QCNT parameter specifies that global queue count information is to be displayed. If the member specified is a super member, global queue count information is displayed for the super member. If the member specified is a regular member whose hold queue output is managed by a super member, the name of the super member is displayed along with the global queue count information for the regular member.

This count represents the message counts of messages that can be processed by the IMS system where the command is issued. This count includes messages that can be processed by any IMS system (messages with no affinity), plus messages that can be processed by the IMS system where the command is issued (messages with an affinity to the IMS system that issued the command).

The /DISPLAY TMEMBER QCNT form of the command is valid only in a shared-queues environment.

### TPIPE

The /DISPLAY TMEMBER TPIPE command can be issued to display transaction pipe status for a member. If the member specified is a super member, transaction pipe status is displayed for the super member. If the member specified is a regular member whose hold queue output is managed by a super member, the name of the super member is displayed along with the queue count information.

The tpipe name can be ended with a generic character (\*). This generic form of tpipe name can also be used with QCNT or SYNC keywords. The tpipe name with a generic character cannot be mixed with other tpipe names. The valid formats for using a tpipe name with the generic character are:

```
/DIS TMEMBER membername TPIPE tpipe_name*
/DIS TMEMBER membername TPIPE tpipe_name* QCNT
/DIS TMEMBER membername TPIPE tpipe_name* SYNC
```

### QCNT

The /DISPLAY TMEMBER TPIPE command with the QCNT parameter specifies that global queue count information is to be displayed. If the member specified is a super member, global queue count information is displayed for the transaction pipes associated with the super member. If the member specified is a regular member whose hold queue output is

managed by a super member, the name of the super member is displayed along with the global queue count information.

The /DISPLAY TMEMBER TPIPE QCNT form of the command is valid only in a shared-queues environment.

### SYNC

The /DISPLAY TMEMBER TPIPE SYNC command displays:

- The number of active synchronous callout messages
- The number of synchronous callout messages waiting for response
- The number of resume tpipe requests that are currently active on the TPIPE
- The resume tpipe option
- The resume tpipe mode
- The tpipe status

The /DISPLAY TMEMBER TPIPE SYNC command can be considered as a superset of the /DISPLAY TMEMBER TPIPE command.

### OUTPUT

The /DISPLAY TMEMBER TPIPE OUTPUT command can be used to display the output counts for both primary and holdq queues when the OTMA client is holdq-capable (for example, IMS Connect). This parameter is ignored for clients that are not holdq-capable (such as IBM MQ) and when IMS has shared queues enabled.

When support for multiple active resume tpipe requests is enabled (MULTIRTP=Y), the output includes the unique identifier, the resume tpipe option, and the mode for each resume tpipe request that is active on a tpipe.

**Important:** Specifying the OUTPUT parameter can double the number of lines for the display. Take care when using the ALL or masked TPIPE names for TMEMBERs with a large number of TPIPEs.

If support for multiple active resume tpipe requests is enabled (MULTIRTP=Y), the output can include even more lines if there are many active resume tpipe requests on the tpipes.

The OUTPUT parameter is mutually exclusive with the SYNC and QCNT parameters.

## Usage notes

The /DISPLAY TMEMBER command includes five variations: /DISPLAY TMEMBER QCNT, /DISPLAY TMEMBER TPIPE, /DISPLAY TMEMBER TPIPE QCNT, /DISPLAY TMEMBER TPIPE SYNC and /DISPLAY TMEMBER TPIPE OUTPUT.

In a shared-queues environment, use the QCNT keyword to obtain the current message queue counts on the shared queues structure. The local queue counts (ENQCT, DEQCT, and QCT) value that is returned on this command when the QCNT keyword is not specified represents the messages being processed by the IMS system where this command is issued. Do not use local queue counts for queue evaluation purposes.

## Output fields

The following output fields are displayed for the /DISPLAY TMEMBER command:

### DEQCT

Total number of messages that are dequeued from the OTMA tpipe for this member. If the OTMA hold queue is used for a client that is capable of the OTMA hold queue, such as IMS Connect, the number of the messages in the hold queue of this tpipe is also included. In a shared-queues environment, this field shows only the messages dequeued for the local subsystem.

### DRUEXIT

The name of any OTMA Destination Resolution exit routine associated with the tmember.

### ENQCT

Total number of messages that are enqueued on the OTMA tpipe for this member. If the OTMA hold queue is used for a client that is capable of the OTMA hold queue, such as IMS Connect, the number of the messages in the hold queue of this tpipe is also included. In a shared-queues environment, this field shows only the messages enqueued for the local subsystem.

### GROUP/MEMBER

Each member in each z/OS cross-system coupling facility (XCF) group.

When you issue /DISPLAY TMEMBER ALL, the server is always the first member displayed.

**INPT** The maximum concurrent input message count for this member that can be waiting at the same time to be processed. If the YTIBs reach the INPT value, an OTMA FLOOD condition exists and the subsequent input messages from the member will be rejected.

### MODE

The resume tpipe mode, which can be one of the following:

- S - Synchronous callout messages
- B - Both asynchronous messages and synchronous callout messages
- A - Asynchronous messages

### NO-COT

The current number of ICAL messages received for this tpipe. If the number is greater than or equal to 65535, it will be reset to 1.

**OPT** The resume tpipe option, which can be one of the following:

- N - No auto flow of messages. When a client issues a resume tpipe call and no message is present, the resume tpipe call ends.
- S - Single message. When a client issues a resume tpipe call, if multiple messages are present, only one message is retrieved. If no message is present, the call ends.
- A - Auto flow of messages. The client retrieves one message at a time. If no messages exist in the tpipe, the client waits for subsequent messages from the hold queue.
- W - Single message with wait option. The client will retrieve only one message. If there are no messages in the hold queue, the client will wait until a message arrives, then the call ends. Only one message is retrieved regardless of how many messages are in the hold queue.

This option is specified in the IRM\_F5 field of the IRM prefix in the messages that are sent to IMS Connect.

**QCT** Total number of messages that are still in the queue for OTMA tpipe for this member. If the OTMA hold queue is used for a client that is capable of the OTMA hold queue, such as IMS Connect, the number of the messages in the hold queue of this tpipe is also included. In a shared-queues environment, this field shows only the messages enqueued for the local subsystem.

**RTA** The number of active Resume TPIPE requests. This field is displayed only when the SYNC keyword is specified on the command.

**RTQ** The number of queued resume tpipe requests to be processed. This field is displayed only for the command issued with the SYNC keyword.

#### **SECURITY**

The security status for each member or server

The security status can be one of the following:

- NONE
- JOIN
- CHECK
- FULL
- PROFILE

#### **SYNCOT**

The number of active synchronous callout messages.

**TIB** The number of current input messages processed by OTMA for this member. This number will be incremented when a new transaction is received, and it will be decremented after OTMA enqueues a CM0 input transaction or OTMA completes a CM1 transaction. There are cases where orphaned transaction instance blocks (YTIBs) are created. These YTIBs will not be deleted until IMS is cold started.

#### **TPCNT**

The maximum TPIPE count for each OTMA member. For an OTMA member other than the IMS server, this field indicates the current TPIPE count for that member. For the IMS server, this field indicates the global TPIPE limit, which is the highest TPIPE limit set for the OTMA members. IMS displays a nonzero value only when the MAXTP= keyword is specified for this member in the OTMA client descriptor DFSYDTx member of the IMS PROCLIB data set. If the MAXTP= keyword is not specified for any of the members in the OTMA client descriptor, the TPCNT field of the IMS server member will be zero.

**T/O** The timeout value for OTMA send-then-commit message responses.

#### **USER-STATUS**

The user status for each member.

The status can be one of the following:

- ACCEPT TRAFFIC
- DISCONNECTED
- IN SLOWDOWN
- MAX TPIPE: MAX TPIPE indicates that the OTMA member's TPIPE count has reached the limit set. This status is cleared when the global TPIPE count falls below 50%.
- SERVER
- SUPER MEMBER

- WAIT BID
- WAIT RESPONSE

If the member specified is a regular member whose member name is displayed as part of the command output. If the member specified is a super member, the character string 'SUPER MEMBER' is displayed for the user status. Since a super member does not interface with XCF, the XCF-STATUS field is left blank. Since super members do not process input messages, OTMA security authorization is not performed.

#### **XCF-STATUS**

The XCF status for each member.

The status can be one of the following:

- ACTIVE
- CREATED
- FAILED
- FLOOD
- NOT DEFINED
- QUIESCED
- STO-INPUT
- UNKNOWN

If the server leaves the XCF group (for example, because of a /STOP OTMA command), then the XCF status is displayed as NOT DEFINED.

The output from the /DISPLAY TMEMBER QCNT or /DISPLAY TMEMBER TPIPE QCNT command includes the following:

#### **AFFINITY**

The IMSID (or the first 7 bytes of the RSENAME, if XRF capable) of the IMS to which the output messages have affinity.

#### **GBLQCT**

The total number of OTMA output messages on the global queue. The global queue count includes messages that were placed on the global queue before the latest cold start of the owning IMS. Synchronous and asynchronous OTMA output messages are included in the global queue count. Messages on an OTMA asynchronous hold queue for IMS Connect are also included in the global queue count.

#### **MEMBER/TPIPE**

OTMA member name. If the TPIPE keyword is specified, the transaction pipe name is also displayed. The first line displays the OTMA member name. The second and subsequent lines display the names of the transaction pipes associated with the OTMA member.

One of the following status terms is displayed for the /DISPLAY TMEMBER command specified with the TPIPE keyword:

**DQF** Dequeue request for OTMA REPresynch command failed.

#### **FLOOD**

Input from this tmember is suppressed because of a message flood condition.

#### **HLDQ**

Queue counts for the holdq queue.

- INPCT** The current number of input messages received for this tpipe.
- PMRY** Queue counts for the primary queue.
- REP** IMS is waiting for an OTMA REPresynch command from the client.
- REQ** IMS is sending an OTMA REQresynch command to the client.
- RSF** Reset request failed for an OTMA REPresynch command.
- STO** The transaction pipe is stopped.
- STO-INPUT**  
Input from this tmember is stopped.
- SYN** The transaction pipe is being synchronized.
- TBR** IMS is waiting for an OTMA TBResynch command from the client.
- TMP** The transaction pipe is temporary.
- TRA** The transaction pipe is being traced.
- WAIT\_A**  
The transaction pipe is waiting for an ACK or NAK for a commit-then-send (CM0) output response.
- WAIT\_H**  
The transaction pipe is waiting for an ACK or NAK for a commit-then-send (CM0) output response from the asynchronous hold queue.
- WAIT\_R**  
The transaction pipe is waiting for an ACK or NAK response to a commit-then-send (CM0) output message that was sent to a remote IMS system through a TCP/IP connection managed by IMS Connect.
- WAIT\_S**  
The transaction pipe is waiting for an ACK or NAK for a synchronous callout message.

## Examples

The following are examples of the /DISPLAY TMEMBER command.

### *Example 1 for /DISPLAY TMEMBER command*

Entry ET:

```
/DISPLAY TMEMBER ALL
```

Response ET:

GROUP/MEMBER	XCF-STATUS DRUEXIT	USER-STATUS T/O	SECURITY	TIB	INPT	SMEM
HARRY						
IMS1	ACTIVE	SERVER	FULL			
HWS001	ACTIVE	ACCEPT TRAFFIC	FULL	0	0	SM01
HWS001	HWSYDRU0	5				
SM01		SUPER MEMBER				SM01
MQS001	ACTIVE	ACCEPT TRAFFIC	FULL	0	0	
MQS001	MQSYDRU0	10				

Explanation: The /DISPLAY TMEMBER ALL command can be issued to display the current status of IMS server and members. This command is functionally

equivalent to the /DIS OTMA command. The member information is displayed in random order. In the above example, HARRY is the XCF group name for IMS and its OTMA members. IMS1 is the server running with OTMA activated. HWS001 and MQS001 are OTMA client members. The SM01 is an OTMA super member for asynchronous output messages. Each OTMA member has two-line status information

*Example 2 for /DISPLAY TMEMBER command*

Entry ET:

```
/DISPLAY TMEMBER MQS001
```

Response ET:

GROUP/MEMBER	XCF-STATUS	USER-STATUS	SECURITY	TIB	INPT	SMEM
	DRUEXIT	T/O				
MQS001	ACTIVE	ACCEPT TRAFFIC	FULL	0	0	
MQS001	MQSYDRU0	10				

**Explanation:** In the above example, MQS001 is an OTMA member without using super member function. There are two-line status information for the MQS001.

*Example 3 for /DISPLAY TMEMBER command*

Entry ET:

```
/DISPLAY TMEMBER HWS001 HWS002 HWS003
```

Response ET:

GROUP/MEMBER	XCF-STATUS	USER-STATUS	SECURITY	TIB	INPT	SMEM
	DRUEXIT	T/O				
HWS001	ACTIVE	ACCEPT TRAFFIC	FULL	0	0	SM01
HWS001	HWSYDRU0	5				
HWS002	ACTIVE	ACCEPT TRAFFIC	FULL	0	0	SM01
HWS002	HWSYDRU1	5				
HWS003	ACTIVE	ACCEPT TRAFFIC	FULL	0	0	
HWS003	HWSYDRU2	5				

**Explanation:** In the above example, HWS001, HWS002 and HWS003 are all regular OTMA members. OTMA hold queue output for HWS001 and HWS002 is managed by super member SM01. HWS003 manages its own OTMA hold queue output without using the super member.

*Example 4 for /DISPLAY TMEMBER command*

Entry ET:

```
/DISPLAY TMEMBER HWS001 SM01
```

Response ET:

GROUP/MEMBER	XCF-STATUS	USER-STATUS	SECURITY	TIB	INPT	SMEM
HWS001	ACTIVE	ACCEPT TRAFFIC	FULL	0	0	SM01
SM01		SUPER MEMBER				SM01

**Explanation:** HWS001 is a regular member whose hold queue output is managed by super member SM01. SM01 is a super member. The XCF-STATUS field and the SECURITY field are left blank. The super member name is repeated in the SMEM field.

*Example 5 for /DISPLAY TMEMBER command*

Entry ET:

```
/DISPLAY TMEMBER CLIENT1 TPIPE ALL
```

Response ET:

MEMBER/TPIPE	ENQCT	DEQCT	QCT	STATUS
CLIENT1				TRA
-TPIPE1	0	0	0	TRA,STO
-TPIPE2	2	2	0	TRA,STO
-TPIPE3	1	0	1	TRA,STO

\*94165/170756\*

*Example 6 for /DISPLAY TMEMBER command*

Entry ET:

```
/DIS TMEMBER CLIENT1 QCNT
```

Response ET:

MEMBER/TPIPE	GBLQCT	AFFINITY
CLIENT1	34	SYS3

\*03203/124035\*

*Example 7 for /DISPLAY TMEMBER command*

Entry ET:

```
/DIS TMEMBER CLIENT1 TPIPE TPIPE3 TPIPE1 QCNT
```

Response ET:

MEMBER/TPIPE	GBLQCT	AFFINITY
CLIENT1		
-TPIPE3	16	SYS3
-TPIPE1	8	SYS3

\*03202/143253\*

*Example 8 for /DISPLAY TMEMBER command*

Entry ET:

```
/DIS TMEMBER CLIENT1 TPIPE ALL QCNT
```

Response ET:

MEMBER/TPIPE	GBLQCT	AFFINITY
CLIENT1		
-TPIPE3	16	SYS3
-TPIPE2	16	SYS3
-TPIPE1	2	SYS3

\*03202/143253\*

*Example 9 for /DISPLAY TMEMBER command*

Entry ET:

```
/DIS TMEMBER HWS001 QCNT
```

Response ET:

MEMBER/TPIPE	GBLQCT	AFFINITY	SMEM
HWS001	1	SYS3	SM01

\*03203/124035\*

Explanation: HWS001 is a regular member whose hold queue output is managed by super member SM01. There is one CM0 output message on the OTMA output queue for member HWS001. That message has affinity to SYS3. The /DIS TMEMBER QCNT command can be issued with the super member name (SM01) specified as the member name to determine the number of messages on the super member's output queue.

*Example 10 for /DISPLAY TMEMBER command*

Entry ET:

```
/DIS TMEMBER SM01 QCNT
```

Response ET:

MEMBER/TPIPE	GBLQCT	AFFINITY	SMEM
SM01	46		SM01
*03203/124035*			

Explanation: SM01 is a super member. There are 46 CM0 output messages on the output queue for SM01. These messages do not have affinity to any IMS subsystem. Any member whose hold queue output is managed by SM01 can retrieve the output messages.

*Example 11 for /DISPLAY TMEMBER command*

Entry ET:

```
/DISPLAY TMEMBER HWS003 TPIPE TPIPE1
```

Response ET:

MEMBER/TPIPE	ENQCT	DEQCT	QCT	STATUS	SMEM
HWS003					
-TPIPE1	3	2	1	TRA	

Explanation: HWS003 is a regular member that manages its own hold queue output. There is one output message queued for TPIPE1. This message is either on the I/O PCB output queue or the hold queue.

*Example 12 for /DISPLAY TMEMBER command*

Entry ET:

```
/DISPLAY TMEMBER HWS001 TPIPE TPIPE1
```

Response ET:

MEMBER/TPIPE	ENQCT	DEQCT	QCT	STATUS	SMEM
HWS001					SM01
-TPIPE1	0	0	0	TRA	

Explanation: HWS001 is a regular member whose hold queue output is managed by super member SM01. There are no output messages queued to HWS001, but there might be some messages queued to the hold queue of super member SM01.

*Example 13 for /DISPLAY TMEMBER command*

Entry ET:

```
/DISPLAY TMEMBER SM01 TPIPE TPIPE1 TPIPE2
```

Response ET:

MEMBER/TPIPE	ENQCT	DEQCT	QCT	STATUS	SMEM
SM01					SM01
-TPIPE1	3	2	1	TRA	
-TPIPE2	2	2	0	TRA	

Explanation: SM01 is a super member with transaction pipes defined for TPIPE1 and TPIPE2. Local queue count information is displayed for TPIPE1 and TPIPE2. There is one output message on the hold queue for TPIPE1.

**Example 14 for /DISPLAY TMEMBER command**

Entry ET:

```
/DIS TMEMBER SM01 TPIPE ALL QCNT
```

Response ET:

MEMBER/TPIPE	GBLQCT	AFFINITY	SMEM
SM01			SM01
-TPIPE1	2		
-TPIPE2	5		
-TPIPE3	16		
*03202/143253*			

Explanation: SM01 is a super member. There are 23 CM0 output messages on the output queues for SM01. Two messages are queued for TPIPE1, five for TPIPE2 and sixteen for TPIPE3. These messages do not have affinity to any IMS subsystem. Any member whose hold queue output is managed by super member SM01 can retrieve the output messages.

**Example 15 for /DISPLAY TMEMBER command**

Entry ET:

```
/DIS TMEMBER HWS1 TPIPE ALL SYNC
```

Response ET:

```
(screen 1)
MMR/TPIP      ENQCT      DEQCT      QCT      INPCT OPT MODE
HWS1
-TPIPE1          0          0          0          0 W  S

(scrolled to the right, screen 2)
SYNCOT NO-COT RTA RTQ STATUS  SMEM  IMS1
      1      1      1      1 WAIT_S  IMS1
```

Explanation: HWS1 is a regular member with a transaction pipe defined for TPIPE1. The resume tpipe call is to retrieve only synchronous callout messages from the hold queue ("S" for MODE). The request message (from the hold queue) has been sent to the client, IMS Connect, which in turn will send the message to the application client as a result of the resume tpipe call with MODE of "S" and OPT of "W". This accounts for the count of 1 under the column heading SYNCOT. The count of 1 under the column heading NO-COT indicates that the accumulated ICAL count for this OTMA tpipe is 1. Because OTMA is waiting for ACK or NAK, WAIT\_S is displayed under column heading STATUS.

To extend the processing, assuming that the client application sends an ACK, the WAIT\_S under column heading STATUS will no longer be displayed, and the count of 1 under column heading NO-COT will still be displayed. This output indicates that OTMA has processed one ICAL message for the IMS callout

application. Once the OTMA has received the message response and passed it back to the IMS callout application, SYNCOT will show 0, and NO-COT will still show 1. And, when OTMA has completed processing the message response, the resume tpipe call ends, and the output display should show OPT='-', MODE='-', SYNCOT=0, NO-COT=1, and STATUS=' '.

The number of queued tpipe requests is 1.

Entry ET:

```
/DISPLAY TMEMBER HWS1 TPIPE TPIPE1
```

Response ET:

MEMBER/TPIPE	ENQCT	DEQCT	QCT	INPCT	STATUS	SMEM	IMS1
HWS1							
-TPIPE1	0	0	0	0	WAIT_S	IMS1	

Explanation: TPIPE1 is waiting for a NAK or ACK for a synchronous callout message.

**Example 16 for /DISPLAY TMEMBER command**

Entry ET:

```
/DISPLAY TMEMBER HWS1
```

Response ET:

GROUP/MEMBER	XCF-STATUS	USER-STATUS	SECURITY	TIB	
INPT	SMEM	DRUEXIT	T/O	TPCNT	ACEEAGE
XCFGRP1					
-HWS1	ACTIVE	MAX	TPIPE	FULL	0
5000					
-HWS1		200	1000	999999	

Explanation: In this example, the TPCNT field indicates the current TPIPE count for member HWS1. MAX TPIPE indicates that the TPIPE count for the OTMA member HWS1 has reached the limit set. This status is cleared when the global TPIPE count falls below 50%.

**Example 17 for /DISPLAY TMEMBER command**

Entry ET:

```
/DISPLAY TMEMBER HWS1
```

Response ET:

GROUP/MEMBER	XCF-STATUS	USER-STATUS	SECURITY	TIB		
INPT	SMEM	DRUEXIT	T/O	TPCNT	ACEEAGE	MAXTP
XCFGRP1						
-HWS1	ACTIVE	ACCEPT	TRAFFIC	FULL	0	
5000						
-HWS1		200	500	120	0	

Explanation: In this example, the TPCNT field indicates that the current TPIPE count for member HWS1 is 500 and that there is no MAX TPIPE limit set for this member in the OTMA client descriptor.

**Example 18 for /DISPLAY TMEMBER command**

Entry ET:

```
/DISPLAY TMEMBER HWS2
```

Response ET:

GROUP/MEMBER INPT SMEM	XCF-STATUS	USER-STATUS	SECURITY	TIB
	DRUEXIT	T/O	TPCNT	ACEEAGE MAXTP
XCFGRP1				
-HWS2	ACTIVE	ACCEPT	TRAFFIC	FULL 0
5000				
-HWS2		200	500	120 10000

Explanation: In this example, the TPCNT field indicates that the current TPIPE count for member HWS2 is 500 and that there is a MAX TPIPE limit of 10000 set for this member.

*Example 19 for /DISPLAY TMEMBER command*

Entry ET:

```
/DIS TMEM HWS1 TPIPE ALL
```

Response ET:

DFS000I	MEMBER/TPIPE	ENQCT	DEQCT	QCT	INPCT	STATUS	SMEM
DFS000I	HWS1						
DFS000I	-9999	0	0	0	1		
DFS000I	-CLIENT02	3	2	1	2		
DFS000I	-CLIENT01	8	5	3	5		

*Example 20 for /DISPLAY TMEMBER command*

Entry ET:

```
/DIS TMEM HWS1 TPIPE ALL OUTPUT
```

Response ET:

DFS000I	MEMBER/TPIPE	ENQCT	DEQCT	QCT	INPCT	STATUS	SMEM
DFS000I	HWS1						
DFS000I	-CLIENT03	1	1	0	1	PMRY	
DFS000I	-CLIENT03	1	0	1	0	HLDQ	
DFS000I	-9999	0	0	0	1	PMRY	
DFS000I	-CLIENT02	1	1	0	1	PMRY	
DFS000I	-CLIENT04	1	1	0	1	PMRY	
DFS000I	-CLIENT04	1	0	1	0	HLDQ	

*Example 21 for /DISPLAY TMEMBER command*

Entry ET:

```
/DIS TMEM HWS1 TPIPE CLIENT* OUTPUT
```

Response ET:

DFS000I	MEMBER/TPIPE	ENQCT	DEQCT	QCT	INPCT	STATUS	SMEM
DFS000I	HWS1						
DFS000I	-CLIENT02	2	2	0	2	PMRY	
DFS000I	-CLIENT02	1	0	1	0	HLDQ	
DFS000I	-CLIENT01	5	5	0	5	PMRY	
DFS000I	-CLIENT01	3	0	3	0	HLDQ	

*Example 22 of /DISPLAY TMEMBER command for TPIPEs that support multiple active resume TPIPE requests*

Entry ET:

/DIS TMEM HWS1 TPIPE OTMACL\* OUTPUT

Response ET:

DFS000I	MEMBER/TPIPE	ENQCT	DEQCT	QCT	INPCT	STATUS
DFS000I	HWS1					
DFS000I	-OTMACL99	0	0	0	0	PMRY WAIT_S
DFS000I	-OTMACL99	0	0	0	0	HLDQ
DFS000I	-OTMACL99	RT 1234567890123456		OPT A	MODE B	
DFS000I	-OTMACL99	RT 1234567890123457		OPT A	MODE B	WAIT_S
DFS000I	-OTMACL99	RT 1234567890123458		OPT A	MODE B	

**Related reference:**

“Status and attributes for the /DISPLAY command” on page 523

“/DISPLAY OTMA command” on page 652

---

## /DISPLAY TRACE command

Use the /DISPLAY TRACE command to display the status and options of the current IMS traces. TRACE must be followed by one or more keywords or by the ALL parameter. There is no default.

Subsections:

- “Environment”
- “Syntax” on page 735
- “Keywords” on page 735
- “Usage notes” on page 736
- “Equivalent IMS type-2 commands” on page 736
- “Output fields” on page 736
- “Examples” on page 737

### Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

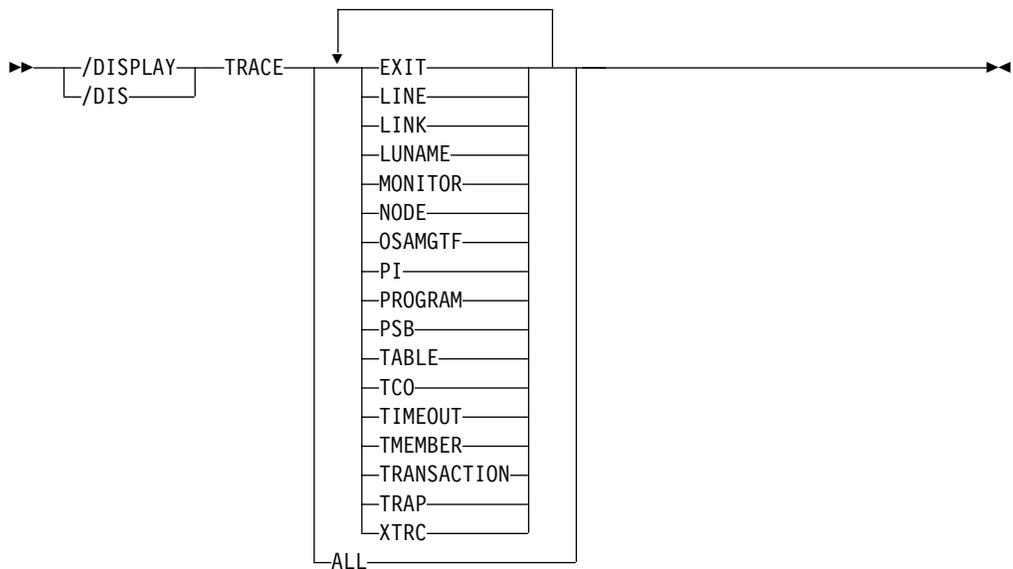
*Table 313. Valid environments for the /DISPLAY TRACE command and keywords*

Command / Keywords	DB/DC	DBCTL	DCCTL
/DISPLAY	X	X	X
EXIT	X		X
LINE	X		X
LINK	X		X
LUNAME	X		X
MONITOR	X	X	X
NODE	X		X
OSAMGTF	X	X	
PI	X	X	
PROGRAM	X	X	X
PSB	X	X	X
TABLE	X	X	X
TCO	X	X	X

Table 313. Valid environments for the /DISPLAY TRACE command and keywords (continued)

Command / Keywords	DB/DC	DBCTL	DCCTL
TIMEOUT	X		X
TMEMBER	X		X
TRACE	X	X	X
TRANSACTION	X		X
TRAP	X		X
XTRC	X	X	X

## Syntax



## Keywords

The following keywords are valid for the /DISPLAY TRACE command:

### ALL

In a DBCTL system, ALL displays only DB-related traces. It does not show DC-related traces such as LINE, LUNAME, LINK, NODE, TRANSACTION, TRAP, or TIMEOUT.

In a DCCTL system, ALL displays only DC-related traces. It does not show DB-related traces such as OSAMGTF or PI.

For a DB/DC RSR tracking subsystem, ALL displays only LINE, NODE, PSB, TABLE, and TRAP information.

The display shows which external data set is active and whether the OLDS is being traced for trace logging. The command also indicates whether XTRC is usable and the status of the XTRC data sets.

For other keywords, see the descriptions for corresponding output fields under "Output fields" on page 736.

## Usage notes

The /DISPLAY TRACE ALL and /DISPLAY TRACE TMEMBER commands can be used to display IMS OTMA transaction pipes that are currently being traced for a OTMA client. The command output includes transaction pipes that are defined to super members. The super member name is displayed if the member is a super member or if the member is a regular member whose hold queue output is managed by a super member.

## Equivalent IMS type-2 commands

The following table shows variations of the /DISPLAY TRACE command and the IMS type-2 commands that perform similar functions.

Table 314. Type-2 equivalents for the /DISPLAY TRACE command

Task	/DISPLAY TRACE command	Similar IMS type-2 command
Displays nodes with trace status.	/DISPLAY TRACE NODE	QUERY NODE STATUS(TRACE)

## Output fields

The following output fields are displayed by the /DISPLAY TRACE command:

**EXIT** Displays user exit tracing. Currently, only tracing for the DFSMSCE0 user exit is supported. The following table lists the display exit entry points.

Table 315. Display exit entry points for tracing

LRDI	LRIN	LRLT	LRTR
PRCH	PRGU	PRIS	TR62
TRBT	TROT	TRVT	

**Note:** The display indicates the status of each exit entry point. If the status is ON, the trace is active for the selected entry point. If the status is OFF, the trace is inactive for the selected entry point. If the status is N/A, the specified DFSMSCE0 trace entry point does not exist in the IMS.

**LINE** The relative line number and line type of all traced lines.

**LINK** The link number and link partner for all logical links.

### LUNAME

All LU 6.2 application programs that are being traced.

### OSAMGTF

Whether the OSAM Buffer Trace facility is active or inactive.

### MONITOR

The status of the IMS Monitor, and if active, the options that are specified for it.

### NODE

The VTAM or ISC TCP/IP node name and the node type. If the node is defined for a parallel session, each traced user of that node is displayed. If the users are not active, an N/A is displayed.

### PI

The status of the program isolation trace, and if active, the status of the options and the volume level.

### PROGRAM

The traced program name.

**PSB** The PSB name and trace option for each traced PSB.

**TABLE**

The trace status, logging status, total number of entries in each trace table, and the volume level.

For a DL/I trace, the volume level is displayed in the subcategories.

For an RSR tracking subsystem, TABLE does not show information about the LUMI, QMGR, RETR, SCHED, SQTT, or SUBS table traces.

In an IMSplex, the OCMD, RM, and SCI table traces are included in the table traces displayed.

**TCO** Displays whether tracing is active for the Time Controlled Operation (TCO).

**TIMEOUT**

The status of the IMS VTAM I/O Timeout Detection Facility, including the specifications given when the /TRACE SET ... TIMEOUT command was issued. The /DISPLAY TRACE TIMEOUT command will show active or inactive status, the timeout value in minutes, and the action to be taken when timeout occurs (AUTO, MSG, or no action). Option and Timeout values will be displayed only when the facility is active.

**TMEMBER**

The IMS Open Transaction Manager Access (OTMA) transaction pipes that are currently being traced for the OTMA client.

**TRANSACTION**

The name of the transaction and the applicable PSB name for each trace.

**TRAP** The trap that is currently set, either 1 or 2; that trap is enabled for MFS serviceability aids.

**XTRC** External trace status. Whether XTRC is usable and the status of the XTRC data sets are also displayed.

**Examples**

The following are examples of the /DISPLAY TRACE command.

*Example 1 for /DISPLAY TRACE command*

Entry ET:

```
/DISPLAY TRACE ALL
```

Response ET:

```
IMS ACTIVE TRACES
```

```
LINE      TYPE
   1  CONSOLE
   4  3270 R
```

```
LINK      PARTNER
   2      AC
```

```
NODE      TYPE  USERS
SLUP1     SLUP
LU6NDPA   LUT6  N/A    N/A    N/A
VAT11     3277
```

```
NO PSB TRACES FOUND
```

MONITOR IS INACTIVE

TRAN CODE PSB-NAME  
ADDINV DFSSAM04

PROGRAM NAME

APOL1  
DEBS  
TACP1

TABLE	TRACE	LOGGING	ENTRIES	VOLUME	SYS3
RETR	ON	N/A	128/PST	N/A	SYS3
DL/I		OFF	2268		SYS3
DL/I	ON			HIGH	SYS3
LOCK	ON			HIGH	SYS3
PI	OFF			OFF	SYS3
LATC	ON	OFF	3024	MEDIUM	SYS3
DISP	ON	OFF	1260	MEDIUM	SYS3
SCHD	ON	OFF	630	MEDIUM	SYS3
SUBS	OFF	OFF	504	OFF	SYS3
DLOG	ON	OFF	756	MEDIUM	SYS3
FAST	OFF	OFF	252	OFF	SYS3
STRG	ON	OFF	1260	MEDIUM	SYS3
IDC0	OFF	OFF	2268	OFF	SYS3
LUMI	OFF	OFF	1512	OFF	SYS3
OTMT	OFF	OFF	1008	OFF	SYS3
QMGR	ON	OFF	1008	MEDIUM	SYS3
ORTT	OFF	OFF	1008	OFF	SYS3
OCMD	OFF	OFF	1008	OFF	SYS3
CSLT	OFF	OFF	1008	OFF	SYS3
RRST	OFF	OFF	1512	OFF	SYS3
MSCT	OFF	OFF	1260	OFF	SYS3
FPTT	OFF	OFF	1008	OFF	SYS3
DIAG	ON	ON	8064	MEDIUM	SYS3
SECT	OFF	OFF	1008	OFF	SYS3

*Example 2 for /DISPLAY TRACE command*

Entry ET:

/DISPLAY TRACE LINE LINK NODE

Response ET:

IMS ACTIVE TRACES

LINE	TYPE	USERS
1	CONSOLE	
30	3270 R	
30	3270 R	
LINK	PARTNER	
4	RF	
NODE	TYPE	USERS
L3270A	3277	
LU6NDPA	LUT6	N/A N/A N/A
LEVEL: 4	MODULE: ALL	
*90290/143519*		

*Example 3 for /DISPLAY TRACE command*

Entry ET:

/DISPLAY TRACE LUNAME

Response ET:

IMS ACTIVE TRACES

LUNAME	TPNAME	
LUNAME3		INPUT
LUNAME3	TPNAME1	OUTPUT
LUNAME3	TPNAME7890123456+	OUTPUT
	789END	
LUNAME4		INPUT
*90332/114253*		

Entry ET:

/DISPLAY TRACE LUNAME

Response ET:

IMS ACTIVE TRACES

LUNAME/TPNAME	
NET1.LU1	INP
-TPN1	
NET2.LU1	OUTP
-TPN2	OUTP
LU3	INP, OUTP
-TPN3	OUTP
-TPN4	OUTP
NET4.LU3	OUTP
NET5.LU5	INP
*93069/114435*	

*Example 4 for /DISPLAY TRACE command*

Entry ET:

/DISPLAY TRACE OSAMGTF

Response ET:

IMS ACTIVE TRACES  
OSAMGTF TRACE IS INACTIVE  
\*92107/090819

Entry ET:

/TRACE SET ON OSAMGTF

Response ET:

DFS058I 09:08:32 TRACE COMMAND COMPLETED

Entry ET:

/DISPLAY TRACE OSAMGTF

Response ET:

IMS ACTIVE TRACES  
OSAMGTF TRACE IS ACTIVE  
\*92107/090841\*

*Example 5 for /DISPLAY TRACE command*

Entry ET:

/DISPLAY TRACE MONITOR

Response ET:

```

IMS ACTIVE TRACES
MONITOR IS ACTIVE:  LA SCHD  APMQ  APDB
  APDB dbname/partition-name/area-name ...
  REGION reg# ... region-name ...
  INTERVAL #seconds EXPIRING hh:mm:ss.tt
*98029/114114*

```

*Example 6 for /DISPLAY TRACE command*

Entry ET:

```

/TRACE SET ON TABLE QMGR
/TRACE SET ON TABLE SQTT
/DISPLAY TRACE TABLE

```

Response ET:

```

IMS ACTIVE TRACES  SYS3S VoIume  SYS3
      SYS3  N/A  128/PST  N/A  SYS3
TABLE  TRACE  LOGGING  ENTRIES  VOLUME  SYS3
RETR   ON     N/A     128/PST  N/A     SYS3
DL/I   ON     OFF      2268    HIGH    SYS3
  DL/I  ON     OFF      2268    HIGH    SYS3
  LOCK  ON     OFF      2268    HIGH    SYS3
  PI    OFF    OFF      2268    OFF     SYS3
LATC   ON     OFF      3024    MEDIUM  SYS3
DISP   ON     OFF      1260    MEDIUM  SYS3
SCHD   ON     OFF      630     MEDIUM  SYS3
SUBS   OFF    OFF      504     OFF     SYS3
DLOG   ON     OFF      756     MEDIUM  SYS3
FAST   OFF    OFF      252     OFF     SYS3
STRG   ON     OFF      1260    MEDIUM  SYS3
IDC0   OFF    OFF      2268    OFF     SYS3
LUMI   OFF    OFF      1512    OFF     SYS3
OTMT   OFF    OFF      1008    OFF     SYS3
QMGR   ON     OFF      1008    MEDIUM  SYS3
ORRT   OFF    OFF      1008    OFF     SYS3
OCMD   OFF    OFF      1008    OFF     SYS3
CSLT   OFF    OFF      1008    OFF     SYS3
RRST   OFF    OFF      1512    OFF     SYS3
MSCT   OFF    OFF      1260    OFF     SYS3
FPTT   OFF    OFF      1008    OFF     SYS3
DIAG   ON     ON       8064    MEDIUM  SYS3

SECT   OFF    OFF      1008    OFF     SYS3
*04197/152006*  SYS3

```

*Example 7 for /DISPLAY TRACE command*

Entry ET:

```

/DISPLAY TRACE TCO

```

Response ET:

```

IMS ACTIVE TRACES

TCO IS NOT ACTIVE
*89100/170509*

```

*Example 8 for /DISPLAY TRACE command*

Entry ET:

```

/DISPLAY TRACE TIMEOUT

```

Response ET:

IMS ACTIVE TRACES  
VTAM I/O TIMEOUT FACILITY IS INACTIVE  
\*89033/170200\*

*Example 9 for /DISPLAY TRACE command*

Entry ET:  
/DISPLAY TRACE TIMEOUT

Response ET:  
IMS ACTIVE TRACES  
VTAM I/O TIMEOUT FACILITY IS ACTIVE  
OPTION = NONE TIMEOUT VALUE = 0  
\*89110/131429\*

*Example 10 for /DISPLAY TRACE command*

Entry ET:  
/DISPLAY TRACE TIMEOUT

Response ET:  
IMS ACTIVE TRACES  
VTAM I/O TIMEOUT FACILITY IS ACTIVE  
OPTION = AUTO TIMEOUT VALUE = 10  
\*89033/170200\*

*Example 11 for /DISPLAY TRACE command*

Entry ET:  
/DISPLAY TRACE TIMEOUT

Response ET:  
IMS ACTIVE TRACES  
VTAM I/O TIMEOUT FACILITY IS ACTIVE  
OPTION = MSG TIMEOUT VALUE = 60  
\*89033/131630\*

*Example 12 for /DISPLAY TRACE command*

Entry ET:  
/DISPLAY TRACE TMEMBER

Response ET:  
IMS ACTIVE TRACES  
TMEMBER/TPIPE TYPE  
CLIENT1  
-TPIPE1 TRA  
\*94168/095325\*

*Example 13 for /DISPLAY TRACE command*

Entry ET:

/DISPLAY TRACE XTRC

Response ET:

IMS ACTIVE TRACES

IMS EXTERNAL TRACE IS USABLE

SYS3

XTRC DDNAME	ALLOC	STATUS	STATUS
DFSTRA01	DYNA	UNALLOCATED	CLOSED
DFSTRA02	DYNA	UNALLOCATED	CLOSED
DFSTRA0T	DYNA	UNALLOCATED	CLOSED

\*91091/124215\*

*Example 14 for /DISPLAY TRACE command*

Entry ET:

/DIS TRACE EXIT

Response ET:

55/DIS TRACE EXIT

IEE600I REPLY TO 55 IS;/DIS TRACE EXIT

DFS000I IMS ACTIVE TRACES IMS3

DFS000I IMS3

DFS000I	EXIT	FUNC	STATUS	IMS3
DFS000I	DFSMSCE0	TRBT	ON	IMS3
DFS000I	DFSMSCE0	TRVT	ON	IMS3
DFS000I	DFSMSCE0	TR62	OFF	IMS3
DFS000I	DFSMSCE0	TROT	OFF	IMS3
DFS000I	DFSMSCE0	LRTR	ON	IMS3
DFS000I	DFSMSCE0	LRLT	ON	IMS3
DFS000I	DFSMSCE0	LRDI	ON	IMS3
DFS000I	DFSMSCE0	LRIN	ON	IMS3
DFS000I	DFSMSCE0	PRCH	ON	IMS3
DFS000I	DFSMSCE0	PRIS	N/A	IMS3
DFS000I	DFSMSCE0	PRGU	ON	IMS3
DFS000I	*99096/103002*			IMS3

56 DFS996I \*IMS READY\* IMS3

Response ET:

39/DIS TRACE EXIT

IEE600I REPLY TO 39 IS;/DIS TRACE EXIT

DFS000I IMS ACTIVE TRACES IMS3

DFS000I IMS3

DFS000I	EXIT	FUNC	STATUS	IMS3
DFS000I	DFSMSCE0	TRBT	N/A	IMS3
DFS000I	DFSMSCE0	TRVT	N/A	IMS3
DFS000I	DFSMSCE0	TR62	N/A	IMS3
DFS000I	DFSMSCE0	TROT	N/A	IMS3
DFS000I	DFSMSCE0	LRTR	N/A	IMS3
DFS000I	DFSMSCE0	LRLT	N/A	IMS3
DFS000I	DFSMSCE0	LRDI	N/A	IMS3
DFS000I	DFSMSCE0	LRIN	N/A	IMS3
DFS000I	DFSMSCE0	PRCH	N/A	IMS3
DFS000I	DFSMSCE0	PRIS	N/A	IMS3
DFS000I	DFSMSCE0	PRGU	N/A	IMS3
DFS000I	*99105/222611*			IMS3

46 DFS996I \*IMS READY\* IMS3

*Example 15 for /DISPLAY TRACE Command*

Entry ET:

```
/DISPLAY TRACE TMEMBER
```

Response ET:

```
IMS ACTIVE TRACES
```

TMEMBER/TPIPE	TYPE	SMEM
HWS001		SM01
-TPIPE1	TRA	
SM01		SM01
-TPIPE1	TRA	
-TPIPE2	TRA	
HWS003		
-TPIPE1	TRA	

\*05049/095325\*

Explanation: HWS001 is a regular member whose hold queue output is managed by super member SM01. HWS001 has one transaction pipe that is being traced. SM01 is a super member with two transaction pipes that are being traced. HWS003 also has a transaction pipe that is being traced. HWS003 is a regular member that manages its own hold queue output.

*Example 16 for /DISPLAY TRACE command*

Entry ET:

```
/DISPLAY TRACE TABLE
```

Response ET:

```
IMS ACTIVE TRACES SYS3S Volume SYS3
SYS3 N/A 128/PST N/A SYS3
TABLE TRACE LOGGING ENTRIES VOLUME SYS3
RETR ON N/A 128/PST N/A SYS3
DL/I OFF 2268 SYS3
DL/I ON HIGH SYS3
LOCK ON HIGH SYS3
PI OFF OFF SYS3
LATC ON OFF 3024 MEDIUM SYS3
DISP ON OFF 1260 MEDIUM SYS3
SCHD ON OFF 630 MEDIUM SYS3
SUBS OFF OFF 504 OFF SYS3
DLOG ON OFF 756 MEDIUM SYS3
FAST OFF OFF 252 OFF SYS3
STRG ON OFF 1260 MEDIUM SYS3
IDCO OFF OFF 2268 OFF SYS3
LUMI OFF OFF 1512 OFF SYS3
OTMT OFF OFF 1008 OFF SYS3
QMGR ON OFF 1008 MEDIUM SYS3
ORRT OFF OFF 1008 OFF SYS3
OCMD ON OFF 1008 ERROR SYS3
CSLT OFF OFF 1008 OFF SYS3
RRST OFF OFF 1512 OFF SYS3
MSCT OFF OFF 1260 OFF SYS3
FPTT OFF OFF 1008 OFF SYS3
DIAG ON ON 8064 MEDIUM SYS3

SECT OFF OFF 1008 OFF SYS3
*04197/152006* SYS3
```

*Example 17 for /DISPLAY TRACE command*

The following example shows the output returned for ISC TCP/IP nodes.

Entry ET:  
/DISPLAY TRACE TABLE

Response ET:  
IMS ACTIVE TRACES

NODE	TYPE	USERS
CICS	TCP	IMSUS01
LU6NDPA	TCP	LU6SPA

LEVEL: 4    MODULE: ALL  
\*12062/122024\*

**Related reference:**

 [QUERY NODE command \(Commands\)](#)

---

## /DISPLAY TRACKING STATUS command

In an RSR complex, use the /DISPLAY TRACKING STATUS command to display the status of a tracking subsystem. The /DISPLAY TRACKING STATUS command is valid on active and RSR tracking subsystems.

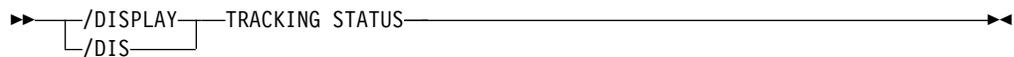
Subsections:

- “Environment”
- “Syntax”
- “Output fields: /DISPLAY TRACKING STATUS on an active subsystem”
- “Output fields: /DISPLAY TRACKING STATUS on a tracking subsystem” on page 746
- “Examples” on page 751

### Environment

The /DISPLAY TRACKING STATUS command is valid in DB/DC, DBCTL, and DCCTL environments.

### Syntax



### Output fields: /DISPLAY TRACKING STATUS on an active subsystem

/DISPLAY TRACKING STATUS on an active subsystem displays the following:

- Subsystem IMSID
- The global service group (GSG)
- The service group (SG)
- The service group status
- The status of the conversation with log router

/DISPLAY TRACKING STATUS on an active subsystem shows the following information:

**ACTIVE SUBSYSTEM**

The IMSID, global service group, service group, and service group status of the active subsystem

**IMSID**

The IMSID of the active subsystem.

**GSG-NAME**

The 1- to 8-character global service group (GSG) name. The global service group is the collection of all IMS subsystems in an RSR complex (such as online IMS, utilities, and so on) that access a particular set of databases. An active subsystem can only be defined as part of one global service group.

**SG-NAME**

The name of the service group (SG) of which the active subsystem is a part. The service group name is 1 to 8 characters long.

**SG-STATUS**

The service group status, which can be one of the following:

**IDENTIFICATION IN PROGRESS**

An the IMS logger is in the process of identifying to the Transport Manager Subsystem (TMS) but has not yet completed.

**IDENTIFIED**

The IMS logger has identified to the TMS. A status of IDENTIFIED appears as long as the logger is connected or identified to the TMS. This status also appears after a /START SERVGRP command.

**LOST** The IMS logger's connection to the TMS is gone because of TMS or VTAM failure.

**NO LOG ROUTER SG DEFINED**

There is an error and logger has no knowledge of a log router with which to communicate. There is no Log Router Service Group defined to DBRC.

**NOT IDENTIFIED**

The IMS logger has never identified to the TMS (that is, is in its initial state) or the IMS logger has not been able to identify since the most recent /STOP SERVGRP command.

**STOPPED**

The /STOP SERVGRP command was issued. If there is an active conversation, /STOP SERVGRP causes it to be terminated.

At each OLDS switch, the logger no longer attempts to establish a conversation with the log router. Logs are no longer sent to the tracking subsystem. A /START SERVGRP command is required to undo the STOPPED condition.

**ACTIVE'S CONVERSATIONS WITH LOG ROUTERS**

Status of the conversations between the active and tracking subsystems.

**SG-NAME**

The name of the service group (SG) of which the tracking subsystem is a part.

**RECORDS-NOT-SENT**

The number of log records that have not yet been sent to the tracker. The number of records is a number in the form of *nnnK*,

where K represents 1024 records. The number is rounded off to the nearest K. If the number of records exceeds 999K, then >999K is displayed.

This field applies only if the conversation status is SENDING LOGS. Otherwise, N/A is displayed.

#### **CONVERSATION STATUS**

The conversation status, which can be one of the following:

##### **ACCEPT IN PROGRESS**

A log router is in the process of accepting a conversation request, but has not yet allocated the conversation.

##### **ALLOCATION IN PROGRESS**

Indicates a conversation is in the process of being allocated.

##### **DEALLOCATION IN PROGRESS**

A conversation is in the process of being deallocated.

##### **NO ACTIVE CONVERSATION**

Indicates there is no active conversation between the logger and the log router. Although there is no conversation, the logger does know about the log router.

At each OLDS switch, the logger will attempt to establish a conversation with the log router until it is successful. A /START SERVGRP command can be issued to cause the logger to attempt to establish a conversation with the log router before the next OLDS switch.

##### **SENDING LOGS**

The logger has an active conversation with the log router and is sending logs to the tracking site.

##### **SUSPENDED LOGS**

The logger has an active conversation with the log router but has suspended sending logs to the tracking site because of resource shortage. The conversation is still intact.

### **Output fields: /DISPLAY TRACKING STATUS on a tracking subsystem**

When issued on a tracking subsystem, /DISPLAY TRACKING STATUS displays the following:

- Subsystem IMSID
- Readiness level
- Global service group (GSG)
- Service group (SG)
- Service group status
- Tracking status
- Tracking activity
- Gaps
- Isolated log sender status
- Online forward recovery (OFR) status
- The current and restart milestone indexes

- DL/I tracking status
- Fast Path tracking status

/DISPLAY TRACKING STATUS shows the same information for an IMS DB/DC tracking subsystem or a DBCTL tracking subsystem.

/DISPLAY TRACKING STATUS on a tracker shows the following information:

#### **TRACKING SUBSYSTEM**

The following tracking subsystem information is displayed:

##### **IMSID**

The IMSID of the tracking subsystem. The IMSID is 1 to 4 characters long.

##### **READINESS-LEVEL**

The readiness level of the tracking subsystem, which is either RECOVERY or DATABASE.

##### **GSG-NAME**

The global service group (GSG) name. The GSG is the collection of all IMS subsystems in an RSR complex, such as online IMS, utilities, and so on, which access a particular set of databases. A tracking subsystem tracks only one global service group. The global service group name is 1 to 8 characters long.

##### **SG-NAME**

The 1- to 8-character service group (SG) name.

#### **SG-STATUS**

The service group status, which can be IDENTIFIED, NOT IDENTIFIED, or STOPPED.

##### **IDENTIFIED**

The tracking subsystem has identified to the TMS, and can establish conversations with active loggers and isolated log sender (ILS) instances. IDENTIFIED appears as long as the tracking subsystem is connected or identified to the TMS.

##### **NOT IDENTIFIED**

The tracking subsystem has not attempted to identify to the TMS. NOT IDENTIFIED is the initial state.

##### **STOPPED**

The tracking subsystem is not identified to the TMS. This status can be the result of any of the following: a /STOP SERVGRP command entered from the tracking subsystem, an unsuccessful attempt to identify, or termination of the TMS.

#### **TRACKING STATUS**

The status of the tracking subsystem, which can be one of the following:

##### **INACTIVE**

The tracking subsystem is idle. The tracking subsystem has completed all tracking work and has received no more log records from the active subsystems it tracks.

For example, a status of INACTIVE appears if the active subsystems have shut down and the tracking subsystem has completed its tracking work.

## TRACKING

The tracking subsystem is tracking one or more active subsystems and has tracking work to do.

## PLANNED TAKEOVER REQUESTED

One of the active subsystems being tracked issued an /RTAKEOVER command, notified the tracker, and shut down. As soon as all of the active systems in the GSG have shut down, all log gaps have been filled, and log routing has completed, the tracking subsystem will shut down for the planned remote takeover.

## PLANNED TAKEOVER IN PROGRESS

An /RTAKEOVER command was entered on the active subsystem and all of the active subsystems being tracked have successfully shut down. The tracking subsystem is in the process of shutting down for the planned remote takeover. The tracking subsystem must shut down before active subsystems can be started at the new active site.

## UNPLANNED TAKEOVER IN PROGRESS

An /RTAKEOVER command was entered on the tracking subsystem. The tracking subsystem will shut down for the unplanned remote takeover after it has completed routing received log data.

## NUMBER OF ACTIVE SUBSYSTEMS

The number of active subsystems currently being tracked. It is followed by an integer.

## TRACKING ACTIVITY

Status of the conversations between the tracker and the loggers of the active subsystems being tracked. The following information is displayed for each active subsystem being tracked:

### NAME

The IMSID of the subsystem being tracked or the jobname for batch or batch backout being tracked. Subsystem names are 1 to 4 characters long. Job names are 1 to 8 characters long.

If the active subsystems are XRF-capable, the subsystem name displayed includes a hyphen and the one-character HSBID.

**Example:** The XRF alternate for IMSA would be displayed as IMSA-2.

**TYPE** Type of subsystem, which can be ONLINE, ACTIVE, BATCH, or BBO (batch backout):

### ONLINE

An online (not batch) subsystem that has not sent data to the tracking subsystem. The subsystem might be an active subsystem that has not completed restart or it might be an XRF alternate subsystem.

### ACTIVE

An online (not batch) subsystem that has sent data to the tracking subsystem.

### BATCH

A batch subsystem that has sent data to the tracking subsystem.

**BBO** A batch backout job that has sent data to the tracking subsystem.

**RECEIVED-LOG**

The universal coordinated time of the last log record received from the subsystem, BATCH job, or BBO job. The time stamp is of the format *hhmmss*, where *hh* represents hours, *mm* represents minutes, and *ss* represents seconds. RECEIVED-LOG does not apply to conversations with XRF-alternate subsystems, in which case N/A is displayed.

**ROUTED-LOG**

The universal coordinated time of the last log routed by the log router. The time stamp is of the format *hhmmss*, where *hh* represents hours, *mm* represents minutes, and *ss* represents seconds. ROUTED-LOG does not apply to conversations with XRF-alternate subsystems, in which case N/A is displayed.

**NOT-ROUTED**

The number of log records that have been received by the tracker but not yet routed by the log router. The number is in the format *nnnK*, where K represents 1024 log records. The number is rounded off to the nearest K. If this number exceeds 999K, then >999K is displayed. NOT-ROUTED does not apply to conversations with XRF-alternate subsystems, in which case N/A is displayed.

**STATUS**

The tracking status. If a conversation exists between the tracking subsystem and the active IMS, the status field contains CONV-ACT. If there is no conversation, the status field is left blank.

**GAPS** The status of gaps between the tracker and the active subsystems, BATCH jobs, or BBO jobs. One line is displayed for each gap associated with an active subsystem.

If the gap information is not known to the tracking subsystem, the PRILOG token associated with the gap is displayed as "MISSING LOG, PRILOG TOKEN=". The following gap information is displayed:

**NAME**

The IMSID of the active system for which there is a gap, or the jobname of the BATCH or BBO job for which there is a gap.

In an XRF environment, the NAME is the RSENAME rather than the IMSID.

**LOG-SEQ-FIRST**

The log sequence number, in hexadecimal, of the first log record in the gap

**IN-GAP**

The number of records, in decimal, that are in the gap. The number is of the form *nnnK*, where K represents 1024 records. The number is rounded off to the nearest K. If this number exceeds 999K, then >999K is displayed.

**RECEIVED**

The number of records, in decimal, that have been received. The

number is of the form *nnnK*, where K represents 1024 records. The number is rounded off to the nearest K. If this number exceeds 999K, then >999K is displayed.

**NOT-ROUTED**

The number of log records, in decimal, that have not been routed for the oldest gap. The number is of the form *nnnK*, where K represents 1024 records. If this number exceeds 999K, then >999K is displayed. This number does not apply to gaps other than the oldest one.

**STATUS**

The status of the gap, which is one of the following:

**FILLING**

The gap is currently being filled.

**WAITING**

The gap is not being filled yet. For example, the gap might be waiting to be filled if there are not enough conversations to send the log records, or a batch job is still running.

**PERMNT ERR**

There is a permanent error, such as a read error, that prevents this gap from being filled.

**UNAVAIL**

The log data needed to fill the gap is not available at the active site. The log or log data sets might have been deleted.

**ISOLATED LOG SENDER STATUS**

Status of isolated log sender conversations.

**SYSTEM NAME OF TRANSPORT MANAGER SUBSYSTEM**

If the tracking subsystem has a conversation with an isolated log sender, the system name of the TMS to which the isolated log sender is identified is displayed.

**NUMBER OF CONVERSATIONS**

Number of conversations with this instance of isolated log sender.

**ONLINE FORWARD RECOVERY STATUS**

The following online forward recovery (OFR) status is displayed:

**NUMBER OF STREAMS BEING PROCESSED**

Number of OFR processes initiated. The online forward recoveries are processed one at a time.

**OFR ROUTING TIME**

The latest close time of any log data set that has been or is currently being read for OFR.

**MILESTONE INDEX STATUS**

The following milestone information is displayed:

**CURRENT MILESTONE INDEX**

The current milestone index.

**RESTART MILESTONE INDEX**

The restart milestone index.

## **DLI TRACKING STATUS**

Status of DL/I database tracking.

### **TRACKING PSTs**

Number of DL/I database tracking PSTs currently in use. The number can be from 0 to 255.

### **USAGE OF TRACKING PSTs**

Average percent usage of DLI database tracking PSTs currently in use

### **BACKLOG OF REDO RECORDS IN DATASPACE**

The backlog number of redo records in the data space. The backlog is a measure of how well the tracker is keeping up with the workload. The backlog is a number in the form of *nnnK*, where K represents 1024 redo records. The number is rounded off to the nearest K. If the number of redo records exceeds 999K, then >999K is displayed.

## **FP TRACKING STATUS**

Status of Fast Path database tracking.

### **TRACKING PSTs**

Number of FP database tracking PSTs and the reason the PSTs are invoked. The number of tracking PSTs can be from 0 to 255. The reason the PSTs are invoked can be one of the following and only appears if there are one or more PSTs:

#### **AREAWRITE**

The PSTs are invoked by a write request for a specific area.

#### **MILESTONE**

The PSTs are invoked by milestone processing.

#### **THRESHOLD**

The PSTs are invoked by threshold write.

### **CURRENT LOG VOLUME IN DATASPACE**

The current log volume in Fast Path tracking data space. This number can be from 0 to *nnnnnnnK*, where K represents 1024 log records. The maximum possible number is 2000000K (two gigabytes). The number is rounded off to the nearest K.

### **MAX LOG VOLUME IN DATASPACE**

Maximum log volume in Fast Path tracking data space. This number can be from 0 to *nnnnnnnK*, where K represents 1024 log records. The maximum possible number is 2000000K (two gigabytes). The number is rounded off to the nearest K.

## **Examples**

The following are examples of the `/DISPLAY TRACKING STATUS` command.

### *Example 1 for /DISPLAY TRACKING STATUS command*

The following is an example of a `/DISPLAY TRACKING STATUS` entered from a tracking subsystem in Los Angeles. The two active systems in San Jose are named IMSA and IMSC. IMSA has an XRF alternate named IMSB. IMSC has an XRF alternate named IMSD. The Coordinated Universal Time is 23:32:09. The local Los Angeles time is 15:32:09. OFR is also in progress and has started processing log data up through 14:11:34 local time.

Entry ET:  
/DIS TRACKING STATUS

Response ET:

```
**** TRACKING SUBSYSTEM *****
IMSID READINESS-LEVEL GSG-NAME SG-NAME SG-STATUS
IMST RECOVERY          MSGSG1  STLSITE2 IDENTIFIED
**** TRACKING STATUS: TRACKING
NUMBER OF ACTIVE SUBSYSTEMS:                4
**** TRACKING ACTIVITY *****
NAME      TYPE      RECEIVED-LOG  Routed-LOG  NOT-ROUTED  STATUS
IMSD      ONLINE    N/A          N/A         N/A         CONV-ACT
IMSC      ACTIVE    15:31:00    15:29:54   2K         CONV-ACT
IMSB      ONLINE    N/A          N/A         N/A         CONV-ACT
IMSA      ACTIVE    15:31:10    15:29:47   3K         CONV-ACT
**** GAPS *****
NAME      LOG-SEQ-FIRST  IN-GAP  RECEIVED  NOT-ROUTED  STATUS
IMSC      0000000000007D31  2K      OK        N/A         WAITING
**** ISOLATED LOG SENDER STATUS *****
SYSTEM NAME OF TRANSPORT MANAGER SUBSYSTEM:  TMPAR
NUMBER OF CONVERSATIONS:                    1
**** ONLINE FORWARD RECOVERY STATUS *****
NUMBER OF STREAMS BEING PROCESSED:          0
OFR ROUTING TIME:                          2002.090 14:11:34
**** MILESTONE INDEX STATUS *****
CURRENT MILESTONE INDEX:                    38
RESTART MILESTONE INDEX:                    37
**** DLI TRACKING STATUS *****
TRACKING PSTS:                              0
USAGE OF TRACKING PSTS:                     0%
BACKLOG OF REDO RECORDS IN DATASPACE:      0K
**** FP TRACKING STATUS *****
TRACKING PSTS:                              0
CURRENT LOG VOLUME IN DATASPACE:           0K
MAX LOG VOLUME IN DATASPACE:               0K
*02090/153209*
```

### *Example 2 for /DISPLAY TRACKING STATUS command*

The following is an example of a /DISPLAY TRACKING STATUS command entered on an active system IMSC, located in San Jose.

Entry ET:  
/DIS TRACKING STATUS

Response ET:

```
**** ACTIVE SUBSYSTEM *****
IMSID GSG-NAME SG-NAME SG-STATUS
IMSC  ACCTGRP1 SITESJ  IDENTIFIED
**** ACTIVE'S CONVERSATIONS WITH LOG ROUTERS *****
IMSID RECORDS-NOT-SENT
IMST          1K SENDING LOGS
*91317/150416*
```

---

## **/DISPLAY TRAN command**

Use the /DISPLAY TRAN command to display the status of transactions.

Subsections:

- "Environment" on page 753
- "Syntax" on page 753

- “Keywords”
- “Usage notes” on page 754
- “Equivalent IMS type-2 commands” on page 754
- “Output fields” on page 754
- “Examples” on page 755

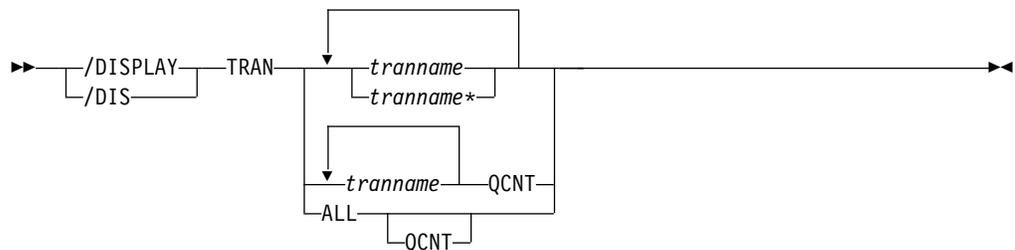
## Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

Table 316. Valid environments for the /DISPLAY TRAN command and keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
/DISPLAY	X	X	X
QCNT	X		X
TRAN	X		X

## Syntax



## Keywords

The following keyword can be specified for the /DISPLAY TRAN command:

### QCNT

Specifies that global queue count information is to be displayed. If the QCNT keyword is not specified, local queue counts are displayed. The QCNT keyword is valid only in a shared-queues environment.

IMS does not check that the specified name is a valid transaction in the IMS subsystem on which the command is entered.

The /DIS TRAN *tranname* QCNT command returns the count of messages on the shared-queue structure for transaction *tranname*. The message count includes the messages that can be processed by any IMS (messages with no affinity), plus messages that can be processed only by the IMS where the /DIS TRAN command was entered (messages with affinity to the IMS that issued the command). Thus, a /DIS TRAN *tranname* QCNT command that is routed to several IMS subsystems in an IMSplex environment might show different queue counts on different IMS subsystems if there are messages on the shared-queue structure that are queued with affinity to an IMS.

The /DISPLAY TRAN ALL QCNT command displays all the transactions on the shared queues with a global queue count. This command returns one line for each transaction that has messages queued to the global (non-affinity) queue on the shared queue structure. This line shows the number of messages

that are queued to the global queue of that transaction and indicates the messages that can be processed by any IMS. Additionally, the /DIS TRAN ALL QCNT command returns one line for each transaction that has messages queued with affinity to an IMS in the IMSplex. This line shows the number of messages queued with affinity, as well as the IMSID of the IMS to which they have affinity.

On the IMS where this message will be processed, you might need to register this transaction for affinity message processing by entering a type-1 /START TRAN *trancode* AFFINITY command or a type-2 UPDATE TRAN NAME(*trancode*) START(SCHD) OPTION(AFFIN) command.

For the AFFINITY status, the type-1 command can display messages on the shared queues that have affinity, but not the local AFFINITY status of the transaction (*trancode*) name in IMS. To query the affinity status of a transaction, use a type-2 QUERY TRAN STATUS(AFFIN) command or a QUERY TRAN NAME(*trancode*) SHOW(STATUS) command.

## Usage notes

The TRAN keyword parameter can be generic.

IMS Open Transaction Manager Access (OTMA) supports an architected format for the output from the /DISPLAY TRAN command. The architected output is returned to the OTMA client, not to an IMS operator.

In a shared-queues environment, use the QCNT keyword to obtain the current message queue counts on the shared queues structure. The local queue counts (ENQCT, DEQCT, and QCT) value that is returned on this command when the QCNT keyword is not specified represents the messages being processed by the IMS system where this command is issued. Do not use local queue counts for queue evaluation purposes.

## Equivalent IMS type-2 commands

The following table shows variations of the /DISPLAY TRAN command and the IMS type-2 commands that perform similar functions.

Table 317. Type-2 equivalents for the /DISPLAY TRAN command

Task	/DISPLAY TRAN command	Similar IMS type-2 command
Displays information about a transaction.	/DISPLAY TRAN <i>trancode</i>	QUERY TRAN NAME( <i>trancode</i> ) SHOW(ALL)
Displays all of the transactions.	/DISPLAY TRAN ALL	QUERY TRAN SHOW(ALL)
Displays all of the transactions on the shared queues with a global queue count.	/DISPLAY TRAN <i>trancode</i> QCNT	QUERY TRAN NAME( <i>trancode</i> ) SHOW(QCNT)

## Output fields

The /DISPLAY TRAN command displays the transaction name, class (CLS), current priority(CP), normal priority (NP), local priority (LP), local enqueue queue count (ENQCT), local queue count (QCT), limit count (LCT), processing limit count (PLCT), parallel limit count (PARLM), the number and size of output segments for a specified transaction code (SESZ, SEGNO), and the number of regions the transaction is currently scheduled in (RC), all on the first line.

The next two indented lines show the PSB name associated with the transaction and the status. DFSCPIC is displayed as the PSB name for CPI Communications driven transactions that have not issued a DL/I APSB call to allocate a PSB.

Fast Path exclusive transaction codes have only the transaction code and class returned, where the class is displayed as FPE. The queue counts are not shown because Fast Path processing bypasses the IMS message queues.

Fast Path potential transactions have two lines returned, the first line with the transaction code and class, where the class is displayed as FPP, and the second line with the transaction information.

For remote transactions, the class column is displayed as RMT.

The status displayed can be one of the following:

```
BAL
DYNAMIC
I/O PREV
LOCK
NOTINIT
PSTOP
PUR
QERR
QSTP
SPND
STOP
TRA
USTOP
```

A status of N0INIT appears in the output of a /DISPLAY TRAN command when IMS is unable to initialize a transaction resource during restart processing.

The /DISPLAY TRAN command also displays I/O prevention when a BMP program containing GSAM cannot complete scheduling.

The /DISPLAY TRAN QCNT *count* command returns the global queue counts from shared queues and any affinity of messages on the IMS shared queues.

## Examples

The following are examples of the /DISPLAY TRAN command.

### *Example 1 for /DISPLAY TRAN command*

Entry ET:

```
/DISPLAY TRAN ALL
```

Response ET:

TRAN	CLS	ENQCT	QCT	LCT	PLCT	CP	NP	LP	SEGSZ	SEGNO	PARLM	RC
TPN1	22	14	7	5	0	8	5	8	0	0	0	5
	PSBNAME: DFSCPIC											
TPN2	23	16	0	0	0	0	0	0	0	0	0	0
	PSBNAME: DFSCPIC											
	STATUS: PUR											
TPN2	24	0	0	0	0	0	0	0	0	0	0	0
	PSBNAME: DFSCPIC											
	STATUS: STOP											
TRAN1	4	0	0	10	5	8	8	10	0	0	2	2
	PSBNAME: LU2PGM1											

```

TRAN2      7      0      0 65535 65535 8 8 8      0      0      0 0
  PSBNAME: LU2PGM2
  STATUS: PSTOP
*90332/114253*

```

**Example 2 for /DISPLAY TRAN command**

Entry ET:

```
/DISPLAY TRAN SKS1 SKS2 SKS3 SKS4 SKS5 SKS6 SKS7
```

Response ET:

```

TRAN  CLS  ENQCT  QCT  LCT  PLCT  CP  NP  LP  SEGSZ  SEGNO  PARLM  RC
SKS1   1    0      0 65535 65535 8 8 8      0      0      0 0
  PSBNAME: DFSDDL1
  STATUS: TRA
SKS2   2    0      0 65535 65535 8 8 8      0      0      0 0
  PSBNAME: DFSDDL2
SKS3   3    0      0 65535 65535 8 8 8      0      0      0 0
  PSBNAME: DFSDDL3
  STATUS: STOP,TRA
SKS4   4    0      0 65535 65535 8 8 8      0      0      0 0
  PSBNAME: DFSDDL4
  STATUS: STOP
SKS5   5    0      0 65535 65535 8 8 8      0      0      0 0
  PSBNAME: DFSDDL5
SKS6   6    0      0 65535 65535 8 8 8      0      0      2 0
  PSBNAME: DFSDDL6
  STATUS: BAL( 2)
SKS7   7    0      0 65535 65535 8 8 8      0      0      1 0
  PSBNAME: DFSDDL7
  STATUS: BAL( 2)
*89184/142345*

```

**Example 3 for /DISPLAY TRAN command**

Entry ET:

```
/DISPLAY TRAN TRANA TRANB TRANC QCNT
```

Response ET:

```

TRAN      GBLQCT
TRANA          0
TRANB       1000
TRANC        45
*95200/170817*

```

**Example 4 for /DISPLAY TRAN command**

Entry ET:

```
/DISPLAY TRAN ALL QCNT
```

Response ET:

```

TRAN      GBLQCT  AFFINITY
TRAN1234  1524    IMS1
TRAN1234   3      IMSA
TRAN1234  14
*00305/103034*

```

Explanation: This example command receives multiple lines of output for a transaction, TRAN1234. The output shows there are 1524 messages for the transaction that have an affinity to execute on IMS1. A transaction might have an

affinity if it were an APPC, OTMA, or serial transaction. If an output line shows no affinity, like the example output line with a GBLQCNT of 14, then there could be a single IMS system indicated, or multiple IMS systems that have messages that contribute to the GBLQCNT.

*Example 5 for /DISPLAY TRAN command*

Entry ET:

```
/DISPLAY TRAN ALL QCNT
```

Response ET:

```
DFS000I    TRAN          GBLQCT  AFFINITY
DFS000I    APOL12         1
DFS000I    APOL12         1  IMS1
```

Explanation: This example shows two messages queued to transaction APOL12. One message has affinity to IMS1. The other message does not.

**Related reference:**

“Status and attributes for the /DISPLAY command” on page 523

 [QUERY TRAN command \(Commands\)](#)

## /DISPLAY UOR command

Use the /DISPLAY UOR command to display status information about IMS units of recovery (UORs) for protected resources on the z/OS Resource Recovery Services (RRS) platform.

Subsections:

- “Environment”
- “Syntax”
- “Keywords” on page 758
- “Output fields” on page 758
- “Examples” on page 759

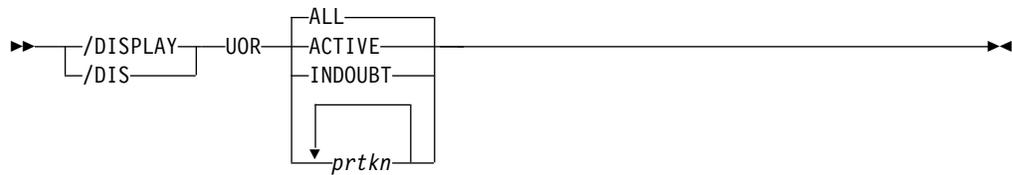
### Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

*Table 318. Valid environments for the /DISPLAY UOR command and keywords*

Command / Keywords	DB/DC	DBCTL	DCCTL
/DISPLAY	X	X	X
ACTIVE	X	X	X
INDOUBT	X	X	
UOR	X	X	

### Syntax



## Keywords

The following keywords are valid for the /DISPLAY UOR command:

### ALL

Displays active and indoubt UORs. ALL is the default.

### ACTIVE

Displays only active UORs.

### INDOUBT

Displays only UORs that were indoubt when they were terminated. Indoubt UORs identify residual work that needs resolution.

### *prtkn*

If you specify the 6-byte pseudorecovery token of the protected UOR, this command displays information about the requested UOR.

## Output fields

The following output fields are displayed by the /DISPLAY UOR command:

**EID** The identifier of a work request across multiple connected distributed systems coordinated by the Encina toolkit. This data appears on a separate line of output if it is available.

### IMS-RECTOKN

The IMS recovery token that uniquely identifies the IMS portion of the UOR.

### LUWID

The identifier of a work request across multiple distributed systems connected by LU 6.2 conversations. This data appears on a separate line of output if it is available.

### PSBNAME

The IMS PSB scheduled for this UOR.

### P-TOKEN

The IMS pseudorecovery token, which can be used in subsequent commands to refer to this particular UOR. This token only displays for indoubt UORs (status RI). Use this token for subsequent /DISPLAY UOR and /CHANGE UOR commands.

### RRS-URID

The unique identifier for a protected unit of recovery (of which IMS work is part). z/OS Resource Recovery Services (RRS) generates and maintains this data.

**ST** Status of the UOR, which can be one of the following:

**A** Active; work is inflight.

**B** Active; work is being backed out.

- C** Active; work is being committed.
  - E** Active; work is in cleanup phase.
  - I** Active; work is indoubt, awaiting phase 2 action.
  - RI** Residual indoubt; indoubt UOR from a prior IMS execution or a dependent-region abend.
- WID** The identifier of a work request across multiple distributed systems that are not associated with a LUWID, EID, or XID task. This data is provided only for the /DIS UOR ACTIVE command, and appears on a separate line of output.
- XID** X/Open identifier (XID). One possible work identifier for a distributed transaction used by a communications resource manager that uses X/Open distributed transaction processing model.

## Examples

### *Example 1 for /DISPLAY UOR command*

Entry ET:

/DISPLAY UOR

Response ET:

ST	P-TOKEN	PSBNAME	RRS-URID	IMS-TOKEN
A		DBOVLFPC	AF3499A27EC3E2980000000401010000	SYS1 0000000400000002
		EID=0001812223768AC1009B5870268346		
C		DBOVLPB	AF1124A27EE1C2380000000501020000	SYS1 0000000700000002
		EID=0002A35549021DA108538121766899670		
RI	00010120	PLAPJK02	123456789012345678901234567890123456789012	SYS1 0000001300000001
		EID=0001C35549021DB176523121445797320		
		*96337/145345*		

### *Example 2 for /DISPLAY UOR command*

Entry ET:

/DISPLAY UOR ACTIVE

Response ET:

ST	P-TOKEN	PSBNAME	RRS-URID	IMS-TOKEN
A		PLAPJK01	AF3278A27EC3E2980000000402010000	SYS1 0000000400000002
		LUWID=0EIMSNET.L62IMS1999E359820810001		
		*96338/091642*		

### *Example 3 for /DISPLAY UOR command*

Entry ET:

/DISPLAY UOR INDOUBT

Response ET:

ST	P-TOKEN	PSBNAME	RRS-URID	IMS-TOKEN
RI	00010040	DBOVLFPC	987688834210978678901234567890123456789012	SYS1 0000008300000001
		LUWID=0EIMSNET.L62IMS1763E358330410001		
RI	00010120	PLAPJK02	AF34A6307EC4E2980000000201010000	SYS1 0000002700000001
		LUWID=0EIMSNET.L62IMS1438E339103810001		
		*96336/113502*		

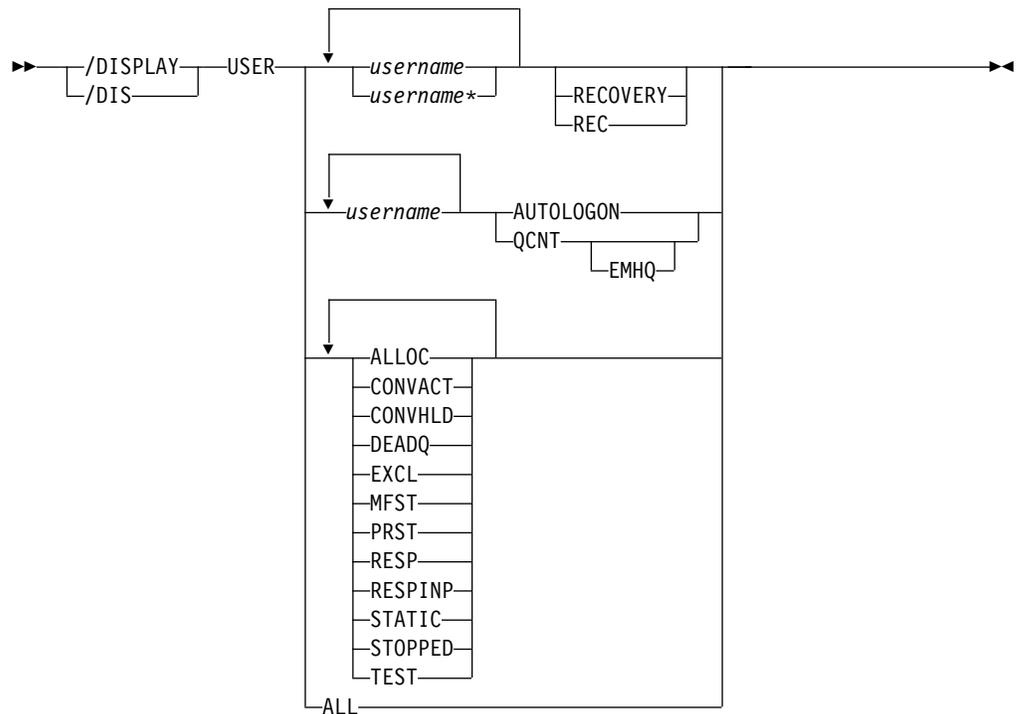
### *Example 4 for /DISPLAY UOR command*



Table 319. Valid environments for the /DISPLAY USER command and keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
/DISPLAY	X	X	X
AUTOLOGON	X		X
EMHQ	X		X
QCNT	X		X
RECOVERY	X	X	
USER	X		X

## Syntax



## Keywords and reserved parameters

The following keywords can be specified for the /DISPLAY USER command:

### AUTOLOGON

Displays the current autologon information for the specified user. The autologon information can be changed dynamically with the /CHANGE command.

### EMHQ

Specifies that global queue count information on the Expedited Message Handler Queues (EMHQ) displays when both EMHQ and QCNT are specified. If EMHQ is specified, QCNT must also be specified. This keyword is valid only in a shared-queues environment.

### QCNT

Specifies that global queue count information is to be displayed. If QCNT is not specified, local queue counts are displayed; when it is specified, no local

queue counts are displayed. If QCNT is specified and EMHQ is not, global queue count information on the shared message queues (MSGQ) is displayed.

This count represents the message counts of messages that can be processed by the IMS system where the command is issued. This count includes messages that can be processed by any IMS system (messages with no affinity), plus messages that can be processed by the IMS system where the command is issued (messages with an affinity to the IMS system that issued the command).

IMS does not check if the specified name is a valid LTERM in the IMS subsystem on which the command is entered.

This command does not display queue counts for Fast Path output messages for the NODE unless EMHQ and QCNT are specified.

This keyword is valid only in a shared-queues environment.

## RECOVERY

The /DISPLAY NODE|USER RECOVERY command can be used to display the recovery values that pertain to the node or user. The recovery values may be displayed from the local control blocks, if available, or from values saved for the node or user in the Resource Manager, if resource information is being kept in Resource Manager, and the node or user is defined to have its status kept in Resource Manager. The recovery information displayed with this command are:

- IMS owner
- The IMS ID (RSEname if an XRF system) of the IMS system that currently owns this resource. If the resource is not currently owned, this field will be displayed as NONE.
- The scope of recovery for a resource, and from where the end-user significant status is maintained and recovered.
- End-user significant status
- The following status are end-user significant status for nodes and users.
  - Conversation
  - STSN
  - Fast Path

These are status that frequently change for a resource, and thus there are performance considerations related to maintaining end-user significant status. The installation can tell IMS how to recover end-user significant status by specifying the level of recovery for each status. Shown for each resource is the level of recovery for each end-user significant status.

### *username*

Specifies either a user structure name or a RACF user ID.

“Attributes” are reserved parameters for the /DISPLAY USER command and cannot be used to name users. Any one attribute can be used with the USER keyword. Users are selected for display based on the attribute or attributes specified. The attribute display format is the same as the standard USER display. Attributes usually correspond to the conditions displayed for users.

Following is a list of the attributes that can be specified with the USER keyword:

- ALLOC
- CONVACT
- CONVHLD

- DEADQ
- EXCL
- MFST
- PRST
- RESP
- RESPINP
- STATIC
- STOPPED
- TEST

RESPINP shows users with RESP-INP and RESP-INP-FP status.

## Usage notes

The USER parameter can be generic when the generic parameter specifies users that already exist.

The following information is displayed:

- The USER field shows the user structure name. N/A appears for all user IDs signed on to static nodes.
- The USERID field shows the RACF user ID that was used to sign on to the node. N/A appears if a user structure is not associated with a node. In that case, the user structure exists to hold status or has messages inserted to it. If no messages or status exist, the user structure is deleted at the next checkpoint.
- The node name appears if the user is allocated to a node or signed on to a node.
- The cumulative queue counts for enqueues, dequeues, and number of items still on the queue for the specified users is displayed. The cumulative counts include all LTERMs associated with the specified users, whether the LTERMs are allocated to a node or not.

For non-ISC static users, queue counts are displayed as N/A, since non-ISC static users are not associated with queues.

- STOPPED or ALLOC is displayed to show whether the user LTERMs are assigned to a node (ALLOC) or the user is stopped (STO).

User IDs can only be used as parameters with the /DISPLAY USER command and the /SIGN ON command. In other commands that support the USER keyword, the user structure name, not the user ID, must be used.

The ALLOC status is followed by the node name (in parentheses) to which the user is allocated or signed on to.

The use of DEADQ with the /DISPLAY USER command shows all users that have queues marked as dead letter queues. When all the messages are purged by the /DEQUEUE command, the DEADQ status is removed. After all the messages have been purged and the status removed, then the user structure is deleted at the next simple checkpoint.

In a shared-queues environment, use the QCNT keyword to obtain the current message queue counts on the shared queues structure. The local queue counts (ENQCT, DEQCT, and QCT) value that is returned on this command when the QCNT keyword is not specified represents the messages being processed by the

IMS system where this command is issued. Do not use local queue counts for queue evaluation purposes.

## Equivalent IMS type-2 commands

The following table shows variations of the /DISPLAY USER command and the IMS type-2 commands that perform similar functions.

Table 320. Type-2 equivalents for the /DISPLAY USER command.

Task	/DISPLAY USER command	Similar IMS type-2 command
Displays preset destination.	/DISPLAY USER <i>user</i>	QUERY USER SHOW(PRESET)
Displays node.	/DISPLAY USER <i>user</i>	QUERY USER SHOW(NODE)
		QUERY USERID SHOW(NODE)
Displays status for a particular user.	/DISPLAY USER <i>user</i>	QUERY USER SHOW(STATUS)
Displays user ID.	/DISPLAY USER <i>user</i>	QUERY USER SHOW(USERID)
Displays status of a particular user ID.	/DISPLAY USER <i>user</i>	QUERY USERID SHOW(STATUS)
Displays user.	/DISPLAY USER <i>user</i>	QUERY USERID SHOW(USER)
Displays autologon parameters.	/DISPLAY USER <i>user</i> AUTOLOGON	QUERY USER SHOW(AUTOLOGON)
Displays queue count.	/DISPLAY USER <i>user</i>	QUERY USER SHOW(QCNT)
	/DISPLAY USER <i>user</i> QCNT	
Displays message queue count in the Expedited Message Handler (EMH) queues.	/DISPLAY USER <i>user</i> QCNT EMHQ	QUERY USER SHOW(EMHQ)
Displays status recovery information.	/DISPLAY USER <i>user</i> RECOVERY	QUERY USER SHOW(RECOVERY)
Displays owner IMSID in RM resource structure.	/DISPLAY USER <i>user</i> RECOVERY	QUERY USER SHOW(OWNER)

## Examples

The following are examples of the /DISPLAY USER command.

### Example 1 for /DISPLAY USER command

Entry ET:

```
/DISPLAY USER IMSUS01 IMSUS02
```

Response ET:

```
USER  USERID  ENQCT DEQCT  QCT
IMSUS01  IMSUS01    1     1     0 ALLOC(DTSLU201) CONV-HLD
N/A     IMSUS01    N/A   N/A   N/A ALLOC(L3270A ) STATIC
N/A     IMSUS02    N/A   N/A   N/A ALLOC(L3270B ) STATIC
DT327002 IMSUS02     0     0     0 ALLOC(DT327002)
*92350/112229*
```

Entry ET:

```
/DISPLAY USER USER1 USER2 USER3 QCNT
```

Response ET:

USERID	USER	GBLQCT
N/A	USER1	25
USER2	USER2A	0
USER2	USER2B	10
USER3	IMSUSR3A	40

\*95200/170817\*

Entry ET:

/DISPLAY USER IMSUS01\*

Response ET:

USER	USERID	ENQCT	DEQCT	QCT
IMSUS01C	N/A	0	0	0 STOPPED
IMSUS01D	N/A	0	0	0 STOPPED
IMSUS01	IMSUS01	1	1	0 ALLOC(DTSLU201) CONV-HLD
N/A	IMSUS01	N/A	N/A	N/A ALLOC(L3270A ) STATIC

\*92350/113904\*

Entry ET:

/DISPLAY USER ALL

Response ET:

USER	USERID	ENQCT	DEQCT	QCT
CA13	N/A	0	0	0 STATIC
IMSUS01C	N/A	0	0	0 STOPPED
IMSUS01D	N/A	0	0	0 STOPPED
LU6SPA	N/A	0	0	0 ALLOC(LU6NDPF ) STATIC
LU6SPB	N/A	0	0	0 STATIC
DT327002	IMSUS02	0	0	0 ALLOC(DT327002)
IMSUS03	IMSUS03	0	0	0 ALLOC(DTSLU601)
IMSUS01	IMSUS01	1	1	0 ALLOC(DTSLU201) CONV-HLD
IMSUS05	N/A	0	0	0 STOPPED
N/A	IMSUS02	N/A	N/A	N/A ALLOC(L3270B ) STATIC
N/A	IMSUS01	N/A	N/A	N/A ALLOC(L3270A ) STATIC
N/A	IMSUS04	N/A	N/A	N/A ALLOC(L3270C ) STATIC

\*92350/114845\*

*Example 2 for /DISPLAY USER command*

Entry ET:

/DISPLAY USER ABC100 EMHQ QCNT

One user is on the queue.

Response ET:

```

14.42.46          57/DIS USER ABC100 QCNT EMHQ.
14.42.46          IEE600I REPLY TO 57 IS;/DIS USER ABC100 QCNT EMHQ.
14.42.46 JOB00110 DFS000I  USER          GBLQCT  IMS1
14.42.46 JOB00110 DFS000I  ABC100          0      IMS1
14.42.46 JOB00110 DFS000I  *98203/144246* IMS1

```

*Example 3 for /DISPLAY USER command*

Entry ET:

/DISPLAY USER IMSUS01 IMSUS02 IMSUS03 AUTOLOGON.

Response ET:

USER	NODE	MODE	DESC	ID
IMSUS01	N/A	N/A	N/A	N/A
IMSUS02	N/A	N/A	N/A	N/A
IMSUS03	N/A	N/A	N/A	N/A

\*99243/082505\*

Entry ET:

/CHANGE USER IMSUS01 AUTOLOGON DT327001 MODE LU032NT4.

Response ET:

DFS058I CHANGE COMMAND COMPLETED

Entry ET:

/CHANGE USER IMSUS02 AUTOLOGON DTSLU601 MODE LU6NEGPS ID IMSID2.

Response ET:

DFS058I CHANGE COMMAND COMPLETED

Entry ET:

/CHANGE USER IMSUS03 AUTOLOGON DT327001 LOGOND DFS3270 MODE LU032NT4.

Response ET:

DFS058I CHANGE COMMAND COMPLETED

Entry ET:

/DISPLAY USER IMSUS01 IMSUS02 IMSUS03 AUTOLOGON.

Response ET:

USER	NODE	MODE	DESC	ID
IMSUS01	DT327001	LU032NT4	N/A	N/A
IMSUS02	DTSLU601	LU6NEGPS	N/A	IMSID2
IMSUS03	DT327001	LU032NT4	DFS3270	N/A

\*99243/083309\*

***Example 4 for /DISPLAY USER RECOVERY command***

Entry ET:

/DISPLAY USER IMSUS\* IMSUS03 LU6SPG RECOVERY

Response ET:

USER	OWNER	SRM	CONV	STSN	FPATH
IMSUS05	IMSA	GLOBAL	Y	Y	Y
IMSUS01	IMSA	LOCAL	Y	Y	Y
IMSUS03	IMSB	GLOBAL	Y	N	N
LU6SPG	IMSA	LOCAL	Y	Y	Y

\*99240/134730\*

**Related reference:**

“Status and attributes for the /DISPLAY command” on page 523

 QUERY USER command (Commands)

 QUERY USERID command (Commands)



/END NODE is not valid for ISC nodes with users or nodes that were dynamically created. /END NODE USER only applies to ISC sessions and affects the half-session allocated to username.

**Restrictions for using NODE and USER parameters together:**

- Commands with the NODE USER keyword pair are valid only if:
  - The USER is signed on to the NODE
  - In an ISC environment, the USER is allocated to the NODE
  - The nodes and users already exist
- /END NODE USER commands are valid for ISC and non-ISC nodes and users.

**USER**

Without the NODE keyword, USER specifies the dynamic user for which special operating modes are terminated. The named user must exist in IMS, must be dynamic, and must not be signed on to a node. When /END USER is issued against a temporary user, which was created solely to retain status that is now reset, the temporary user is deleted at the next simple checkpoint.

**Usage notes**

The /END NODE command applies to dynamic nodes in addition to static nodes because MFSTEST mode is associated with dynamic nodes as well as dynamic users. The /END NODE command and the /END NODE USER command reset MFSTEST mode at the node level. The /END USER command resets MFSTEST at the user level. The /END command with no keywords resets MFSTEST at the node level for static terminals (they have no user level) and at the user level for dynamic terminals. The /END NODE USER command supports static and dynamic ISC sessions. When the /END NODE command or the /END NODE USER command is used for a dynamic node, only MFSTEST is reset. Exclusive mode can only be reset for a dynamic resource with the /END USER command. If global resource information is kept in Resource Manager, MFSTEST mode is reset globally and locally. If global resource information is not kept in Resource Manager, MFSTEST mode is reset locally.

The /END command with no keywords operates on the entering terminal with a signed on user. The /END command with no keywords is not supported through the OM API in an IMSplex.

This command can be issued to an IMSplex using the Batch SPOC utility.

**Example**

Entry ET:

```
/END
```

Response ET:

```
DFS058I  END COMMAND COMPLETED
```

Explanation: All modes that previously existed are ended.

---

## Chapter 28. /ERESTART command

Use the /ERESTART command, which is a multisegment command, to perform a manual restart of an IMS XRF alternate system and to restart IMS any time the system was not terminated with an orderly /CHECKPOINT shutdown.

Three conditions that result in the need for an emergency restart are:

- Abnormal termination of IMS
- Abnormal termination of z/OS
- Forced termination of IMS by using the z/OS MODIFY command

Subsections:

- “Environment”
- “Syntax” on page 770
- “Keywords” on page 771
- “Usage notes” on page 776
- “Examples” on page 777

### Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

Table 322. Valid environments for the /ERESTART command and keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
/ERESTART	X	X	X
BACKUP	X		X
BUILDQ	X		X
CHECKPOINT	X	X	X
CMDAUTH	X		X
CMDAUTHE	X		X
COLDBASE	X	X	
COLDCOMM	X		X
COLDSYS	X	X	X
FORMAT	X	X	X
MULTSIGN	X		X
NOBMP	X	X	X
NOCMDAUTH	X		X
NOCMDAUTHE	X		X
NOTRANAUTH	X		X
NOUSER	X		X
OPTION	X	X	X
OVERRIDE	X	X	X
SNGLSIGN	X		X

Table 322. Valid environments for the /ERESTART command and keywords (continued)

Command / Keywords	DB/DC	DBCTL	DCCTL
TRANAUTH	X		X
USER	X		X

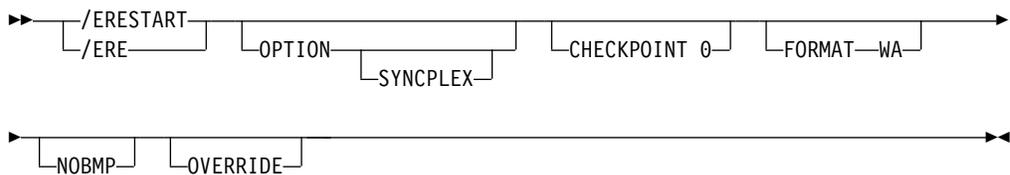
## Syntax

### Manual restart of an XRF alternate system

Use this command only after the active system issues message DFS3804I.

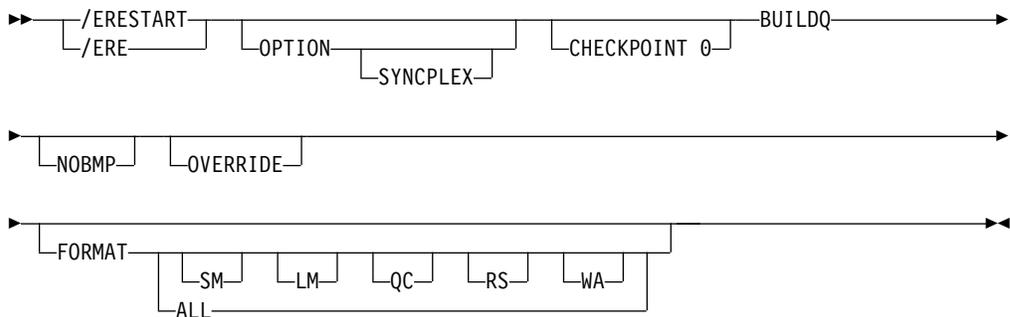


### Restart of IMS following loss of virtual storage only

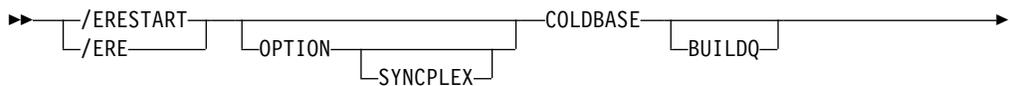


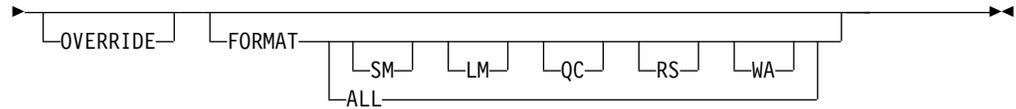
### Restart of IMS following loss of virtual storage and message queue data set integrity

The message queues have not been dumped to the system log since the most recent cold start.

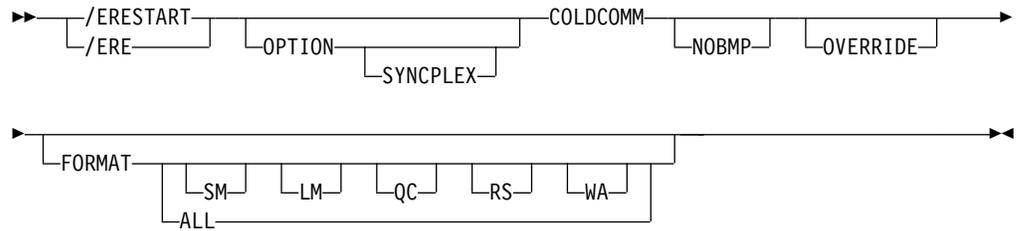


### Restart of IMS following /ERESTART failure of the database component

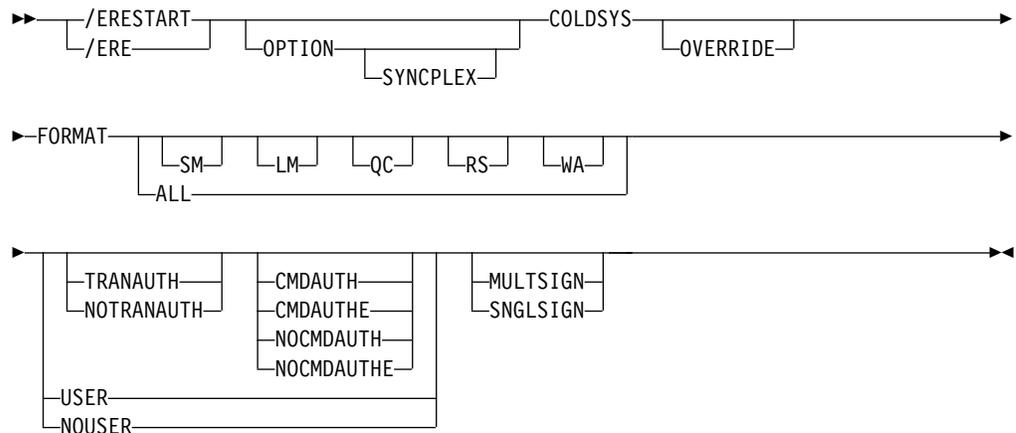




**Restart of IMS following /ERESTART failure of communication component**



**Restart of IMS following /ERESTART failure of both the database and communication components**



**Keywords**

The following keywords are valid for the /ERESTART command:

**BACKUP**

Indicates that the control region is an alternate system in an XRF environment. This command is required when performing manual restart.

**BUILDQ**

Is required for any restart from a failure in which message queue data set integrity was lost. When a checkpoint other than checkpoint 0 is specified, the checkpoint must be one at which the message queues were dumped to the system log. Message queue data set integrity has been lost if one or more of the data sets have been reallocated or reformatted. If a checkpoint is not specified, IMS selects the appropriate checkpoint from which to start.

If an /ERESTART BUILDQ command fails, and then you issue the /ERESTART CHECKPOINT 0, /ERESTART COLDCOMM, or /ERESTART COLDSYS command, messages in local queues are lost. However, IMS Message Requirer (MRQ) can be used to recover local message queues.

In a shared-queues environment, the BUILDQ keyword is ignored because the message queue data sets are not used.

#### **CHECKPOINT**

CHECKPOINT is an optional keyword that identifies whether the restart should use the latest cold start system checkpoint that was taken by the IMS subsystem. If you do not specify this keyword, IMS determines the correct system checkpoint to use. If you do specify this keyword, you can specify only CHECKPOINT 0 to use the latest cold start system checkpoint.

#### **CMDAUTH**

Specifies that both signon (user identification verification) and command authorization for static and ETO terminals are in effect at the end of the emergency restart. (Command authorization is same as specifying RCF=S on the startup parameter.)

To specify CMDAUTH, you must specify RCF=A|Y|T|C|S as an initialization EXEC parameter.

#### **CMDAUTHE**

Specifies that command authorization for ETO terminals (same as RCF=S on the startup parameter) is in effect at the end of the emergency restart. CMDAUTHE also resets command authorization for static terminals, if it was set.

To specify CMDAUTHE, you must specify RCF=A|Y|T|C|S as an initialization EXEC parameter.

#### **COLDBASE**

Indicates a cold start of the database component, while performing an emergency restart of the communications component.

- If this keyword is used, you are responsible for the recovery of the databases. The Fast Path areas will not be redone and no backouts of inflight DL/I databases will be performed. If in-doubts exist, a batch backout run with the cold start option will backout inflight DL/I data. This will place both DL/I and Fast Path data in the aborted state.
- If this keyword is not used, the database component will be warm started.

If IMS is enabled to use the repository, IMS calls Resource Manager (RM) at the end of COLDBASE processing to read the change list, if one exists, for the IMS. The IMS change list is maintained by RM and is built if the IMPORT command with the SCOPE(ALL) keyword is issued while the IMS system is down. IMS imports the stored resource definitions from the repository for the resources and descriptors in the change list. The resources and descriptors in the change list are quiesced and not available for use until the stored resource definitions are imported from the repository. The change list for the IMS is deleted at the end of COLDBASE processing.

#### **COLDCOMM**

Indicates a cold start of the data communication component, while an emergency restart of the database component is being performed. This includes recovering Fast Path DEDBs, reloading MSDBs, backing out inflight changes to DL/I databases, and maintaining all existing indoubt data. COLDCOMM is used to get the DC network reinstated as soon as possible while databases are being recovered. As a result of COLDCOMM, all of the messages on the local message queue will be lost; inflight messages in shared queues are moved to the cold queue for later recovery. If this keyword is not used, the data communication component will be restarted.

If IMS is enabled to use the repository, IMS calls RM at the end of COLDCOMM processing to read the change list, if one exists, for the IMS. The IMS change list is maintained by RM and is built if the IMPORT command with the SCOPE(ALL) keyword is issued while the IMS is down. IMS imports the stored resource definitions from the repository for the resources and descriptors in the change list. The resources and descriptors in the change list are quiesced and not available for use until the stored resource definitions are imported from the repository. The change list for the IMS is deleted at the end of COLDCOMM processing.

### **COLDSYS**

Indicates a cold start of both the database and the data communication components. An /ERE COLDSYS command differs from a /NRE CHECKPOINT 0 command in function, but the COLDSYS keyword is equivalent to, and replaces, the /NRE CHECKPOINT 0 DETACH command. The COLDSYS keyword is the newer syntax.

The use of the COLDSYS keyword causes the release of database authorizations for DBCTL SHARECTL (full DBRC data sharing) and releases any locks held by the subsystem in IRLM. This is the required form of cold start following a failure when using DBRC SHARECTL.

The SGN=, TRN=, and RCF= startup parameters can be overridden by the /ERESTART COLDSYS command using the keywords shown in the following table. This table lists the keywords and the startup parameters that those keywords can override and a brief description.

*Table 323. Security keywords and their startup parameter equivalents*

<b>Keyword <sup>1</sup></b>	<b>Description</b>	<b>Startup parameter</b>
CMDAUTH	RACF command authorization on static and ETO terminals only	RCF=S
CMDAUTHE	RACF command authorization on ETO terminals only	RCF=C
MULTSIGN	Permits multiple signons for each user ID	SGN=M
NOCMDAUTH	Resets the command authorization on static and ETO terminals	Not RCF=S
NOCMDAUTHE	Resets the command authorization on ETO terminals only	Not RCF=C
NOTRANAUTH	Resets the transaction authorization.	Not TRN=F or Y
NOUSER	Resets user identification verification, transaction authorization, and command authorization	Not SGN=F or Y (G or Z becomes M) Not TRN=F or Y Not RCF=C or S
SNGLSIGN	Permits a single signon for each user ID	SGN=F and Y Not SGN=M (G or Z becomes F or Y)
TRANAUTH	Transaction authorization	TRN=F or Y
USER	Sets user identification verification	SGN=Y

**Note:**

1. Valid only with the /ERESTART COLDSYS command.

If AUTOIMPORT=REPO or AUTOIMPORT=AUTO is specified and IMS is enabled to use the repository, the stored resource definitions are read from the repository during COLDSYS processing. The IMS change list is not processed during /ERE COLDSYS processing because all the resource definitions are imported as part of COLDSYS processing. If one exists, the change list in RM for the IMS being restarted with COLDSYS is deleted.

#### FORMAT

Specifies which queues or data sets should be formatted as part of the restart process when:

- A message queue or data set I/O error occurs.
- The size of a message queue or data set is to be changed.
- A message queue or data set is to be reallocated.
- Allocation of the WADS changes.

When queues or data sets must be formatted as part of the restart process, specify one or more of the following:

- SM** Short-message queue
- LM** Long-message queue
- QC** Control record data set (QBLKs)
- RS** Restart data set
- WA** Write ahead data set
- MD** MSDB dump data set (valid only for backup)
- ALL** All message queues (SM and LM) and data sets (QC, WA, and RS). For DBCTL, this supports RS and WA only.

Any combination of SM, LM, QC, WA, MD, and RS can be specified; for example, FORMAT LM RS. When you specify ALL, do not specify SM, LM, QC, WA, MD, or RS.

In a shared-queues environment, the LM, SM, and QC parameters are ignored because the message queue data sets are not used. If you specify ALL, IMS does not attempt to format the message queue data sets.

The following table shows the environments in which the parameters are valid.

Table 324. /ERESTART FORMAT command parameter environments

Parameter	DB/DC	DBCTL	DCCTL
SM	X		X
LM	X		X
QC	X		X
RS	X	X	X
WA	X	X	X
MD	X		X
ALL	X	X	X

If the WADS must be reformatted during emergency restart, the ALL parameter can be used. IMS first closes the OLDS from the WADS and then reformats the WADS. If you generally use the ALL parameter during restart, do not be concerned that the WADS will not be available to close the OLDS.

**MULTSIGN**

Permits multiple signons for each user ID.

In an IMSplex with Resource Manager and a resource structure, if MULTSIGN conflicts with the single user signon definition for the IMSplex, a warning message is issued.

**NOBMP**

Specifies no backout of BMP updates occurs and all affected databases and programs are stopped.

If NOBMP is not specified, all updates made subsequent to the last commit point invoked by the active BMP programs are backed out of the database as part of the restart process.

NOBMP is ignored if no BMP programs were active at the time of the system failure.

**NOCMDAUTH**

Resets command authorization on static and ETO terminals.

**NOCMDAUTE**

Resets command authorization for static and ETO terminals. The command authorization is reset for static terminals because the command authorization for static terminals cannot exist without the command authorization for ETO terminals.

**NOTRANAUTH**

Turns off transaction authorization. NOTRANAUTH is not the opposite of TRANAUTH. TRANAUTH sets transaction authorization and also turns on signon (user identification verification).

If you specify NOTRANAUTH, it will be rejected with an error message if TRN=F was specified as a JCL EXEC parameter.

**NOUSER**

Specifies that none of the following is in effect at the end of the emergency restart:

- Transaction authorization
- User identification verification
- Command authorization

**OPTION**

Specifies parameters that are in effect for the execution of the restart process.

The valid parameters for this keyword are as follows:

**SYNCPLEX**

Synchronize the startup of IMS systems after a multi-IMS failure. The option SYNCPLEX is entered on all the IMS systems that are to be synchronized. At the end of the restart process, the write to operator with replay (WTOR) message, DFS3067A, is issued. Once all of the IMS Systems are at this WTOR, they are now synchronized and can resume normal processing. The operator replies to the WTOR on each IMS and normal processing resumes

**OVERRIDE**

Is required only to restart the system after failure of power, machine, z/OS, or DBRC where IMS abnormal termination was unable to mark the DBRC subsystem record in RECON as abnormally terminated. IMS emergency restart will abort with message DFS0618A when DBRC indicates that the subsystem is currently active and that neither the OVERRIDE keyword nor the BACKUP

keyword is present on the /ERESTART command. If there is any doubt about the status of an IMS system at restart time, the OVERRIDE keyword should not be used.

**Attention:** Use of the OVERRIDE keyword on a currently running IMS system can lead to database and system integrity problems.

#### **SNGLSIGN**

Permits a single signon for each user ID.

In an IMSplex with Resource Manager and a resource structure, if SNGLSIGN conflicts with the single user signon definition for the IMSplex, a warning message is issued.

#### **TRANAUTH**

Specifies both transaction authorization and user identification verification, with or without RACF.

To specify TRANAUTH, you must specify RCF=A|Y|T|C|S as an initialization EXEC parameter.

#### **USER**

Specifies user identification verification. User identification verification means that signon is required by static terminals. This keyword has no effect on ETO terminals because they are always required to sign on. User identification verification can be forced on by TRANAUTH or CMDAUTH.

To specify USER, you must specify RCF=A|Y|T|C|S as an initialization EXEC parameter.

### **Usage notes**

This command can be issued to an IMSplex by using the Batch SPOC utility.

For an /ERESTART command restart, the base security definition is created:

- From the IMS system definition and the EXECUTE parameter specifications, when the COLDSYS keyword is specified.
- From the IMS checkpoint data, when the COLDSYS keyword is not specified.

To override the security definitions, specify the COLDSYS keyword and the security keywords on the COLDSYS keyword.

The format for /ERESTART depends on whether the IMS system is an XRF alternate system. If the system is not an alternate system, the format also depends on the type of system failure that is associated with the termination and whether the message queues have been dumped to the system log since the most recent cold start.

All /ERESTART formats require an EOM indication to denote end-of-message; an EOS indication must be included for all segments that precede the last segment.

With Fast Path, the /ERESTART command loads the latest MSDB checkpoint data set and applies all MSDB changes up to the latest complete commit point. /ERESTART also ensures that DEDB updates created between the restart checkpoint and the latest complete commit point are applied. If the DEDB updates are not applied, they are queued for output. Output messages not acknowledged or not transmitted are also queued for output.

When IMS is initialized, the system parameters that are used for this initialization can come from the IMS system generation, from a PROCLIB member, or from EXEC statements that can override both the defaults and the PROCLIB members. Therefore, message DFS1929I is displayed showing the system parameters used for this particular initialization. The system parameters are also written to the job log.

When the /ERESTART command is issued without any keywords or with the COLDBASE or COLDCOMM keyword, IMS processes the system log during emergency restart processing to build runtime resource definitions from the log. The stored resource definitions from the RDDS or the IMSRSC repository are not used during emergency restart processing to build runtime resource definitions.

If IMS is enabled to use the repository, IMS calls Resource Manager (RM) at the end of emergency restart to read the change list for the IMS. The IMS change list is maintained by RM and is built if the IMPORT command with the SCOPE(ALL) keyword is issued while the IMS is down.

If a change list exists for the IMS that is being restarted, the database, program, transaction, and routing code resources and descriptors in the IMS change list and that apply to the IMS environment are quiesced and are not available for use until the stored resource definitions are imported from the repository.

After the IMS log is processed, IMS imports the stored resource definitions from the repository for the database, program, transaction, and routing code resources and descriptors in the IMS change list. It then applies the changes to the runtime resource and descriptor definitions in the IMS.

For the resources or descriptors that are in the IMS change list and that do not exist in IMS, the runtime resource definitions are created from the stored resource definitions in the repository. For the resource or descriptors that exist in IMS, the runtime resource definitions are updated with the stored resource definitions from the repository. The change list for the IMS is deleted at the end of the emergency restart.

## Examples

The following are examples of the /ERESTART command:

### *Example 1 for /ERESTART command*

All system data sets are intact.

Entry ET:

```
/ERESTART.
```

Response ET:

```
DFS058I (time stamp) ERESTART COMMAND IN PROGRESS  
DFS680I USING CHKPT 85200/123456  
DFS994I *CHKPT 85201/110117**SIMPLE*
```

Explanation: The restart is being performed from checkpoint 85200/123456, which was selected by IMS. If IMS contained active BMPs when the system failed, checkpoint 85200/123456 is the checkpoint that allows backout of all updates made by each active BMP since its latest system checkpoint. A simple checkpoint is written on the system log. The checkpoint number is 85201/110117.

### *Example 2 for /ERESTART command*

Message queues are in error.

Entry ET:

```
/ERESTART BUILDQ.
```

Response ET:

```
DFS058I (time stamp) ERESTART COMMAND IN PROGRESS  
DFS680I USING CHKPT 85119/074811  
DFS994I *CHKPT *85120/192021**SIMPLE*
```

Explanation: IMS is restarted at 192021 (time) on 85120 (\* \* date) from the most recent DUMPQ or SNAPQ checkpoint, 85119/074811. A simple checkpoint is written on the system log. The checkpoint number is 85120/192021.

The message queues from checkpoint 85119/074811 are loaded.

If BMP programs were active when the system failed, all updates invoked by the BMP programs are backed out.

### *Example 3 for /ERESTART command*

Small and large message queues were reallocated. The message queues have been dumped at least once since the most recent cold start.

Entry ET:

```
/ERESTART BUILDQ FORMAT SM LM.
```

Response ET:

```
DFS058I (time stamp) ERESTART COMMAND IN PROGRESS  
DFS680I USING CHKPT 85201/070348  
DFS994I *CHKPT *85204/010203**SIMPLE*
```

Explanation: IMS is restarted at 010203 (time) on 85204 (Julian date). A simple checkpoint was written on the system log. The checkpoint number is 85204/010203.

The large and small message queue data sets are reformatted.

The message queues are loaded from checkpoint 85201/070348.

If BMP programs were active when the system failed, all updates invoked by the BMP programs are backed out.

### *Example 4 for /ERESTART command*

Message queues in error or data sets reallocated and the message queues have not been dumped since the most recent cold start.

Entry ET:

```
/ERESTART BUILDQ FORMAT ALL NOBMP.
```

Response ET:



```

02/05/15 14:26:40 RSENAME: DFSRSENM ACTIVE IMSA
DFS3499I ACTIVE DDNAMES: MODBLKSA IMSACBA FORMATA MODSTAT ID: 11
DFS3804I LAST CHKPT ID VALID FOR RESTART: 85135/141619-BUILDQ: 85135/141619

DFS994I COLD START COMPLETED.
DFS3830 14:16:23 NO DEFERRED PAGE FIXING WAS DONE
DFS3856I REQSYNC WAS PROCESSED TO TAKE SNAPQ CHKPT.
DFS994I *CHKPT 85135/14269***SNAPQ**
DFS3499I ACTIVE DDNAMES: MODBLKSA IMSACBA FORMATA MODSTAT ID: 11
-----

MASTER MESSAGE WAITING PASSWORD:

```

*Figure 5. IMS formatted master for the active system*

**Related concepts:**

-  Synchronizing IMS restart after a multiple IMS system failure (System Administration)
-  Shutting down an IMS system that uses dynamic resource definition (Operations and Automation)

**Related reference:**

- “Commands recovered during emergency restart” on page 29
- “How to enter IMS commands” on page 2
- Chapter 13, “/CHECKPOINT command,” on page 153

---

## Chapter 29. /EXCLUSIVE command

Use the /EXCLUSIVE command to put a terminal into exclusive mode. This command, without keywords, is valid from a terminal with a signed on user.

The LINE PTERM and NODE keywords are only valid for output-only terminals. For a dynamically created user, the exclusive mode status is remembered across signons.

Subsections:

- "Environment"
- "Syntax"
- "Keywords"
- "Usage notes" on page 782
- "Examples" on page 782

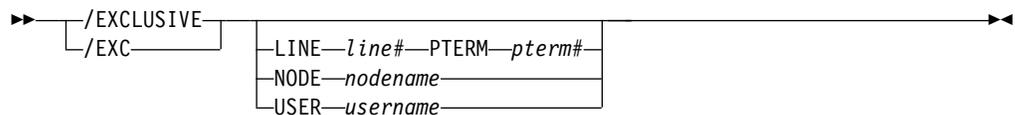
### Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

Table 325. Valid environments for the /EXCLUSIVE command and keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
/EXCLUSIVE	X		X
LINE	X		X
NODE	X		X
PTERM	X		X
USER	X		X

### Syntax



### Keywords

The following keywords are valid for the /EXCLUSIVE command:

#### LINE PTERM

Specifies the communication line to be put into exclusive mode. The /DISPLAY LINE line# PTERM pterm# command identifies a terminal in exclusive mode.

#### NODE

Specifies that the static VTAM node be put into exclusive mode. This command is not valid for nodes that were dynamically created.

#### USER

Specifies that the dynamic user be put into exclusive mode. If the user does

not exist in IMS, it and its associated LTERMs will be created to remember the exclusive status. If /EXCLUSIVE USER is issued against an existing user, the user must be dynamic.

## Usage notes

Exclusive mode restricts the output received by the terminal affected:

- A regular input/output terminal in exclusive mode only receives messages transmitted in response to transactions entered from that physical terminal.
- An output-only terminal in exclusive mode receives messages transmitted in response to transactions entered into its associated input logical terminals.

Except for an IMS system message responding to an error condition, all other output for a terminal in exclusive mode remains queued for later transmission. The queued output is transmitted after exclusive mode is reset with the /END command or a /START command.

In an IMSplex, if global resource information is kept in Resource Manager, the /EXCLUSIVE command sets a global exclusive status for the resource. If global resource information is not kept in Resource Manager, the /EXCLUSIVE command sets the status.

This command can be issued to an IMSplex using the Batch SPOC utility.

The /END command is used to get a terminal out of exclusive mode.

## Examples

The following are examples of the /EXCLUSIVE command:

### *Example 1 for /EXCLUSIVE command*

Entry ET:

```
/EXCLUSIVE
```

Response ET:

```
DFS058I EXCLUSIVE COMMAND COMPLETED
```

Explanation: No messages are sent to this terminal other than application program output resulting directly from transaction input.

### *Example 2 for /EXCLUSIVE command*

Entry ET:

```
/EXCLUSIVE USER alpha
```

Response ET:

```
DFS058I EXCLUSIVE COMMAND COMPLETED
```

## Chapter 30. /EXIT command

Use the /EXIT command to end conversations between users and application programs.

The conversation can be either active or in hold status at the time /EXIT is entered. Depending upon the conversational processing options selected for the IMS system, the application program might or might not be notified of the conversation termination.

Subsections:

- “Environment”
- “Syntax”
- “Keywords”
- “Usage notes” on page 784
- “Examples” on page 785

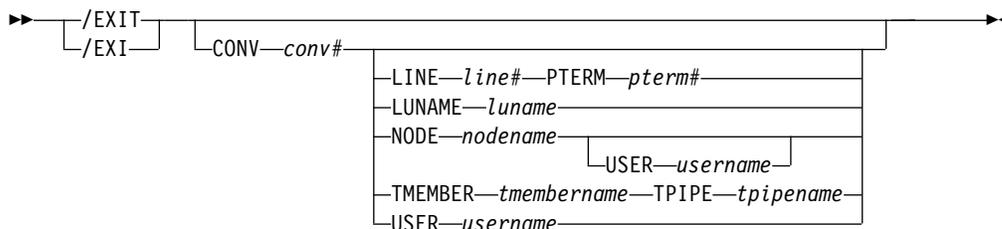
### Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

Table 326. Valid environments for the /EXIT command and keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
/EXIT	X		X
CONVERSATION	X		X
LINE	X		X
LUNAME	X		X
NODE	X		X
PTERM	X		X
TMEMBER TPIPE	X		X
USER	X		X

### Syntax



### Keywords

The following keywords are valid for the /EXIT command:

### **CONVERSATION**

Terminates the specified conversation. The conversation is terminated whether it was active or held. The conversation identifier (conv#) must be specified as a 4-digit number, including leading zeros.

### **LINE PTERM**

Terminates the conversation on the physical terminal specified, regardless of whether the conversation was active or held. The line specified must be in stopped and idle status.

### **LUNAME**

Terminates the IMS conversation and the APPC conversation on the specified luname. In an IMSplex environment, /EXIT LUNAME needs to be issued on the IMS that owns the conversation. Within an IMS system, the conversation ID is unique.

### **NODE**

Terminates the conversation on the node specified, regardless of whether the conversation was active or held. The node specified must be stopped or idle. This form of the command does not support dynamic nodes or LU 6.2 nodes defined with users.

The NODE USER combination applies only to the specified half-session of the ISC parallel session node allocated to USER. The half-session must have been stopped and idled. The command must be issued from a terminal or node other than the one in the conversation.

#### **Restrictions for using NODE and USER parameters together:**

- Commands with the NODE USER keyword pair are valid only if:
  - The USER is signed on to the NODE
  - In an ISC environment, the USER is allocated to the NODE
  - The nodes and users already exist
- /EXIT (CONV) NODE USER commands are valid for ISC, LUP, and 3600 nodes only.

### **TMEMBER TPIPE**

Terminates the IMS conversation on the specified *tmembername tpipe*. In an IMSplex environment, /EXIT TMEMBER TPIPE must be issued on the IMS that owns the conversation. Within an IMS system, the conversation ID is unique.

### **USER**

Terminates the conversation associated with the specified user. The user must not be signed on to a node. The user must be dynamic.

## **Usage notes**

Transactions that are queued for processing when the /EXIT command is issued are discarded. If a transaction is being processed or was sent to a remote system for processing when /EXIT is issued, the command is still processed; however, the conversation will be terminated, but the conversational transaction will still be issued. When the conversational response is received, the Conversational Abnormal Termination exit (DFSCONE0) is called, and the response message is queued to the response terminal.

The /EXIT command with no keywords can only be used if the conversation is active.

If global resource information is kept in Resource Manager (RM), /EXIT NODE or /EXIT USER terminates the conversation in RM. If global resource information is not kept in RM, /EXIT NODE or /EXIT USER terminates a conversation on the IMS where the /EXIT command is entered or routed to by the OM SPOC.

This command can be issued to an IMSplex using the Batch SPOC utility.

/EXIT is not valid from an LU 6.2 device or OTMA device. DEALLOCATE the APPC conversation after receiving all output from an iteration. When the /EXIT command specifies only the CONVERSATION keyword, the command can be entered only from the terminal that owns the conversation.

## Examples

The following are examples of the /EXIT command:

### *Example 1 for /EXIT command*

Entry ET:

```
/EXIT
```

Response ET:

```
DFS058I  EXIT COMMAND COMPLETED
```

Explanation: The active and not held conversation associated with the entering terminal is terminated.

### *Example 2 for /EXIT command*

Entry ET:

```
/EXIT
```

Response ET:

```
DFS576I  EXIT COMPLETED, TRANSACTION DISCARDED
```

Explanation: /EXIT processing found the conversational transaction enqueued for an application or transmission across an MSC link, but it has not yet been scheduled for processing. The transaction has been discarded.

### *Example 3 for /EXIT command*

Entry ET:

```
/EXIT
```

Response ET:

```
DFS577I  EXIT COMPLETED, TRANSACTION STILL ACTIVE
```

Explanation: /EXIT processing found the conversational transaction had been or is in the process of being transmitted across an MSC link.

### *Example 4 for /EXIT command*

Entry ET:

```
/EXIT CONVERSATION 0001
```

Response ET:

```
DFS058I  EXIT COMMAND COMPLETED
```

Explanation: The active or held conversation (0001) is terminated if the terminal is in conversation.

*Example 5 for /EXIT command*

Entry ET:

```
/EXIT CONVERSATION 0001 LINE 10 PTERM 2
```

Response ET:

```
DFS058I  EXIT COMMAND COMPLETED
```

Explanation: Used from another PTERM or master terminal when a PTERM in conversation is "locked," waiting for a response. /EXIT CONVERSATION should be used before a /START LINE command that resets all conversations active on the line.

## Chapter 31. EXPORT command

Use the EXPORT command to export runtime resource and descriptor definitions from the online IMS system to a resource definition data set (RDDS) or the IMSRSC repository as stored resource definitions.

The resource and descriptor definitions can then be:

- Added to IMS dynamically through the IMPORT command
- Imported from an RDDS during an IMS cold start
- Imported from the repository during an IMS cold start, or an IMS warm start or emergency restart for changed list processing

The resource and descriptor definitions are read from the repository during processing of the QUERY xxx SHOW(DEFN) command.

Subsections:

- “Environment”
- “Syntax”
- “Keywords” on page 789
- “Usage notes” on page 796
- “Output fields” on page 799
- “Return, reason, and completion codes ” on page 800
- “Examples” on page 804

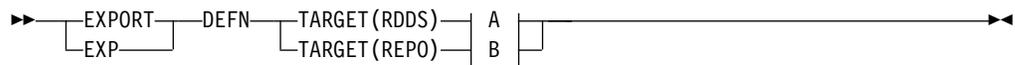
### Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

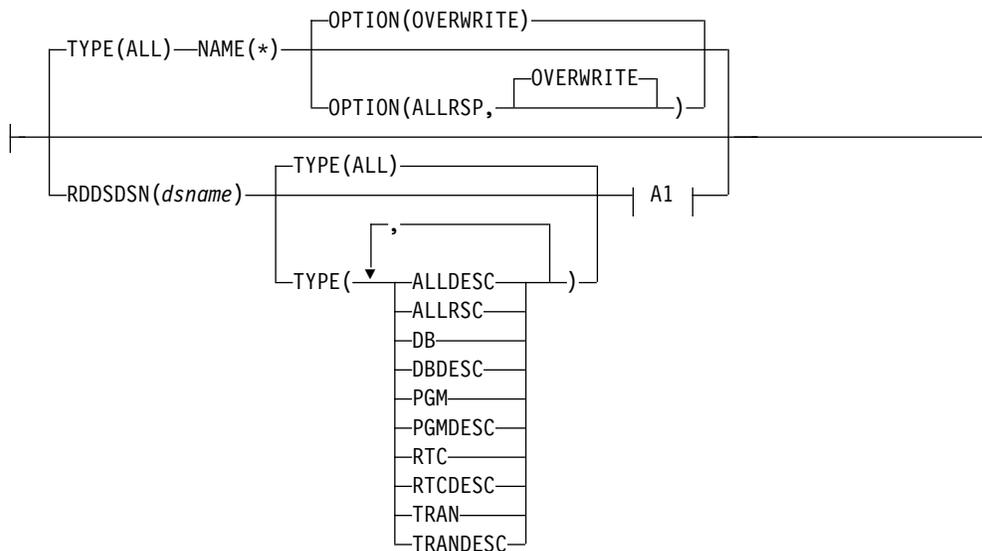
Table 327. Valid environments for the EXPORT command and keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
EXPORT	X	X	X
DEFN	X	X	X
ENDTIME	X	X	X
NAME	X	X	X
OPTION	X	X	X
RDDSDSN	X	X	X
SET	X	X	X
STARTTIME	X	X	X
TARGET	X	X	X
TYPE	X	X	X

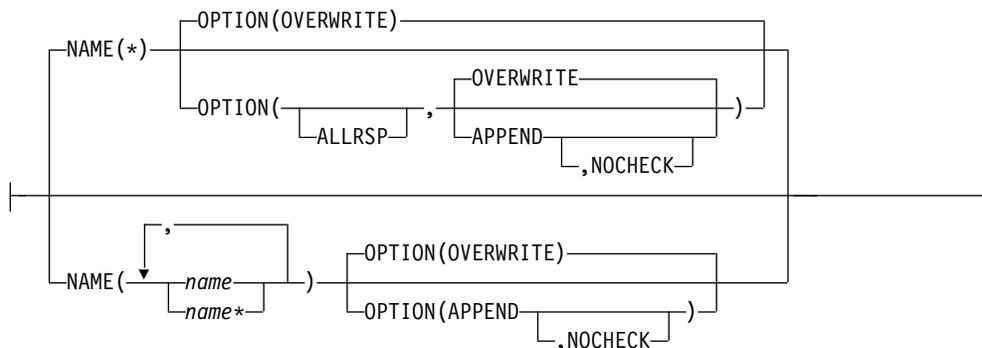
### Syntax



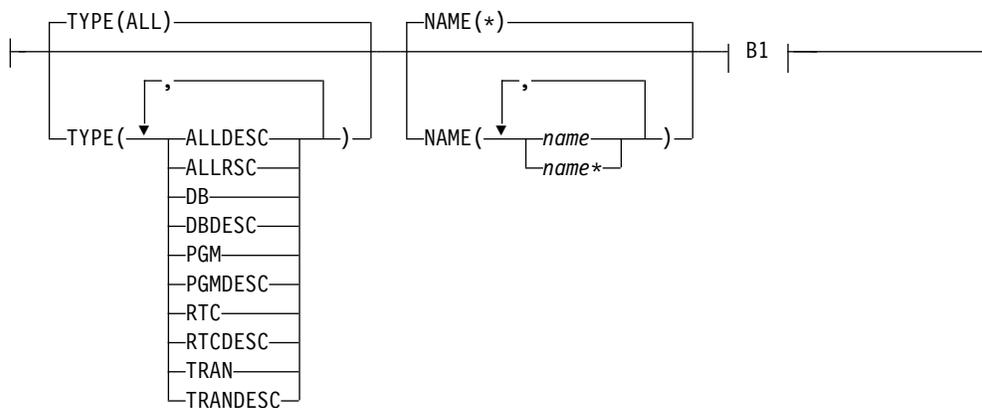
**A:**



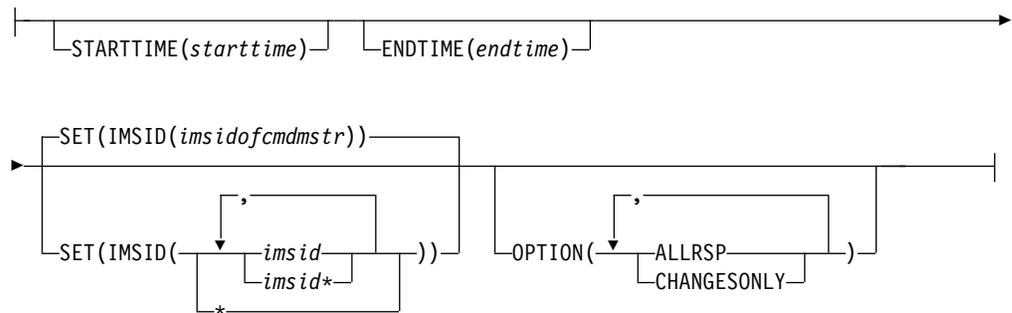
**A1:**



**B:**



**B1:**



## Keywords

The following keywords are valid for the EXPORT command:

### DEFN

Specifies that resource and descriptor definitions are to be exported.

### ENDTIME

Specifies the end time until which new and modified runtime resource definitions are to be gathered to be exported to the repository. ENDTIME is the local time and is specified in the following format:

*yyyy.ddd hh:mm:ss.ms.th*

where *yyyy.ddd* is required, and the rest of the parameters (*hh*, *mm*, *ss*, *ms*, and *th*) are optional and default to 0.

When the ENDTIME keyword is specified, all the resource definitions that have either a create time stamp or an update time stamp less than the ENDTIME value specified are exported to the repository. If the TYPE, NAME, or STARTTIME keyword is specified with ENDTIME, the resource definitions that match all the specified keywords are exported to the repository.

If ENDTIME is not specified, all the resources that are found that match the STARTTIME, NAME, and TYPE keywords are exported to the repository.

**Note:** Consideration must be taken in specifying the ENDTIME value when there is a change in time for the Daylight Saving Time.

### NAME()

Specifies the 1-8 character name of a resource or descriptor. Wildcards can be specified in the name. The name is a repeatable parameter. If the NAME parameter specified is a specific name or a wildcard, command responses are returned for all the resource names that are processed. NAME is an optional parameter, and if not specified, it defaults to NAME(\*). For NAME(\*), command responses are returned only for the resource and descriptor names that resulted in an error. OPTION(ALLRSP) can be specified with NAME(\*) to obtain the command responses for all of the resource and descriptor names that are processed.

### OPTION()

Specifies additional functions to be performed along with the command.

### ALLRSP

Indicates that response lines are to be returned for all resources and descriptors that are processed by the command. If NAME(\*) is specified,

the default action is to return response lines only for the resources and descriptors that resulted in an error. `OPTION(ALLRSP)` can be used with `NAME(*)` to request response lines for all resources and descriptors processed by the command. If a name value other than `NAME(*)` is specified, the default action is to return a response line for all resources and descriptors processed by the command. If `OPTION(ALLRSP)` is specified with a name value other than `NAME(*)`, the `ALLRSP` parameter is ignored.

For HALDB databases, only the definition of the HALDB master database is exported. HALDB partition definitions are not exported, because the partition definitions are maintained in the RECON data set. `OPTION(ALLRSP)` does not include the partitions definitions in the output and therefore does not flag them with a nonzero completion code.

When coding the Destination Creation exit routine (`DFSINSX0`), the user can indicate whether the definitions for transaction and program resources created by the exit should be exported. Transaction and program resource definitions that are not exported are not included in the `OPTION(ALLRSP)` output.

IMS resources such as the Fast Path utility (`DBF#FPU0`) and the IMS defined descriptors (`DBFDSRT1`, `DFSDSDB1`, `DFSDSPG1`, and `DFSDSTR1`) are not exported. `OPTION(ALLRSP)` does not include these resources and descriptors in the output and therefore does not flag them with a nonzero completion code.

#### **APPEND**

Indicates that the resource and descriptor definitions are to be appended to the end of the specified RDDS data set. If `OPTION(NOCHECK)` is not specified, validity checking is done to ensure that the IMS performing the export is the same IMS that previously exported definitions to the RDDS. The `APPEND` option is valid only when exporting to a non-system RDDS.

`APPEND` is mutually exclusive with `OVERWRITE`. If `APPEND` results in multiple definitions for the same resource in the RDDS, IMS uses the definition that was exported most recently when the definitions are imported.

#### **CHANGESONLY**

Specifies that the runtime resource definitions with changes (resources and descriptors that are created or updated) since the last successful export are to be exported to be hardened in the repository.

`OPTION(CHANGESONLY)` is supported only for `TARGET(REPO)`.

The list of the names of the resources that are to be exported to the repository can also be provided on the command input. If a list of resource names is specified, the resources that have changes (creates or updates) since the last successful export are exported to the repository. If the specified resource does not have any changes, a completion code is returned for that resource. The export is completed for all other resources that have changes.

A list of the names of the resources that are exported to the repository is returned on the command response.

When `NAME(*)` is specified with `OPTION(CHANGESONLY)`, it is treated as if `OPTION(ALLRSP)` was specified; all resource names that are exported to the repository are returned on the command response. If no resource

names have any changes when NAME(\*) is specified, no resources are exported and the command results in an error reason code.

#### **NOCHECK**

Indicates that no ID checking is to be done when definitions are appended to an RDDS. If OPTION(NOCHECK) is specified, the IMS performing the export can be a different IMS than the one that performed the previous export. The NOCHECK option is valid only when exporting to a non-system RDDS. NOCHECK is mutually exclusive with OVERWRITE and can be specified only if APPEND is specified.

#### **OVERWRITE**

Indicates that the resource and descriptor definitions are to overwrite any existing definitions in the RDDS. APPEND is mutually exclusive with OVERWRITE. OVERWRITE is the default and the only valid option when exporting to a system RDDS.

#### **RDDSDSN(*dsname*)**

An optional keyword that is used to specify the name of a non-system RDDS to which resource and descriptor definitions are to be exported. When the RDDSDSN() keyword is omitted from the EXPORT command, the definitions are written to the system RDDS that contains the oldest data. The data set name specified in the RDDSDSN() keyword cannot be the name of any of the system RDDSs defined in the DYNAMIC\_RESOURCES section of the DFSDFxxx PROCLIB member. If a set of system data sets is not defined in the DYNAMIC\_RESOURCES section of the DFSDFxxx PROCLIB member, the RDDSDSN() keyword is required on the EXPORT command.

If using the RDDSDSN() keyword to specify a specific data set, care should be taken when routing the EXPORT command by using Operations Manager (OM) routing. If the command is routed to multiple IMS systems in the IMSplex, only the IMS selected as the command master will process the command.

The data set specified for the RDDSDSN() keyword must be a physical sequential (DSORG=PS) BSAM data set. The data set must exist and be cataloged. When creating and allocating the RDDS data set, ensure that an end-of-file (EOF) mark is placed at the beginning of the data set. Failure to do so can produce unpredictable results. To place an EOF mark at the beginning of the data set, use program IEBGENER. The sample job skeleton DFSRDDAL, which is in the IMS.SDFSSLIB data set, can be used for this step. An alternative method of allocating the RDDS data sets is to use the ALLOCATE function of ISPF utilities. The ALLOCATE function places an EOF mark correctly at the beginning of the data set. Do not use program IEFBR14 to allocate the RDDS data sets, because IEFBR14 does not place an EOF mark at the beginning of the data set.

IMS allocates the RDDS with DISP=OLD during the export process. If you attempt to allocate a new data set by using JCL in one step of a job and then try to issue the EXPORT command by using the batch SPOC in another step of the same job, the batch export will fail with return code 10 (environment error) and reason code 4118 (data set allocate error), followed by message DFS2503W with error reason code 210 (data set is allocated to another job). This is because IMS cannot obtain an exclusive enqueue on the newly allocated RDDS.

Generation data group (GDG) data sets are not fully supported for RDDSs. Relative generation numbers cannot be used when specifying an RDDS name. For example, if you specify RDDSDSN(IMS.EXP.GDG(0)) or RDDSDSN(IMS.EXP.GDG(+1)), the EXPORT command will fail with message

DFS2503W and error reason code 35C (invalid parm specified in text unit). Specifying the absolute generation number for a GDG data set does work. For example, RDDSDSN(IMS.EXP.GDG.G0001V00) can be specified.

## **SET**

Specifies what actions are to be set.

### **IMSID**

Specifies one or more IMS IDs whose IMS resource lists are to be updated by the EXPORT command.

An IMS resource list is a list of resource or descriptor names that is used to define the resources for an IMS system. IMS resource lists reside in the repository and are used during IMS cold start to identify all the resource and descriptor definitions that are to be read during the IMS cold start.

For each IMS system defined in the repository, one IMS resource list is created for each resource type defined for the IMS:

- Databases
- Database descriptors
- Programs
- Program descriptors
- Routing codes
- Routing code descriptors
- Transactions
- Transaction descriptors

When a resource definition is written to the repository for an IMS, the resource name is added to the appropriate IMS resource list if it does not already exist in the list. If an IMS resource list does not exist for that IMS and resource type, an IMS resource list for that IMS and resource type is created, and the resource name is added to it.

For example, if the EXPORT DEFN TARGET(REPO) NAME(TRANABC) TYPE(TRAN) SET(IMSID(IMS1,IMS2,IMS3)) command is issued, the IMS resource lists for each IMS ID specified (IMS1, IMS2, and IMS3) might be updated along with the resource definition for transaction TRANABC. If TRANABC is not already in the transaction resource list for an IMS system, it is added to the list. If no transactions have been defined for the IMS, the IMS resource list for transactions is created for the IMS, and TRANABC is added to the list. If TRANABC is already in the resource list for the IMS, the resource list is not updated.

The IMSID value specified on the EXPORT command identifies the IMS resource lists that are to be created or updated when a stored resource definition is created or updated in the repository.

The IMSID value can be a specific IMS ID, a list of specific IMS IDs, a wildcard name of \*, or a wildcard name with \* or %. The IMSID must be a 1- to 4-alphanumeric character value, and can include any of the following special characters: # \$ % \* @

If the IMSID specified is a wildcard name other than \*, the IMS resource lists for those IMS systems whose names match the IMSID wildcard name will be updated. The list of IMS IDs whose IMS resource lists are updated is returned in command output. If the repository is empty and there are no resource definitions in the repository, the EXPORT command fails.

IMSID(\*) indicates that the IMS resource lists for all IMS systems defined in the repository are to be updated with the resource names of the exported definitions. In other words, the resource definitions being

exported will belong to all IMS systems that currently have resource definitions in the repository. If the repository is empty and there are no resource definitions in the repository, the EXPORT command fails.

The IMSID value can also be specified as a list of one or more IMS IDs of the IMS systems for which the resource lists must be updated. When one or more IMS IDs are specified, only the IMS resource lists for the specified IMS IDs are updated in the repository. If the IMS resource lists for the IMS IDs specified do not exist, the IMS resource lists are created.

The IMSID value that is specified does not have to belong to an active IMS in the IMSplex. The IMSID of an IMS that is yet to be started can be specified.

When exporting definitions for a new IMS, you must issue the EXPORT command with a specific IMSID value to create resource definitions in the repository. When a specific IMSID value is specified, the IMS resource list is created for IMS even if the IMS is currently not registered to RM. If the resource definitions are created for an IMS in error, you can use the DELETE DEFN command with the FOR(*imsid*) option to delete the resource definitions and the IMS resource lists.

If SET(IMSID()) is not specified, the IMSID value defaults to the IMS ID of the command master IMS. An IMS resource list is created for the command master IMS if one does not exist. You are required to route or direct the command to each IMS by using the ROUTE keyword.

#### **STARTTIME**

Specifies the start time from which all new and modified runtime resource definitions are to be gathered and exported to the repository. STARTTIME is the local time and is specified in the following format:

*yyyy.ddd hh:mm:ss.ms.th*

where *yyyy.ddd* is required, and the rest of the parameters (*hh*, *mm*, *ss*, *ms*, and *th*) are optional and default to 0.

When the STARTTIME keyword is specified, all the resource definitions that have either a create time stamp or an update time stamp greater than or equal to the STARTTIME specified will be exported to the repository. If the ENDTIME, TYPE, or NAME keyword is specified with STARTTIME, the resource definitions that match all the specified keywords are exported to the repository.

The list of the resource names that are exported to the repository is provided on the command output.

When NAME(\*) is specified with the STARTTIME keyword with or without the ENDTIME keyword, it is treated as if OPTION(ALLRSP) was specified, and all resource names that are exported to the repository are returned on the command response. If no resource names have any changes when NAME(\*) is specified, no resources are exported, and the command results in an error reason code.

**Consideration for daylight saving time:** Consideration must be taken in specifying the STARTTIME parameter when there is a change in time for the Daylight Saving Time in regard to the current local time. For the spring time adjustment, time is adjusted forward by one hour at 2:00 a.m. If you want to export all the resources that have changed since 12:30 a.m. (before the time change), you would have to specify STARTTIME of 1:30 a.m. on the command

because 12:30 a.m. (before time change) corresponds to 1:30 a.m. (after time change). For the fall time adjustment, time is adjusted backward by one hour at 2:00 a.m. If you want to export all resources that have changed since 12:30 a.m. (before the change), you would have to enter a STARTTIME of 11:30 p.m. on the prior day because 12:30 a.m. (before time change) corresponds to 11:30 p.m. (after time change).

#### **TARGET()**

Specifies where resource and descriptor definitions are to be exported.

#### **RDDS**

Specifies that resource and descriptor definitions are to be exported to an RDDS. If the RDDS(SN) keyword is not used to specify the name of a non-system RDDS, the system RDDS with the oldest data is used.

Even when IMS is enabled to use the repository, you can use the EXPORT DEFN TARGET(RDDS) command to export runtime resource and descriptor definitions to a system RDDS or a non-system RDDS if the RDDS is defined.

#### **REPO**

Indicates that the runtime resource definitions of the resource names specified on the NAME keyword are to be exported to the IMSRSC repository. Only the resource definitions of the resource names that match the NAME keyword are created or updated in the repository. If the resource definition does not exist in the repository, it is created. If the resource definition exists in the repository, it is replaced. TARGET(REPO) is valid only when IMS is enabled to use the repository.

A repository can maintain resource and descriptor definitions for up to 64 IMS systems in an IMSplex.

All resource definitions are exported to the repository as a single unit of work (UOW). If any resource results in an error being written to the repository, no resource definitions are written.

The system-defined descriptors are not exported to the repository. The HALDB partition databases are not exported to the repository when TYPE(ALL) or TYPE(DB) is specified.

If any RM CSLRPUPD request errors are returned for the EXPORT command, you must check to see if there are any in-doubts. If the error is before the UOW is committed, all updates to the repository are backed out. If the failure is during UOW commit and if the repository primary data set is updated, the UOW is committed. You must issue the QUERY xxx SHOW(DEFN) commands to verify if the changes are hardened in the repository, and if not, reissue the EXPORT command.

The EXPORT DEFN TARGET(REPO) command is processed only by the IMS that has been selected as the command master. All other IMS systems that receive the command return an IRSN\_NOTMSTR return and reason code. In a cloned IMS configuration, the EXPORT command can be issued with SET(IMSID(\*)) so that the exported definitions are written to the repository and all the IMS resource lists in the repository are updated. If a resource name does not exist in an IMS resource list and SET(IMSID(\*)) is specified, the resource name is added to the IMS resource list. In a non-cloned IMS configuration, issue the EXPORT command to each IMS individually so that each IMS acts as the command master and exports its own resource and descriptor definitions. In this case, the EXPORT command with the

default SET(IMSID()) keyword can be issued and the same command will be routed to each IMS in the IMSplex.

When an FP(E) transaction is created in the repository by the EXPORT DEFN command, the routing code for the FP(E) transaction will also be created in the repository by RM.

**Recommendations:**

- Avoid issuing the EXPORT DEFN command with TARGET(REPO) NAME(\*) TYPE(ALL) during peak hours because exporting all the resource definitions to the repository can incur a larger than normal command response time. However, while the resource definitions are being written to the repository, there is no impact to other work in IMS.
- In order to avoid incurring performance overhead of accessing resources in the repository that have not been updated, export to the repository only those resources that have been updated. If you do not know the names of the resources that have been updated recently and need to be exported, you can issue the EXPORT command with the STARTTIME keyword so that you can export all resources that have been updated since a specific time.

During the EXPORT DEFN TARGET(REPO) command processing, resource validation is performed in RM as follows:

- When a new transaction or routing code definition is added to the repository, resource validation checking is done to ensure that a definition for the associated program either exists in the repository or is being added to the repository with the EXPORT command.
- When an existing program, transaction, or routing code definition is updated in the repository, resource validation checking is done to ensure that the attributes being updated do not conflict with the attributes of any associated resource definitions. For example, if the transaction is modified to FP(E), a check is done to ensure that the associated program is defined as FP exclusive.

**TYPE()**

Specifies the type of resource or descriptor definitions to export.

**ALL**

Specifies that all definitions for all resource and descriptor types (DB, DBDESC, PGM, PGMDESC, RTC, RTCDESC, TRAN, and TRANDESC) are to be exported.

**ALLDESC**

Specifies that all definitions for all descriptor types (DBDESC, PGMDESC, RTCDESC, and TRANDESC) are to be exported. TYPE(ALLDESC) is not valid when exporting to a system RDDS or when exporting from an IMS that does not have DRD enabled.

**ALLRSC**

Specifies that all definitions for all resource types (DB, PGM, RTC, and TRAN) are to be exported. TYPE(ALLRSC) is not valid when exporting to a system RDDS.

**DB**

Specifies that one or more database definitions are to be exported. TYPE(DB) is not valid when exporting to a system RDDS. If TYPE(DB) is specified on a DCCTL system, completion code 10 ("NO RESOURCES FOUND") is returned.

**DBDESC**

Specifies that one or more database descriptor definitions are to be exported. TYPE(DBDESC) is not valid when exporting to a system RDDS or when exporting from an IMS that does not have DRD enabled. If TYPE(DBDESC) is specified on a DCCTL system, completion code 10 ("NO RESOURCES FOUND") is returned.

**PGM**

Specifies that one or more program definitions are to be exported. TYPE(PGM) is not valid when exporting to a system RDDS.

**PGMDESC**

Specifies that one or more program descriptor definitions are to be exported. TYPE(PGMDESC) is not valid when exporting to a system RDDS or when exporting from an IMS that does not have DRD enabled.

**RTC**

Specifies that one or more routing code definitions are to be exported. TYPE(RTC) is not valid when exporting to a system RDDS. If TYPE(RTC) is specified on a DBCTL system, completion code 10 ("NO RESOURCES FOUND") is returned.

**RTCDESC**

Specifies that one or more routing code descriptor definitions are to be exported. TYPE(RTCDESC) is not valid when exporting to a system RDDS or when exporting from an IMS that does not have DRD enabled. If TYPE(RTCDESC) is specified on a DBCTL system, completion code 10 ("NO RESOURCES FOUND") is returned.

**TRAN**

Specifies that one or more transaction definitions are to be exported. TYPE(TRAN) is not valid when exporting to a system RDDS. If TYPE(TRAN) is specified on a DBCTL system, completion code 10 ("NO RESOURCES FOUND") is returned.

**TRANDESC**

Specifies that one or more transaction descriptor definitions are to be exported. TYPE(TRANDESC) is not valid when exporting to a system RDDS or when exporting from an IMS that does not have DRD enabled. If TYPE(TRANDESC) is specified on a DBCTL system, completion code 10 ("NO RESOURCES FOUND") is returned.

## Usage notes

The EXPORT command can be issued through TSO SPOC, IBM Management Console for IMS and DB2 for z/OS, the Manage Resources options in the IMS Applications menu, or other user-written programs that issue commands through OM API. These commands can also be issued to an IMSplex by using the Batch SPOC utility.

**Important:** Commands such as CREATE, DELETE, EXPORT, IMPORT, and UPDATE, which affect the definitional attributes of resources and descriptors, are not allowed while the EXPORT command is in progress.

The EXPORT DEFN TARGET(RDDSDSN) command writes resource and descriptor definitions either to the data set specified by the RDDSDSN() keyword or, if the RDDSDSN() keyword is omitted, to the system RDDS containing the oldest data. A system RDDS is one of the RDDS data sets defined with the RDDSDSN= parameter in the DYNAMIC\_RESOURCES section of the DFSDFxxx PROCLIB

member. A system RDDS contains all of the resource and descriptor definitions for a single IMS. If using automatic import or automatic export, each IMS must have its own set of system RDDS data sets. When exporting to a system RDDS, all of the IMS resource and descriptor definitions must be exported. When exporting to a non-system RDDS, all or some of the IMS resource and descriptor definitions can be exported.

The EXPORT DEFN TARGET(REPO) command writes resource and descriptor definitions to the IMSRSC repository. The repository contains all of the resource and descriptor definitions for all IMS systems in the IMSplex that share the repository. When exporting to the repository, all or some of the IMS resource and descriptor definitions can be exported. During export to the repository, you can specify, by using the SET(IMSID()) keyword, a list of the IMS systems in the IMSplex to which the resource definitions must be defined. The specified resource definitions and the corresponding IMS resource lists in the repository are either updated or created. For more information about IMS resource lists, see the description for the SET(IMSID()) keyword.

The EXPORT DEFN TARGET(REPO) command does not delete resource definitions from the repository. To delete resource definitions from the repository, issue the DELETE DEFN command.

To check the contents of the repository, issue the QUERY DB | DBDESC | PGM | PGMDESC | RTC | RTCDESC | TRAN | TRANDESC commands with the SHOW(DEFN,GLOBAL), SHOW(IMSID), or SHOW(DEFN,IMSID,GLOBAL) keyword. If the repository is empty and contains no resource definitions, the QUERY command fails.

When exporting to a system RDDS, all definitions in the system RDDS are overwritten with the definitions being exported. When exporting to a non-system RDDS, the user has the option of overwriting the existing definitions with the new definitions, or appending the new definitions to the end of the data set. When the APPEND option is used, all of the definitions that meet the criteria specified on the EXPORT command are exported and appended to the existing definitions in the RDDS. If the export results in multiple definitions for the same resource in the RDDS, the last definition written to the RDDS is the definition used by the IMPORT command when creating the runtime resource definition for the resource or descriptor.

Definitions for IMS resources such as the Fast Path utility (DBF#FPU0) and the descriptors defined by IMS (DBFDSRT1, DFSDSDB1, DFSDSPG1, and DFSDSTR1) cannot be exported to the RDDS or the repository.

The EXPORT command is not allowed on the XRF alternate, RSR tracker, or FDBR region.

Export to a system RDDS is allowed only in a DRD environment (DFSDFxxx or DFSCGxxx defined with MODBLKS=DYN). Export to a non-system RDDS is allowed in a non-DRD environment (DFSDFxxx or DFSCGxxx defined with MODBLKS=OLC, or MODBLKS not defined); however, descriptors do not exist and cannot be exported.

The EXPORT command is defined as ROUTE=ALL to OM. When the EXPORT command is issued, OM routes the EXPORT command to all active IMS systems (or to all IMS systems specified in the ROUTE list if the ROUTE parameter is

specified on the CSLOMCMD or CSLOMI request) that have registered interest in the command. The EXPORT command is processed as follows by the IMS systems that receive the command:

- When exporting to a system RDDS, the EXPORT command is processed by all IMS systems that receive the command. Each IMS that receives the command exports its resource and descriptor definitions to its own system RDDS. The system RDDSs are not shared among IMS systems, so all IMS systems that receive the command can process the command simultaneously without data set contention.
- When exporting to a non-system RDDS, the EXPORT command is treated as if the command was defined to OM as ROUTE=ANY. OM selects one IMS to be the command master. The command master IMS exports its resource and descriptor definitions to the specified non-system RDDS. All non-master IMS systems that receive the command return without exporting any definitions with an "IMS not master" return and reason code to OM. The export to the non-system RDDS is processed by only one IMS to avoid data set contention and to prevent the resource and descriptor definitions from being overwritten if the RDDS is shared between IMS systems.
- When exporting to the repository, the EXPORT command is treated as if the command was defined to OM as ROUTE=ANY. OM selects one IMS to be the command master. The command master IMS exports its resource and descriptor definitions to the repository. All non-master IMS systems that receive the command return without exporting any definitions with an "IMS not master" return and reason code to OM.

If the EXPORT command is to be processed by a specific IMS, the EXPORT command must be routed to that IMS.

**Recommendation:** When exporting to a non-system RDDS, each IMS should export its definitions to a unique data set. If one RDDS is shared between systems, the resource and descriptor definitions written by one IMS might be overwritten by another IMS.

**Considerations for Fast Path exclusive transactions and their associated routing codes:** When a definition for a Fast Path exclusive transaction is exported to the repository, the definition for the associated routing code is automatically created or updated in the repository if the routing code definition is not exported at the same time.

The export process fails if all of the following conditions are met:

- A Fast Path exclusive transaction definition is exported to the repository without its associated routing code definition
- The transaction definition is being created for the IMS
- A routing code definition with the same name already exists in the repository
- Either the PGM value or the INQ value in the routing code definition is different from the PGM or INQ value in the new transaction definition

If a Fast Path exclusive transaction definition is exported to the repository without its associated routing code definition and a definition for the transaction already exists in the repository for the IMS and either the PGM value or the INQ value is being updated, the PGM value, the INQ value, or both are automatically updated in the associated routing code definition.

If a routing code definition is exported to the repository without an associated transaction definition, the export process fails if the routing code already exists in

the repository for the IMS and the routing code definition in the repository is associated with a Fast Path exclusive transaction definition.

## Output fields

When TARGET(RDDS) is specified, message DFS3404I is returned by each IMS that processes the EXPORT command. It indicates whether all, some, or none of the specified resource or descriptors were successfully exported. It also contains the name of the data set to which the resource or descriptor definitions were exported. Each DFS3404I message is encapsulated in a single set of <msg> </msg> XML tags.

When TARGET(REPO) is specified, the EXPORT command results in message DFS3406I, which indicates that the export to the repository was successful, or DFS3407E, which indicates that the export was not successful. These messages include the repository name and repository type. If the export to the repository is not successful due to a CSLRPUPD error, the DFS3407E message contains the CSLRPUPD request return code and reason code. The CSLRPUPD return and reason codes are described in the CSLRRR macro.

In addition to the message, command response output is also returned in XML tag structure. The following table shows the EXPORT response output fields. The columns in the table are as follows:

### Short label

Contains the short label generated in the XML output.

### Long label

Contains the long label generated in the XML output.

### Keyword

Identifies keyword on the command that caused the field to be generated. The value *error* appears for output fields that are returned only in the case of an error. N/A (not applicable) appears for output fields that are always returned.

### Meaning

Provides a brief description of the output field.

Table 328. Output fields for the EXPORT command

Short label	Long label	Keyword	Meaning
CC	CC	N/A	Completion code.
CCTXT	CCText	<i>error</i>	Completion code text that briefly explains the meaning of the nonzero completion code.
ERRT	ErrorText	N/A	The error text that describes the error completion code.
MBR	MbrName	N/A	IMSpIex member that built the output line.
NAME	Name	NAME	The name of the resource or descriptor definition that was created or updated in the repository, or the name of the IMS resource list that was created or updated in the repository.
TYPE	Type	TYPE	The resource or descriptor type, which can be DB, DBDESC, PGM, PGMDESC, RTC, RTCDESC, TRAN, or TRANDESC; or LIST for IMS resource lists.

## Return, reason, and completion codes

The following table includes the return and reason codes and a brief explanation of the codes. The return or reason code returned for the command might also indicate an error from a CSL request.

*Table 329. Return and reason codes for the EXPORT command*

Return code	Reason code	Meaning
X'00000000'	X'00000000'	Command completed successfully. The command output contains a line for each resource, accompanied by its completion code. If NAME(*) is specified without OPTION(ALLRSP), no output lines are returned. See the completion code table for details.
X'00000004'	X'00001000'	The EXPORT command was not processed on the IMS system, because the IMS system is not the command master. No resource information is returned.
X'00000008'	X'0000211F'	Unable to determine the export data set name. The RDDSDSN() keyword was not specified on the command, and there are no system RDDSs defined.
X'00000008'	X'0000213A'	The data set specified for the RDDSDSN() keyword is a system RDDS.
X'00000008'	X'0000213B'	OPTION(OVERWRITE) is mutually exclusive with OPTION(APPEND).
X'00000008'	X'0000213C'	A value other than NAME(*) was specified. NAME(*) is required when exporting to a system RDDS.
X'00000008'	X'0000213D'	A value other than TYPE(ALL) was specified. TYPE(ALL) is required when exporting to a system RDDS.
X'00000008'	X'0000213E'	OPTION(APPEND) is not allowed when exporting to a system RDDS.
X'00000008'	X'0000213F'	OPTION(NOCHECK) can be specified only if OPTION(APPEND) is specified.
X'00000008'	X'00002146'	Invalid STARTTIME value.
X'00000008'	X'00002147'	TARGET(RDDS) and OPTION(CHANGESONLY) conflict.
X'00000008'	X'00002148'	TARGET(RDDS) and SET(IMSID()) conflict.
X'00000008'	X'00002149'	TARGET(RDDS) and STARTTIME conflict.
X'00000008'	X'0000214A'	TARGET(RDDS) and ENDTIME conflict.
X'00000008'	X'0000214B'	TARGET(REPO) and OPTION(APPEND) conflict.
X'00000008'	X'0000214C'	TARGET(REPO) and RDDSDSN conflict.
X'00000008'	X'0000214D'	Either an invalid ENDTIME value was specified or the ENDTIME value specified was less than or equal to the STARTTIME value specified.
X'00000008'	X'0000214F'	RM resource validation error.
X'0000000C'	X'00003000'	Command was successful for some resources but failed for others. The command output contains a line for each resource, accompanied by its completion code. If NAME(*) is specified without OPTION(ALLRSP), output lines are only returned for resources with nonzero completion codes.

Table 329. Return and reason codes for the EXPORT command (continued)

Return code	Reason code	Meaning
X'0000000C'	X'00003004'	Command was not successful for any of the resources. The command output contains a line for each resource, accompanied by its completion code. See the completion code table for details.
X'00000010'	X'00004004'	No CQS address space.
X'00000010'	X'0000400C'	Command is not valid on the XRF alternate.
X'00000010'	X'00004014'	Command is not valid on the RSR tracker.
X'00000010'	X'00004018'	No resource structure, or resource structure is not available.
X'00000010'	X'00004100'	Resource structure is full.
X'00000010'	X'00004104'	No RM address space.
X'00000010'	X'00004108'	No SCI address space.
X'00000010'	X'00004114'	Unable to access the RDDDS. The RDDDS is either a system RDDDS or the data set specified with the RDDSDSN() keyword.
X'00000010'	X'00004118'	Unable to allocate the RDDDS. The RDDDS is either a system RDDDS or the data set specified with the RDDSDSN() keyword.
X'00000010'	X'0000411C'	The EXPORT command was issued to append definitions to an RDDDS, but the RDDDS contains invalid data.
X'00000010'	X'00004120'	Online change phase is in progress.
X'00000010'	X'00004300'	Command is not allowed, because online change for MODBLKS is enabled (DFSDFxxx or DFSCGxxx defined with MODBLKS=OLC, or MODBLKS not defined) and an RDDDS was not specified with the RDDSDSN() keyword.
X'00000010'	X'00004302'	The RDDDS is full. No more definitions can be written to the RDDDS.
X'00000010'	X'0000432C'	Invalid value is specified for the TYPE() keyword. ALLDESC, DBDESC, PGMDESC, RTCDESC and TRANDESC are not allowed if online change for MODBLKS is enabled (DFSDFxxx or DFSCGxxx defined with MODBLKS=OLC, or MODBLKS not defined).
X'00000010'	X'00004330'	None of the system RDDDSs defined in the DFSDFxxx PROCLIB member are available for export.
X'00000010'	X'00004334'	The RECFM of the RDDDS is not VB.
X'00000010'	X'00004338'	Invalid RDDDS block size. The RDDDS must be defined with a block size 4096 - 32760.
X'00000010'	X'0000433C'	The RDDDS specified with the RDDSDSN() keyword contains definitions written by another IMS. If new definitions are to be appended to the RDDDS by a different IMS, OPTION(NOCHECK) must be specified.
X'00000010'	X'00004500'	IMS is not enabled to use the repository.
X'00000010'	X'00004502'	Repository is not available.
X'00000010'	X'00004504'	Repository spare recovery is in progress.

Table 329. Return and reason codes for the EXPORT command (continued)

Return code	Reason code	Meaning
X'00000010'	X'00004505'	No IMS resource list. This return code can be returned if the EXPORT command with SET(IMSID(*)) is issued and the repository is empty. The SET(IMSID(*)) keyword cannot be processed because there are no IMS resource lists in the repository. Issue the EXPORT command again by specifying a list of IMS IDs, or issue the EXPORT command again without SET(IMSID()) so that the command defaults to the command master IMS.
X'00000010'	X'00004507'	Repository access denied.
X'00000010'	X'00004508'	Repository maximum put length exceeded.
X'00000010'	X'00004509'	RM data version is lower than the IMS data version.
X'00000010'	X'0000450A'	Repository Server is being shut down.
X'00000010'	X'0000450B'	Repository Server is not available.
X'00000010'	X'0000450C'	Repository Server is busy.
X'00000010'	X'0000450D'	RM failed to define some of the internal fields related to the IMSRSC repository.
X'00000010'	X'0000450E'	An IMSID list is required.
X'00000010'	X'0000450F'	Maximum number of IMSIDs for the repository was reached.
X'00000010'	X'00004510'	IMSID member table changed during command processing.
X'00000014'	X'00005000'	IMODULE GETSTOR error.
X'00000014'	X'0000500C'	AWE could not be obtained.
X'00000014'	X'00005004'	DFSOCMD response buffer could not be obtained.
X'00000014'	X'00005008'	DFSPOOL storage could not be obtained.
X'00000014'	X'00005010'	Unable to obtain latch.
X'00000014'	X'00005018'	Unable to obtain a 64-bit buffer.
X'00000014'	X'0000501C'	IMODULE GETMAIN error.
X'00000014'	X'00005100'	RM request error.
X'00000014'	X'00005104'	Unexpected CQS error.
X'00000014'	X'00005108'	SCI request error.
X'00000014'	X'00005110'	Repository error.
X'00000014'	X'0000551C'	A request to update the repository was rejected. Either a corrupted resource list was detected, or the update request was rejected to prevent a corruption of the resource list. Contact IBM Software Support. Send in the output from the command in error, a dump of the local RM address space where the command executed or the dump of all RM address spaces in the IMSplex. Also, send a copy of the IMSRSC repository RID and RMD data sets.
X'00000014'	X'00005FFF'	The EXPORT command could not be processed because of an internal error.

Errors unique to the processing of this command are returned as completion codes. The following table includes an explanation of the completion codes.

*Table 330. Completion codes for the EXPORT command*

Completion code	Completion code text	Meaning
0		Command completed successfully for this resource or resource descriptor.
10	NO RESOURCES FOUND	Resource or descriptor was not found, or the wildcard parameter specified does not match any resource or descriptor names.
2E	NOT ALLOWED FOR RESOURCE	Export of resource or descriptor definition is not allowed. The definitions for IMS defined resources and HALDB partitions cannot be exported.
36	FP=E/FP=N PGM CONFLICT	The transaction is defined as Fast Path exclusive FP(E), and the program is defined as non-Fast Path FP(N).
39	FP=E TRAN FOR RTC EXISTS	The routing code definition is being updated, but the routing code is associated with a Fast Path exclusive FP(E) transaction.
3E	FP=N/FP=E PGM CONFLICT	The transaction is defined as non-Fast Path FP(N), and the program is defined as Fast Path exclusive FP(E).
3F	FP=P/BMPTYPE=Y CONFLICT	The transaction is defined as Fast Path potential FP(P), and the program is defined as program type batch BMPTYPE(Y).
40	PARLIM/SCHDTYPE=SERIAL CONFLICT	The transaction is defined with a PARLIM value other than 65535, and the program is defined as SCHDTYPE(SERIAL).
48	NOT ALLOWED FOR IMS RESOURCE	Command not allowed for IMS descriptors or IMS resources. DFSDSDB1, DFSDSPG1, DBFDSRT1, and DFSDSTR1 are examples of IMS descriptors. DBF#FPU0 is an example of an IMS resource.
59	MSC KEYWORD/FP=E PGM CONFLICT	The transaction is defined as remote, but the program is defined as Fast Path exclusive.
5F	INVALID CHARACTERS IN NAME	Resource or descriptor name contains an invalid character.
67	NO PGM DEFINED	No program is defined. A program name is required for non-remote transaction definitions and routing code definitions. This error should not occur if the definition was created by a valid export or by a utility supplied by IBM.
6A	FP=P/FP=E PGM CONFLICT	The transaction is defined as Fast Path potential FP(P), and the program is defined as Fast Path exclusive FP(E).
6D	INVALID PROGRAM NAME	The program referenced by the transaction or routing code is not defined in the repository for the IMS or being exported for the IMS. A program definition is required for non-remote transaction and routing code definitions.

Table 330. Completion codes for the EXPORT command (continued)

Completion code	Completion code text	Meaning
7A	RTC/FP=N PGM CONFLICT	The program is defined as non-Fast Path FP(N), and a routing code definition that references the program either exists in the repository or is being exported.
90	INTERNAL ERROR	Resource or descriptor definition was not exported because of an internal error.
94	RM REQUEST ERROR	An RM request error occurred. The resource definition was not exported.
B5	ROUTING CODE ALREADY EXISTS	A routing code with the same name as the transaction already exists in the repository for the IMS, but either the program name or the INQ value is different. The transaction definition cannot be exported for the IMS.
121	RESOURCE HAS NO CHANGE	The resource name specified has no changes since the last export.
149	STARTTIME/ENDTIME MISMATCH	The resource name specified does not match the specified STARTTIME or ENDTIME value.
14D	INVALID CHARACTERS IN IMSID	The IMSID value contains an invalid character.
1D0	NOT DONE DUE TO ERROR	An error resulted in the command not being processed for the resource name in the response line. The resource name that had the error will show the error that was encountered. All other resource names will have 1D0 completion code because the export is aborted due to one error. All members are exported as a single unit of work.
1D1	NO REPOSITORY DEFINED	No repository is defined.
1D2	RM VALIDATION ERROR	RM validation error.
1D3	MEMBER NOT FOUND	The specified resource name is not found.
1D4	MEMBER IN USE	The specified resource name is in use.
1D5	NO IMS RESOURCE LIST	No IMS resource list.
1D6	REPOSITORY ERROR	Repository error.
1D7	IMPORT CHANGE LIST ERROR	Resource or descriptor is in the process of being imported from the change list in the IMSRSC repository, or the import from the change list failed. The resource or descriptor cannot be exported until it is successfully imported.

## Examples

The following are examples of the EXPORT command:

### *Example 1 for EXPORT command*

TSO SPOC input:

EXPORT DEFN TARGET(RDDS)

TSO SPOC output:

Log for: EXPORT DEFN TARGET(RDDS)

IMSpIex . . . . . : PLEX1  
Routing . . . . . :  
Start time. . . . . : 2007.074 12:40:02.30  
Stop time . . . . . : 2007.074 12:40:03.32  
Return code . . . . : 00000000  
Reason code . . . . : 00000000  
Reason text . . . . :  
Command master. . : IMS2

MbrName Messages

-----  
IMS1 DFS3404I EXPORT COMMAND COMPLETE FOR ALL , DSN=IMSPLEX1.IMS1.RDDS1.DEFN  
IMS2 DFS3404I EXPORT COMMAND COMPLETE FOR ALL , DSN=IMSPLEX1.IMS2.RDDS3.DEFN

**Explanation:** The EXPORT command is routed to two IMS systems: IMS1 and IMS2. Each IMS exports all of its database, program, routing code, and transaction resource and descriptor definitions to its own system RDDS. Because OPTION(ALLRSP) is not specified, the list of exported resources is not returned.

**Note:** Each DFS3404I message is encapsulated in a single set of <msg> </msg> tags and is shown on the same TSO SPOC screen as the return and reason codes. If the message does not fit on one screen, you might have to scroll to the right to view the entire message.

*Example 2 for EXPORT command*

TSO SPOC input:

EXPORT DEFN TARGET(RDDS) RDDS(SN(IMS1.IMS2.RDDS.DEFN) OPTION(ALLRSP)

TSO SPOC output:

Name	Type	MbrName	CC
DBDFLT	DBDESC	IMS2	0
DEDBJN01	DB	IMS2	0
PGMDFLT	PGMDESC	IMS2	0
CDEBS	PGM	IMS2	0
RTCDFLT	RTCDESC	IMS2	0
SMQFP5A	RTC	IMS2	0
TRANDFLT	TRANDESC	IMS2	0
CDEBTRN1	TRAN	IMS2	0

When the PF4 key is pressed, the following is displayed:

Log for: EXPORT DEFN TARGET(RDDS) RDDS(SN(IMS1.IMS2.RDDS.DEFN) OPTION(ALLRSP)

IMSpIex . . . . . : PLEX1  
Routing . . . . . :  
Start time. . . . . : 2007.074 12:40:02.30  
Stop time . . . . . : 2007.074 12:40:03.32  
Return code . . . . : 00000000  
Reason code . . . . : 00000000  
Reason text . . . . :  
Command master. . : IMS2

MbrName Messages

-----  
IMS2 DFS3404I EXPORT COMMAND COMPLETE FOR ALL , DSN=IMSPLEX1.IMS2.RDDS.DEFN

**Explanation:** The EXPORT command is routed to IMS2. IMS2 exports all of its database, program, routing code, and transaction resource and descriptor definitions to RDDS IMSPLEX1.IMS2.RDDS.DEFN. Because OPTION(ALLRSP) is specified, the list of exported resources is returned.

*Example 3 for EXPORT command*

TSO SPOC input:

```
EXPORT DEFN TARGET(RDDS) TYPE(ALL) NAME(*) OPTION(ALLRSP)
```

TSO SPOC output:

Name	Type	MbrName	CC	CCText
PART	DB	IMS2	0	
PARTMAST	DB	IMS2	0	
PGMD01	PGM	IMS2	0	
RTC001	RTC	IMS2	0	
TRND01	TRAN	IMS2	0	
TRND11	TRAN	IMS2	0	
TRND123	TRAN	IMS2	0	
TRND01	TRANDESC	IMS2	0	

When the PF4 key is pressed, the following is displayed:

```
Log for: EXPORT DEFN TARGET(RDDS) TYPE(ALL) NAME(*) OPTION(ALLRSP)
```

```

IMSpIex . . . . . : PLEX1
Routing . . . . . :
Start time. . . . : 2007.074 12:40:02.30
Stop time . . . . : 2007.074 12:40:03.32
Return code . . . : 0200000C
Reason code . . . : 00003000
Reason text . . . : At least one request successful
Command master. . : IMS2

```

MbrName	Return Code	Reason Code	Description
IMS1	00000010	00004330	No system RDDSs are available
MbrName Messages			
IMS2	DFS3404I	EXPORT COMMAND COMPLETE FOR ALL	, DSN=IMSPLEX1.IMS2.RDDS1.DEFN

**Explanation:** The EXPORT command is routed to two IMS systems: IMS1 and IMS2. IMS1 attempts to export its resource and descriptor definitions to a system RDDS, but none of the system RDDSs are available for export. IMS2 successfully exports its resource and descriptor definitions to RDDS IMSPLEX1.IMS2.RDDS1.DEFN.

*Example 4 for EXPORT command*

TSO SPOC input:

```
EXPORT DEFN TARGET(RDDS) TYPE(DB, DBDESC, TRAN, TRANDESC, PGM, PGMDESC)
NAME(PART*, TRND*, PGMD*) RDDSDSN(IMSPLEX1.IMS2.RDDS.DEFN)
```

TSO SPOC output:

Name	Type	MbrName	CC	CCText
PGMD*		IMS2	10	NO RESOURCES FOUND
PART	DB	IMS2	0	
PARTMAST	DB	IMS2	0	

```

TRND11  TRAN      IMS2      0
TRND120 TRAN      IMS2      0
TRND123 TRAN      IMS2      0
TRND123 TRANDESC IMS2      0

```

When the PF4 key is pressed, the following is displayed:

Log for . . : EXPORT DEFN TARGET(RDDS) TYPE(DB, DBDESC, TRAN, ... More:

```

IMSpIex . . . . . : PLEX1
Routing . . . . . :
Start time . . . . : 2011.124 09:36:06.09
Stop time . . . . . : 2011.124 09:36:06.17
Return code . . . . : 0200000C
Reason code . . . . : 0000300C
Reason text . . . . : The command completed with warning(s).
Command master . . : IMS2

```

MbrName	Return Code	Reason Code	Reason text
IMS2	0000000C	00003000	At least one request successful
IMS1	00000004	00001000	IMS not master, cmd ignored

```

MbrName  Messages
-----  -
IMS2     DFS3404I EXPORT COMMAND COMPLETE FOR SOME, DSN=IMSPLEX1.IMS2.RDDS.DEFN
IMS1     DFS3404I EXPORT COMMAND COMPLETE FOR NONE, DSN=IMSPLEX1.IMS2.RDDS.DEFN

```

OM API input:

```

CMD(EXPORT DEFN TARGET(RDDS) TYPE(DB, DBDESC, TRAN, TRANDESC, PGM, PGMDESC)
NAME(PART*, TRND*, PGMD*) RDDS(SN(IMSPLEX1.IMS2.RDDS.DEFN))

```

OM API output:

```

<imsout>
<ctl>
<omname>OM10M </omname>
<omvsn>1.5.0</omvsn>
<xmlvsn>20 </xmlvsn>
<statime>2011.124 16:36:06.112826</statime>
<stotime>2011.124 16:36:06.171441</stotime>
<staseq>C7B85E23C774F3FA</staseq>
<stoseq>C7B85E23D94B927E</stoseq>
<rqsttkn1>USRT011 10091756</rqsttkn1>
<rc>0200000C</rc>
<rsn>0000300C</rsn>
<rsnmsg>CSLN055I</rsnmsg>
<rsntxt>The command completed with warning(s).</rsntxt>
</ctl>
<cmderr>
<mbr name="IMS2 ">
<typ>IMS </typ>
<styp>DBDC </styp>
<rc>0000000C</rc>
<rsn>00003000</rsn>
<rsntxt>At least one request successful</rsntxt>
</mbr>
<mbr name="IMS1 ">
<typ>IMS </typ>
<styp>DBDC </styp>
<rc>00000004</rc>
<rsn>00001000</rsn>
<rsntxt>IMS not master, cmd ignored</rsntxt>
</mbr>
</cmderr>
<cmd>

```

```

<master>IMS2 </master>
<userid>USRT011 </userid>
<verb>EXP </verb>
<kwd>DEFN </kwd>
<input>EXPORT DEFN TARGET(RDDS) TYPE(DB, DBDESC, TRAN, TRANDESC, PGM,
PGMDESC) NAME(PART*, TRND*, PGMD*) RDDS(SN(IMSplex1.IMS2.RDDS.DEFN)
</input>
</cmd>
<cmdrsphdr>
<hdr s1b1="NAME" l1b1="Name" scope="LCL" sort="a" key="2" scroll="no"
len="8" dtype="CHAR" align="left" />
<hdr s1b1="TYPE" l1b1="Type" scope="LCL" sort="a" key="1" scroll="no"
len="8" dtype="CHAR" skipb="yes" align="left" />
<hdr s1b1="MBR" l1b1="MbrName" scope="LCL" sort="a" key="3" scroll="no"
len="8" dtype="CHAR" align="left" />
<hdr s1b1="CC" l1b1="CC" scope="LCL" sort="n" key="0" scroll="yes"
len="4" dtype="INT" align="right" skipb="no" />
<hdr s1b1="CCTXT" l1b1="CCText" scope="LCL" sort="n" key="0"
scroll="yes" len="*" dtype="CHAR" skipb="yes" align="left" />
<hdr s1b1="ERRT" l1b1="ErrorText" scope="LCL" sort="n" key="0"
scroll="yes" len="*" dtype="CHAR" skipb="yes" align="left" />
</cmdrsphdr>
<cmdrspdata>
<rsp>NAME(PART ) TYPE(DB) MBR(IMS2 ) CC( 0) </rsp>
<rsp>NAME(PARTMAST) TYPE(DB) MBR(IMS2 ) CC( 0) </rsp>
<rsp>NAME(TRND11 ) TYPE(TRAN) MBR(IMS2 ) CC( 0) </rsp>
<rsp>NAME(TRND120 ) TYPE(TRAN) MBR(IMS2 ) CC( 0) </rsp>
<rsp>NAME(TRND123 ) TYPE(TRAN) MBR(IMS2 ) CC( 0) </rsp>
<rsp>NAME(TRND123 ) TYPE(TRANDESC) MBR(IMS2 ) CC( 0) </rsp>
<rsp>NAME(PGMD* ) TYPE() MBR(IMS2 ) CC( 10) CCTXT(NO RESOURCES
FOUND) </rsp>
</cmdrspdata>
<msgdata>
<mbr name="IMS2 ">
<msg>DFS3404I EXPORT COMMAND COMPLETE FOR SOME,
DSN=IMSplex1.IMS2.RDDS.DEFN </msg>
</mbr>
<mbr name="IMS1 ">
<msg>DFS3404I EXPORT COMMAND COMPLETE FOR NONE,
DSN=IMSplex1.IMS2.RDDS.DEFN </msg>
</mbr>
</msgdata>
</imsout>

```

**Explanation:** The EXPORT command is routed to two IMS systems: IMS1 and IMS2. Because the data set is a non-system RDDS, only the IMS that is selected as the command master can process the command. OM selects IMS2 as the command master. IMS2 exports a set of resource and descriptor definitions whose names match the PART\* and TRND\* wildcard values. No resources or descriptors were found matching the PGMD\* wildcard value. IMS1 does not process the command because it is not the command master.

#### *Example 5 for EXPORT command*

TSO SPOC input:

```
EXPORT DEFN TARGET(RDDS) TYPE(ALL) NAME(*)
```

TSO SPOC output:

```

Log for . . . : EXPORT DEFN TARGET(RDDS) TYPE(ALL) NAME(*)

IMSplex . . . . . : PLEX1
Routing . . . . . : IMS1
Start time. . . . : 2011.124 14:22:57.39
Stop time . . . . : 2011.124 14:22:57.41

```

```

Return code . . . : 0200000C
Reason code . . . : 00003004
Reason text . . . : No requests were successful.
Command master. . : IMS1

```

Member Name	Return Code	Reason Code	Reason text
IMS1	00000008	0000211F	No system RDDSs defined, rddsdns() required

OM API input:

```
CMD(EXPORT DEFN TARGET(RDDS) TYPE(ALL) NAME(*))
```

OM API output:

```

<imsout>
<ctl>
<omname>OM10M </omname>
<omvsn>1.5.0</omvsn>
<xmlvsn>20 </xmlvsn>
<statime>2011.124 21:22:57.408936</statime>
<stotime>2011.124 21:22:57.409591</stotime>
<staseq>C7B8A25066BA8BF2</staseq>
<stoseq>C7B8A25066E37072</stoseq>
<rqsttkn1>USRT011 10142257</rqsttkn1>
<rc>0200000C</rc>
<rsn>00003004</rsn>
<rsnmsg>CSLN024I</rsnmsg>
<rsntxt>No requests were successful.</rsntxt>
</ctl>
<cmderr>
<mbr name="IMS1 ">
<typ>IMS </typ>
<styp>DBDC </styp>
<rc>00000008</rc>
<rsn>0000211F</rsn>
<rsntxt>No system RDDSs defined, rddsdns() required</rsntxt>
</mbr>
</cmderr>
<cmd>
<master>IMS1 </master>
<userid>USRT011 </userid>
<verb>EXP </verb>
<kwd>DEFN </kwd>
<input>EXPORT DEFN TARGET(RDDS) TYPE(ALL) NAME(*) </input>
</cmd>
</imsout>

```

**Explanation:** The EXPORT command is sent to IMS1. There are no system RDDSs defined in DFSDFxxx PROCLIB member of IMS1, and the RDDSDSN() keyword was not specified on the EXPORT command. The EXPORT command cannot be processed.

#### Example 6 for EXPORT command

TSO SPOC input:

```
EXPORT DEFN TARGET(REPO) NAME(*) TYPE(ALL) SET(IMSID(*))
```

TSO SPOC output:

```

Log for . . . : EXPORT DEFN TARGET(REPO) NAME(*) TYPE(ALL) SET(I... More: >
IMSpIex . . . . . : PLEX1
Routing . . . . . : IMS1
Start time. . . . : 2011.123 22:16:24.38

```

```

Stop time . . . . : 2011.123 22:16:28.15
Return code . . . : 00000000
Reason code . . . : 00000000
Reason text . . . :
Command master. . : IMS1

```

```

MbrName  Messages
-----

```

```

IMS1      DFS3406I EXPORT COMMAND COMPLETE FOR ALL, REPOTYPE=IMSRSC, REPONAME=IMS_REPOS

```

**OM API input:**

```

CMD(EXPORT DEFN TARGET(REPO) NAME(*) TYPE(ALL) SET(IMSID(*)))

```

**OM API output:**

```

<imsout>
<ctl>
<omname>OM10M </omname>
<omvsn>1.5.0</omvsn>
<xmlvsn>20 </xmlvsn>
<statime>2011.124 05:16:24.395621</statime>
<stotime>2011.124 05:16:28.150725</stotime>
<staseq>C7B7CA45F4365B32</staseq>
<stoseq>C7B7CA4988FC5C3A</stoseq>
<rqsttkn1>USRT011 10221624</rqsttkn1>
<rc>00000000</rc>
<rsn>00000000</rsn>
</ctl>
<cmd>
<master>IMS1 </master>
<userid>USRT011 </userid>
<verb>EXP </verb>
<kwd>DEFN </kwd>
<input>EXPORT DEFN TARGET(REPO) NAME(*) TYPE(ALL) SET(IMSID(*)) </input>
</cmd>
<msgdata>
<mbr name="IMS1 ">
<msg>DFS3406I EXPORT COMMAND COMPLETE FOR ALL, REPOTYPE=IMSRSC,
      REPONAME=IMS_REPOS </msg>
</mbr>
</msgdata>
</imsout>

```

**Explanation:** All the runtime resource definitions for all the database, transaction, program, routing code resources and descriptors from IMS are exported to the repository as the stored resource definitions for all IMS systems in the IMSplex. The command is processed by the command master IMS, IMS1. The IMS resource lists for all IMS systems defined to use the repository are updated. The IMS resource lists that have been updated are also returned on the command response.

**Example 7 for EXPORT command**

**TSO SPOC input:**

```

EXPORT DEFN TARGET(REPO) NAME(TRANA,DBA,RTCA,PGMA) TYPE(ALL) SET(IMSID(IMS1,IMS3))

```

**TSO SPOC output:**

```

Response for: EXPORT DEFN TARGET(REPO) NAME(TRANA,DBA,RTCA,PG...
Name      Type      MbrName  CC
DBA       DB        IMS1     0
IMS1      LIST     IMS1     0
IMS3      LIST     IMS1     0
PGMA      PGM      IMS1     0
RTCA      RTC      IMS1     0
TRANA     TRAN     IMS1     0

```

When the PF4 key is pressed, the following is displayed:

```
Log for . . : EXPORT DEFN TARGET(REPO) NAME(TRANA,DBA,RTCA,PG... More: >
```

```
IMSpIex . . . . . : PLEX1
Routing . . . . . : IMS1
Start time. . . . : 2011.123 22:39:45.76
Stop time . . . . : 2011.123 22:39:45.96
Return code . . . : 00000000
Reason code . . . : 00000000
Reason text . . . :
Command master. . : IMS1
```

```
MbrName Messages
```

```
-----
IMS1 DFS3406I EXPORT COMMAND COMPLETE FOR ALL, REPOTYPE=IMSRSC, REPOName=IMS_REPOS
```

OM API input:

```
CMD(EXPORT DEFN TARGET(REPO) NAME(TRANA,DBA,RTCA,PGMA) TYPE(ALL) SET(IMSID(IMS1,IMS3)))
```

OM API output:

```
<imsout>
<ctl>
<omname>OM10M </omname>
<omvsn>1.5.0</omvsn>
<xmIvsn>20 </xmIvsn>
<statime>2011.124 05:39:45.784253</statime>
<stotime>2011.124 05:39:45.958539</stotime>
<staseq>C7B7CF7E6C1BD337</staseq>
<stoseq>C7B7CF7E96A8B4F7</stoseq>
<rqsttkn1>USRT011 10223945</rqsttkn1>
<rc>00000000</rc>
<rsn>00000000</rsn>
</ctl>
<cmd>
<master>IMS1 </master>
<userid>USRT011 </userid>
<verb>EXP </verb>
<kwd>DEFN </kwd>
<input>EXPORT DEFN TARGET(REPO) NAME(TRANA,DBA,RTCA,PGMA)
TYPE(ALL) SET(IMSID(IMS1,IMS3)) </input>
</cmd>
<cmdrsphdr>
<hdr sIbl="NAME" lIbl="Name" scope="LCL" sort="a" key="2" scroll="no"
len="8" dtype="CHAR" align="left" />
<hdr sIbl="TYPE" lIbl="Type" scope="LCL" sort="a" key="1" scroll="no"
len="8" dtype="CHAR" skipb="yes" align="left" />
<hdr sIbl="Mbr" lIbl="MbrName" scope="LCL" sort="a" key="3" scroll="no"
len="8" dtype="CHAR" align="left" />
<hdr sIbl="CC" lIbl="CC" scope="LCL" sort="n" key="0" scroll="yes"
len="4" dtype="INT" align="right" skipb="no" />
<hdr sIbl="CCTXT" lIbl="CCText" scope="LCL" sort="n" key="0"
scroll="yes" len="*" dtype="CHAR" skipb="yes" align="left" />
<hdr sIbl="ERRT" lIbl="ErrorText" scope="LCL" sort="n" key="0"
scroll="yes" len="*" dtype="CHAR" skipb="yes" align="left" />
</cmdrsphdr>
<cmdrspdata>
<rsp>NAME(DBA ) TYPE(DB) Mbr(IMS1 ) CC( 0) </rsp>
<rsp>NAME(PGMA ) TYPE(PGM) Mbr(IMS1 ) CC( 0) </rsp>
<rsp>NAME(RTCA ) TYPE(RTC) Mbr(IMS1 ) CC( 0) </rsp>
<rsp>NAME(TRANA ) TYPE(TRAN) Mbr(IMS1 ) CC( 0) </rsp>
<rsp>NAME(IMS1 ) TYPE(LIST) Mbr(IMS1 ) CC( 0) </rsp>
<rsp>NAME(IMS3 ) TYPE(LIST) Mbr(IMS1 ) CC( 0) </rsp>
</cmdrspdata>
<msgdata>
<mbr name="IMS1 ">
<msg>DFS3406I EXPORT COMMAND COMPLETE FOR ALL, REPOTYPE=IMSRSC,
```

```

    REPONAME=IMS_REPOS                                </msg>
</mbr>
</msgdata>
</imsout>

```

**Explanation:** The runtime resource definitions for the database named DBA, transaction named TRANA, program named PGMA, and routing code named RTCA are exported to the repository as the stored resource definitions for IMS1 and IMS3.

The command is processed by the command master IMS. In this case, only IMS2 processed the command.

The IMS resource lists for IMS systems IMS1 and IMS3 are updated and are available for those IMS systems to import runtime resource definitions. The IMS resource list for the command master IMS2 is not updated in the repository because it is not specified in the SET(IMSID()) keyword.

This example shows how to use the EXPORT command to update the IMS resource list for another IMS in the IMSplex.

#### *Example 8 for EXPORT command*

TSO SPOC input:

```

EXPORT DEFN TARGET(REPO) NAME(*) TYPE(PGM,PGMDESC,TRAN,TRANDESC)
SET(IMSID(*))

```

TSO SPOC output:

```

Log for . . : EXPORT DEFN TARGET(REPO) NAME(*) TYPE(PGM,PGMDES... More:  +>

```

```

IMSplex . . . . . : PLEX1
Routing . . . . . : IMS2
Start time . . . . : 2011.124 14:05:50.22
Stop time . . . . . : 2011.124 14:05:50.27
Return code . . . . : 0200000C
Reason code . . . . : 00003008
Reason text . . . . : None of the clients were successful.
Command master. . : IMS2

```

MbrName	Return Code	Reason Code	Reason text
IMS2	00000010	0000450B	Repository Server is not available

MbrName Messages

```

-----
IMS2      DFS3407E EXPORT COMMAND COMPLETE FOR NONE, REQUEST=CSLRPUPD,
RC=03000014, RSN=00005508, ERRORTXT=, MBR=RSC, REPOTYPE=IMSRSC, REPONAME=IMS_REPOS

```

OM API input:

```

CMD(EXPORT DEFN TARGET(REPO) NAME(*) TYPE(PGM,PGMDESC,TRAN,TRANDESC)
SET(IMSID(*)))

```

OM API output:

```

<imsout>
<ctl>
<omname>OM10M </omname>
<omvsn>1.5.0</omvsn>
<xmlvsn>20 </xmlvsn>
<statime>2011.124 21:05:50.248438</statime>

```

```

<stotime>2011.124 21:05:50.270743</stotime>
<staseq>C7B89E7CD31F676E</staseq>
<stoseq>C7B89E7CD891703C</stoseq>
<rqsttkn1>USRT011 10140550</rqsttkn1>
<rc>0200000C</rc>
<rsn>00003008</rsn>
<rsnmsg>CSLN054I</rsnmsg>
<rsntxt>None of the clients were successful.</rsntxt>
</ctl>
<cmderr>
<mbr name="IMS2 ">
<typ>IMS </typ>
<styp>DBDC </styp>
<rc>00000010</rc>
<rsn>0000450B</rsn>
<rsntxt>Repository Server is not available</rsntxt>
</mbr>
</cmderr>
<cmd>
<master>IMS2 </master>
<userid>USRT011 </userid>
<verb>EXP </verb>
<kwd>DEFN </kwd>
<input>EXPORT DEFN TARGET(REPO) NAME(*) TYPE(PGM,PGMDESC,TRAN,TRANDESC)
SET(IMSID(*)) </input>
</cmd>
<msgdata>
<mbr name="IMS2 ">
<msg>DFS3407E EXPORT COMMAND COMPLETE FOR NONE, REQUEST=CSLRPUPD,
RC=03000014, RSN=00005508, ERRORTXT=, MBR=RSC, REPOTYPE=IMSRSC,
REPONAME=IMS_REPOS </msg>
</mbr>
</msgdata>
</imsout>

```

**Explanation:** The export of the program and transaction resources and descriptors to the repository failed because the Repository Server was not available. The CSLRPUPD return and reason codes of 03000014/00005508 indicates that the Repository Server is not available. See the CSLRRR macro for a description of the return code and reason code.

### *Example 9 for EXPORT command*

TSO SPOC input:

```
EXPORT DEFN TARGET(REPO) NAME(DBA,PGMA,RTCA,TRANA) TYPE(ALL)
SET(IMSID(*)) OPTION(ALLRSP)
```

TSO SPOC output:

```
Response for: EXPORT DEFN TARGET(REPO) NAME(DBA,PGMA,RTCA,TRAN...
Name      Type      MbrName  CC CText
DBA       DB        IMS2     1D0 NOT DONE DUE TO ERROR
IMS1      LIST     IMS2     1D0 NOT DONE DUE TO ERROR
IMS2      LIST     IMS2     1D0 NOT DONE DUE TO ERROR
IMS3      LIST     IMS2     1D0 NOT DONE DUE TO ERROR
PGMA      PGM      IMS2     1D0 NOT DONE DUE TO ERROR
RTCA      RTC      IMS2     1D0 NOT DONE DUE TO ERROR
TRANA     TRAN     IMS2     67 NO PGM DEFINED
```

When the PF4 key is pressed, the following is displayed:

```
Log for . . : EXPORT DEFN TARGET(REPO) NAME(DBA,PGMA,RTCA,TRAN... More:  +>

IMSpIex . . . . . : PLEX1
Routing . . . . . : IMS2
```

```

Start time. . . . : 2011.124 11:32:16.90
Stop time . . . . : 2011.124 11:32:17.19
Return code . . . : 0200000C
Reason code . . . : 00003008
Reason text . . . : None of the clients were successful.
Command master. . : IMS2

```

Member Name	Return Code	Reason Code	Reason text
IMS2	0000000C	00003004	No requests were successful

```

MbrName Messages
-----
IMS2 DFS3407E EXPORT COMMAND COMPLETE FOR NONE, REQUEST=CSLRPUPD,
RC=0300000C, RSN=00003004, ERRORTXT=00000000 03000008 00002514 00000000
00000000 00000000, MBR=RSC, REPOTYPE=IMSRSC, REPONAME=IMS_REPOS

```

### OM API input:

```

CMD(EXPORT DEFN TARGET(REPO) NAME(DBA,PGMA,RTCA,TRANA) TYPE(ALL)
SET(IMSID(*)) OPTION(ALLRSP))

```

### OM API output:

```

<imsout>
<ctl>
<omname>OM10M </omname>
<omvsn>1.5.0</omvsn>
<xm1vsn>20 </xm1vsn>
<statime>2011.124 18:32:16.943085</statime>
<stotime>2011.124 18:32:17.187129</stotime>
<staseq>C7B87C2A54FEDBEC</staseq>
<stoseq>C7B87C2A909390F0</stoseq>
<rqsttkn1>USRT011 10113216</rqsttkn1>
<rc>0200000C</rc>
<rsn>00003008</rsn>
<rsnmsg>CSLN054I</rsnmsg>
<rsntxt>None of the clients were successful.</rsntxt>
</ctl>
<cmderr>
<mbr name="IMS2 ">
<typ>IMS </typ>
<styp>DBDC </styp>
<rc>0000000C</rc>
<rsn>00003004</rsn>
<rsntxt>No requests were successful</rsntxt>
</mbr>
</cmderr>
<cmd>
<master>IMS2 </master>
<userid>USRT011 </userid>
<verb>EXP </verb>
<kwd>DEFN </kwd>
<input>EXPORT DEFN TARGET(REPO) NAME(DBA,PGMA,RTCA,TRANA)
TYPE(ALL)SET(IMSID(*)) OPTION(ALLRSP) </input>
</cmd>
<cmdrsphdr>
<hdr s1b1="NAME" l1b1="Name" scope="LCL" sort="a" key="2" scroll="no"
len="8" dtype="CHAR" align="left" />
<hdr s1b1="TYPE" l1b1="Type" scope="LCL" sort="a" key="1" scroll="no"
len="8" dtype="CHAR" skipb="yes" align="left" />
<hdr s1b1="MBR" l1b1="MbrName" scope="LCL" sort="a" key="3" scroll="no"
len="8" dtype="CHAR" align="left" />
<hdr s1b1="CC" l1b1="CC" scope="LCL" sort="n" key="0" scroll="yes"
len="4" dtype="INT" align="right" skipb="no" />
<hdr s1b1="CCTXT" l1b1="CCText" scope="LCL" sort="n" key="0"
scroll="yes" len="*" dtype="CHAR" skipb="yes" align="left" />
<hdr s1b1="ERRT" l1b1="ErrorText" scope="LCL" sort="n" key="0"
scroll="yes" len="*" dtype="CHAR" skipb="yes" align="left" />
</cmdrsphdr>
<cmdrspdata>
<rsp>NAME(DBA ) TYPE(DB) MBR(IMS2 ) CC( 1D0) CCTXT(NOT DONE DUE TO ERROR) </rsp>
<rsp>NAME(PGMA ) TYPE(PGM) MBR(IMS2 ) CC( 1D0) CCTXT(NOT DONE DUE TO ERROR) </rsp>
<rsp>NAME(RTCA ) TYPE(RTC) MBR(IMS2 ) CC( 1D0) CCTXT(NOT DONE DUE TO ERROR) </rsp>
<rsp>NAME(TRANA ) TYPE(TRAN) MBR(IMS2 ) CC( 67) CCTXT(NO PGM

```

```

DEFINED) </rsp>
<rsp>NAME(IMS1 ) TYPE(LIST) MBR(IMS2 ) CC( 1D0) CCTXT(NOT DONE DUE TO ERROR) </rsp>
<rsp>NAME(IMS2 ) TYPE(LIST) MBR(IMS2 ) CC( 1D0) CCTXT(NOT DONE DUE TO ERROR) </rsp>
<rsp>NAME(IMS3 ) TYPE(LIST) MBR(IMS2 ) CC( 1D0) CCTXT(NOT DONE DUE TO ERROR)</rsp>
</cmdrspdata>
<msgdata>
<mbr name="IMS2 ">
<msg>DFS3407E EXPORT COMMAND COMPLETE FOR NONE, REQUEST=CSLRPUPD,
RC=0300000C, RSN=00003004, ERRORTXT=00000000 03000008 00002514
00000000 00000000 00000000, MBR=RSC, REPOTYPE=IMSRSC,
REPONAME=IMS_REPOS </msg>
</mbr>
</msgdata>
</imsout>

```

**Explanation:** The export of the program, transaction, database, and routing code resource definitions to the repository failed because the definition for the program referenced by TRANA does not exist in the repository and is not being exported. A completion code of 1D0, Not done due to error, is returned for all members being updated in the repository except the one that failed. If the error is due to a resource, MBR=RSC is set in the DFS3407E message.

### Example 10 for EXPORT command

TSO SPOC input:

```

EXPORT DEFN TARGET(REPO) NAME(DBA,PGMA,RTCA,TRANA) TYPE(ALL)
SET(IMSID(IM1*)) OPTION(ALLRSP)

```

TSO SPOC output:

```

Response for: EXPORT DEFN TARGET(REPO) NAME(DBA,PGMA,RTCA,TRAN...
Name      Type      MbrName   CC CCText
DBA       DB         IMS2      1D0 NOT DONE DUE TO ERROR
IM1*     LIST      IMS2      1D5 NO IMS RESOURCE LIST
PGMA     PGM       IMS2      1D0 NOT DONE DUE TO ERROR
RTCA     RTC       IMS2      1D0 NOT DONE DUE TO ERROR
TRANA    TRAN      IMS2      1D0 NOT DONE DUE TO ERROR

```

When the PF4 key is pressed, the following is displayed:

```

Log for . . : EXPORT DEFN TARGET(REPO) NAME(DBA,PGMA,RTCA,TRAN... More: +>

```

```

IMSpIex . . . . . : PLEX1
Routing . . . . . : IMS2
Start time. . . . : 2011.124 12:21:47.67
Stop time . . . . : 2011.124 12:21:47.76
Return code . . . : 0200000C
Reason code . . . : 00003008
Reason text . . . : None of the clients were successful.
Command master. . : IMS2

```

MbrName	Return Code	Reason Code	Reason text
IMS2	0000000C	00003004	No requests were successful

MbrName Messages

```

-----
IMS2 DFS3407E EXPORT COMMAND COMPLETE FOR NONE, REQUEST=CSLRPUPD,
RC=0300000C, RSN=00003004, ERRORTXT=, MBR=LST, REPOTYPE=IMSRSC,
REPONAME=IMS_REPOS

```

OM API input:

```

CMD(EXPORT DEFN TARGET(REPO) NAME(DBA,PGMA,RTCA,TRANA) TYPE(ALL)
SET(IMSID(IM1*)) OPTION(ALLRSP))

```

## OM API output:

```
<imsout>
<ctl>
<omname>OM10M </omname>
<omvsn>1.5.0</omvsn>
<xmlvsn>20 </xmlvsn>
<statime>2011.124 19:21:47.695676</statime>
<stotime>2011.124 19:21:47.758331</stotime>
<staseq>C7B8873B7663C765</staseq>
<stoseq>C7B8873B85AFBBEA</stoseq>
<rqsttkn1>USRT011 10122147</rqsttkn1>
<rc>0200000C</rc>
<rsn>00003008</rsn>
<rsnmsg>CSLN054I</rsnmsg>
<rsntxt>None of the clients were successful.</rsntxt>
</ctl>
<cmderr>
<mbr name="IMS2 ">
<typ>IMS </typ>
<styp>DBDC </styp>
<rc>0000000C</rc>
<rsn>00003004</rsn>
<rsntxt>No requests were successful</rsntxt>
</mbr>
</cmderr>
<cmd>
<master>IMS2 </master>
<userid>USRT011 </userid>
<verb>EXP </verb>
<kwd>DEFN </kwd>
<input>EXPORT DEFN TARGET(REPO) NAME(DBA,PGMA,RTCA,TRANA)
TYPE(ALL)SET(IMSID(IM1*)) OPTION(ALLRSP) </input>
</cmd>
<cmdrsphdr>
<hdr s1b1="NAME" l1b1="Name" scope="LCL" sort="a" key="2" scroll="no"
len="8" dtype="CHAR" align="left" />
<hdr s1b1="TYPE" l1b1="Type" scope="LCL" sort="a" key="1" scroll="no"
len="8" dtype="CHAR" skipb="yes" align="left" />
<hdr s1b1="MBR" l1b1="MbrName" scope="LCL" sort="a" key="3" scroll="no"
len="8" dtype="CHAR" align="left" />
<hdr s1b1="CC" l1b1="CC" scope="LCL" sort="n" key="0" scroll="yes"
len="4" dtype="INT" align="right" skipb="no" />
<hdr s1b1="CCTXT" l1b1="CCText" scope="LCL" sort="n" key="0"
scroll="yes" len="*" dtype="CHAR" skipb="yes" align="left" />
<hdr s1b1="ERRT" l1b1="ErrorText" scope="LCL" sort="n" key="0"
scroll="yes" len="*" dtype="CHAR" skipb="yes" align="left" />
</cmdrsphdr>
<cmdrspdata>
<rsp>NAME(DBA ) TYPE(DB) MBR(IMS2 ) CC( 1D0) CCTXT(NOT DONE DUE TO ERROR) </rsp>
<rsp>NAME(PGMA ) TYPE(PGM) MBR(IMS2 ) CC( 1D0) CCTXT(NOT DONE DUE TO ERROR) </rsp>
<rsp>NAME(RTCA ) TYPE(RTC) MBR(IMS2 ) CC( 1D0) CCTXT(NOT DONE DUE TO ERROR) </rsp>
<rsp>NAME(TRANA ) TYPE(TRAN) MBR(IMS2 ) CC( 1D0) CCTXT(NOT DONE DUE TO ERROR) </rsp>
<rsp>NAME(IM1* ) TYPE(LIST) MBR(IMS2 ) CC( 1D5) CCTXT(NO IMS
RESOURCE LIST) </rsp>
</cmdrspdata>
<msgdata>
<mbr name="IMS2 ">
<msg>DFS3407E EXPORT COMMAND COMPLETE FOR NONE, REQUEST=CSLRPUPD,
RC=0300000C, RSN=00003004, ERRORTXT=, MBR=LST, REPOTYPE=IMSRSC,
REPONAME=IMS_REPOS </msg>
</mbr>
</msgdata>
</imsout>
```

**Explanation:** The export of the program, transaction, database, and routing code resource definitions to the repository failed because there are no IMS resource lists in the repository the match the wildcard value IM1\*. A completion code of 1D0, Not done due to error, is returned for all the members being updated in the repository except the one that failed. MBR=LST indicates that the error is for an IMS resource list.

### Related concepts:

 [How to interpret CSL request return and reason codes \(System Programming APIs\)](#)

➤ CSL RM, IMS, and Repository Server termination (System Administration)

➤ IMSRSC repository administration (System Administration)

➤ Resource definition data sets (System Definition)

**Related reference:**

Chapter 2, “Command keywords and their synonyms,” on page 63



---

## Chapter 32. /FORMAT command

Use the /FORMAT command to cause a specific format to be displayed on a physical terminal using the IMS Message Format Service (MFS).

The displayed format is used by IMS in processing the next input from the formatted terminal.

Subsections:

- "Environment"
- "Syntax"
- "Keywords"
- "Usage notes" on page 820
- "Example" on page 820

### Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) from which the command and keyword can be issued.

Table 331. Valid environments for the /FORMAT command and keyword

Command / Keyword	DB/DC	DBCTL	DCCTL
/FORMAT	X		X
LTERM	X		X

### Syntax



### Keywords

The following keywords are valid for the /FORMAT command:

#### **modname**

Specifies the name of the message output descriptor (MOD) to be used to format the terminal.

#### **LTERM**

Specifies the name of the logical terminal to be formatted. LTERM is required when a terminal other than the input terminal is to be formatted. If LTERM is not specified, the input physical terminal must have at least one LTERM assigned.

If the *ltermname* does not exist, an attempt is made to create the *lterm* and associated user structure. The command creates this structure by queuing the format and data, if any exists, to the terminal as if it were a message. This process is similar to a message switch, a message insert, or a /BROADCAST LTERM, all of which queue a message to an LTERM.

The format request is rejected if the specified LTERM is:

- Assigned to a physical terminal that is not supported by MFS
- In line-response mode or exclusive mode
- Involved in an active conversation
- An input-only terminal
- Not the alternate master in an XRF system
- A remote LTERM

In an IMSplex, /FORMAT LTERM can be used to queue a format to an LTERM that is not active or that is active on any IMS in the IMSplex.

#### ***data***

Is a string consisting of 1 to 8 characters to be inserted into the output message created for the terminal being formatted. Since MFS treats the data as application program output data, the data string can be used to select a logical page (LPAGE) within the requested MOD. The way the selection is performed is determined by the requested MOD's LPAGE definitions.

### **Usage notes**

Using a password with the /FORMAT command is not valid.

The /FORMAT command, where an IMS-supplied default *modname* is used, is not effective for a 3270 master terminal that uses the MFS master terminal formatting option.

**Recommendation:** If a format is lost during a conversation, do not use the /FORMAT command to restore the format. Instead, enter a /HOLD command followed by a /RELEASE command to display the first physical page of the current message and restore the format.

### **Example**

Remote Terminal entry:

```
/FORMAT DFSMO4
/BROADCAST LTERM WTOR (eos)
this is segment 1 (eos)
this is segment 2 (eos)
this is segment 3 (eom)
```

Response ET:

```
DFS058I BROADCAST COMMAND COMPLETED
```

Response RT:

```
THIS IS SEGMENT 1
THIS IS SEGMENT 2
THIS IS SEGMENT 3
```

Explanation: The remote terminal is first formatted by the /FORMAT command, where default format DFSMO4 supports the input of 4 segments. This is followed by /BROADCAST with four segments.

---

## Chapter 33. /HOLD command

Use the /HOLD command to suspend and save a currently active conversation.

Subsections:

- “Environment”
- “Syntax”
- “Usage notes”
- “Example”

### Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) from which the command can be issued.

Table 332. Valid environments for the /HOLD command

Command	DB/DC	DBCTL	DCCTL
/HOLD	X		X

### Syntax



### Usage notes

The IMS response to /HOLD provides a conversation identification that must be used later when the held conversation is resumed (/RELEASE command) or terminated (/EXIT).

In an IMSplex, if global resource information is kept in Resource Manager, /HOLD suspends and saves the active conversation globally in Resource Manager. If global resource information is not kept in Resource Manager, /HOLD suspends and saves the active conversation locally.

### Example

Entry ET:

```
/HOLD
```

Response ET:

```
DFS999I HELD CONVERSATION ID IS 0001
```

Explanation: The active conversation is saved and is assigned an identification of 0001.



## Chapter 34. /IDLE command

The /IDLE command is used to immediately terminate input from, or output to, physical terminals assigned to the lines, logical links, or nodes specified in the command.

Partially processed input messages on the lines specified in the command are discarded. Output messages being sent are returned to the message queues for later transmission

This command can be issued to an IMSplex using the Batch SPOC utility.

Subsections:

- "Environment"
- "Syntax"
- "Keywords" on page 824
- "Examples" on page 826

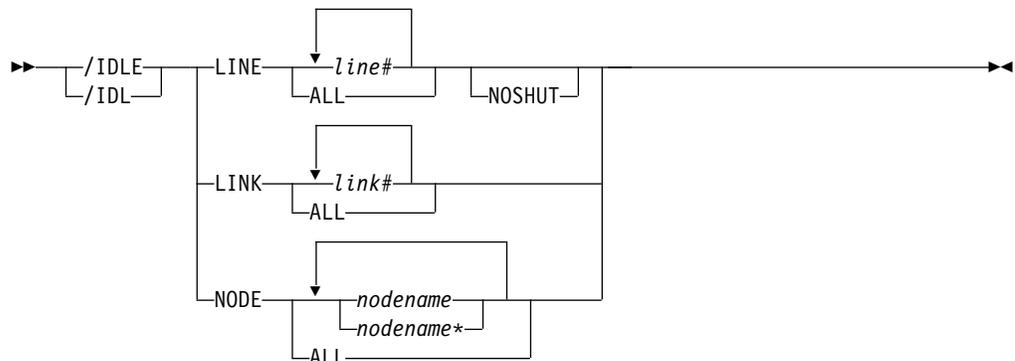
### Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

Table 333. Valid environments for the /IDLE command and keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
/IDLE	X		X
LINE	X		X
LINK	X		X
NODE	X		X
NOSHUT	X		X

### Syntax



## Keywords

The following keywords are valid for the /IDLE command:

### LINE

Specifies the communication line to be immediately terminated.

/IDLE does not apply to the system console line, the master terminal line, or the secondary master terminal line. If these lines are specified, a DFS058 COMMAND COMPLETED EXCEPT LINE X message is returned. If the master terminal is assigned to a line that must be idle, the master terminal must be reassigned to a different line before /IDLE is specified.

### NOSHUT

Immediately terminates input to and output from 3270 remote non-VTAM lines without a checkpoint shutdown. /IDLE cannot be entered before a checkpoint unless the NOSHUT keyword is included in the command. NOSHUT is only valid if IMS is not in the process of shutting down and is mutually exclusive with the NODE parameter. When /IDLE LINE line# NOSHUT is entered, the 3270 remote non-VTAM lines must be stopped or process stopped. A DFS058 COMMAND COMPLETED EXCEPT LINE X message is returned if the command is entered and one of the following conditions exists:

- The line is the master terminal line or system console line, including the secondary master terminal line.
- The line is not stopped or process stopped.
- The line is not a 3270 remote non-VTAM line.
- The line is not open.
- The line is deactivated by IMS.

If the line specified with the NOSHUT keyword belongs to a line group, the input and output for all lines within the group are immediately terminated. If the line belonging to the line group has the master terminal assigned to it, the /IDLE LINE line# NOSHUT command is rejected.

To restart the lines idled by the /IDLE LINE line# NOSHUT command, either the /START LINE or /RSTART LINE command must be entered.

### LINK

Specifies the logical link from which input, or to which output, is to be terminated. (No input messages are lost with /IDLE LINK.)

If a checkpoint shutdown is not in progress, only MSC VTAM links are valid. All non-MSC VTAM links require a shutdown checkpoint in progress. IMS returns a DFS058 IDLE COMMAND COMPLETED EXCEPT LINK X message if one of the following conditions exists:

- The link is idled already.
- The link is not connected.
- The link cannot be opened.
- A shutdown checkpoint is not in progress (for non-MSC VTAM links).

For MSC TCP/IP logical links that are active, the /IDLE LINK command immediately terminates the link. The /IDLE LINK command is similar to the /PSTOP LINK FORCE command because the /PSTOP LINK FORCE command also immediately terminates a TCP/IP logical link that is active. However, the /IDLE LINK command is processed only during IMS shutdown, whereas the /PSTOP LINK FORCE can be processed at any time.

IMS performs the following actions when it processes the /IDLE LINK command for an active TCP/IP link:

1. Terminates the link in the IMS system where the command is issued.
2. Notifies the local IMS Connect instance to clean up the send socket.
3. Issues the DFS2169I DISCONNECTION COMPLETED ON LINK informational message.
4. Places the link in PSTOPPED ERE IDLE status.

An /IDLE LINK link# command internally causes issuing of the VTAM command:

```
VARY NET,TERM,TYPE=FORCE,NOTIFY=NO,SCOPE=ALL,LU1=xxxxxxxx,LU2=yyyyyyyy
```

All logical links, assigned to the same physical link, associated with the named logical link (link#) will be stopped. That is, all parallel sessions on the same physical link will be stopped. The /RSTART LINK command is used to reactivate links or sessions.

The /IDLE LINK link# command should not be used as the first attempt to stop the link. If the MSC link appears to be hung, the IMS operator should:

1. First attempt to stop the link with the IMS /PSTOP command.
2. If this fails, display the link session status through VTAM, using the VTAM command:

```
D NET,ID=xxxxxxxx
```

or the VTAM command:

```
D NET,SESSIONS,LIST=ALL
```

3. If the link session is still active (STATUS=ACTIV), terminate it with the /IDLE LINK command or the VTAM command:

```
VARY NET,TERM,TYPE=FORCE,LU1=xxxxxxxx
```

**Note:** All parallel sessions associated with the link will be terminated.

## NODE

Specifies the VTAM node to which output is to be terminated, or from which input is to be terminated. The master terminal only can be idled when specifically named.

If the /IDLE NODE *nodename* command is entered for an ISC node defined with users, all half-sessions of the specified node are affected.

An /IDLE NODE ALL command causes the VTAM command VARY NET,INACT, to be issued to all nodes still connected to IMS, except the master terminal.

If the /IDLE NODE command is entered prior to IMS entering shutdown, the /ACT command can be entered to reactivate the node in VTAM. If the /IDLE NODE command is entered while in IMS shutdown, a VARY NET,ACT command might have to be entered directly to VTAM. However, before using the /IDLE command, a /CLSDST and then a /CLSDST FORCE command should be tried.

The node parameter can be generic. The generic parameter specifies nodes that already exist.

## Examples

The following are examples of the /IDLE command:

### *Example 1 for /IDLE command*

Entry ET:

```
/IDLE LINE 14 24
```

Response ET:

```
DFS058I IDLE COMMAND COMPLETED
```

Explanation: The I/O on lines 14 and 24 is forced to terminate.

### *Example 2 for /IDLE command*

Entry ET:

```
/IDLE LINE 4 NOSHUT
```

Response ET:

```
DFS058I IDLE COMMAND COMPLETED
```

Explanation: The I/O on line 4, which is a 3270 remote non-VTAM line, is forced to terminate.

### *Example 3 for /IDLE command*

Entry ET:

```
/IDLE LINK 2
```

Response ET:

```
DFS058I IDLE COMMAND COMPLETED
```

Explanation: The input/output on logical link 2 is forced to terminate.

### *Example 4 for /IDLE command*

Entry ET:

```
/IDLE NODE ABC
```

Response ET:

```
DFS058I IDLE COMMAND COMPLETED
```

Explanation: A VARY NET, INACT command is executed that will cause the IMS VTAM LOSTERM exit to be entered with an indication that the session has been terminated.

---

## Chapter 35. IMPORT commands

Use the IMS IMPORT commands to import resources and resource descriptors.

These commands can be issued through TSO SPOC, IBM Management Console for IMS and DB2 for z/OS, the Manage Resources options in the IMS Applications menu, or other user-written programs that issue commands through OM API. These commands can also be issued to an IMSplex by using the Batch SPOC utility.

Subsections:

- “CREATE DB command” on page 177
- “CREATE DBDESC command” on page 185
- “CREATE IMSCON commands” on page 191
- “CREATE OTMADESC command” on page 250
- “CREATE PGM command” on page 260
- “CREATE PGMDESC command” on page 271
- “CREATE RTC command” on page 281
- “CREATE RTCDESC command” on page 287
- “CREATE TRAN command” on page 292
- “CREATE TRANDESC command” on page 316

---

### IMPORT DEFN SOURCE(CATALOG) command

When the IMS management of ACBs is enabled, use the IMPORT DEFN SOURCE(CATALOG) command to activate application and database resource definitions (PSBs and DBDs) in an online IMS system by importing them from the staging data set of the IMS catalog.

The IMS management of ACBs must be enabled before you can use the IMPORT DEFN SOURCE(CATALOG) command. If the IMS management of ACBs is not enabled, IMPORT DEFN SOURCE(CATALOG) is rejected. For more information about the IMS management of ACBs, see IMS management of ACBs (System Definition).

If you are modifying certain types of PSBs in an IMSplex that uses a shared IMS catalog, you can activate the modified PSBs in a subset of IMS systems by using the UPDATEPSB and REFRESHPSB options. You can then test the application changes in the subset before you activate the changes across the entire IMSplex.

The IMPORT command can be issued through TSO SPOC, IBM Management Console for IMS and DB2 for z/OS, the Manage Resources options in the IMS Applications menu, or any user-written programs that issue commands through OM API. These commands can also be issued to an IMSplex by using the Batch SPOC utility.

Subsections:

- “Environment” on page 828
- “Syntax” on page 828
- “Keywords” on page 828

- “Usage notes” on page 830
- “Output fields” on page 832
- “Return and reason codes” on page 834
- “Completion codes” on page 836
- “Examples” on page 844

## Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

Table 334. Valid environments for the IMPORT command and keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
IMPORT DEFN	X	X	X
OPTION	X	X	X
SOURCE(CATALOG)	X	X	

## Syntax

**A:**

## Keywords

The following keywords are valid for the IMPORT command:

### DEFN

A required keyword.

### FOR()

When the OPTION() keyword is specified, specifies which IMS systems in the IMS sysplex are to participate in the IMPORT process. Depending on the type of resources that are being imported and the state of processing in each IMS system, the options that are specified on the OPTION() keyword might impact systems that are not listed in the FOR(IMSID()) keyword. For more information, see the description of the OPTIONS() keyword.

### IMSID()

Specify one or more comma-separated IMS IDs of the IMS systems that are to process the IMPORT command and the specified option.

The IMS ID specified on the FOR(IMSID()) keyword must be an IMS that exists within the IMSplex. The IMS specified might be active or down when the command is issued.

**Note:** Depending on the value that is specified on the OPTION() keyword, the IMS systems that are listed on the IMSID() keyword must either include or exclude the command master.

IMS selects a command master from the IMS systems that the command is routed to when the command is submitted. Therefore, to include the command master, route the command to one or more of the IMS systems that are specified on the IMSID() keyword. To exclude the command master, route the command to one or more IMS systems that are not specified on the IMSID() keyword.

When OPTION(UPDATEPSB) is specified, one of the IMS systems that is specified on the IMSID() keyword must be the command master, so route the IMPORT DEFN command to only the specified IMS systems.

When OPTION(REFRESHPSB) is specified, the command master cannot be an IMS system that is specified on the IMSID() keyword, so route the IMPORT DEFN command to one or more IMS systems that are not specified on the IMSID() keyword.

When either of the preceding options is specified, if the command is routed to all IMS systems in the IMSplex, either explicitly or by default, the success of the command is unpredictable because the IMS system that is selected as the command master is unpredictable.

To see a list of the IMS systems in your IMSplex, issue the QUERY IMSPLEX TYPE(IMS) SHOW(ALL) command.

#### **OPTION()**

Specifies optional functions of the command.

You can specify the following options:

##### **DELPENDERR**

Specifies that if errors occur during the prepare phase of IMPORT DEFN SOURCE(CATALOG) command processing, the pending resources are deleted from the staging data set. An error in the prepare phase is indicated by OLCPREPF in the IMPORT DEFN command output.

##### **REFRESHPSB**

Causes the IMS systems that are listed in the FOR(IMSID()) option to reload from the IMS catalog directory data set all PSBs in the system that have a *refresh-pending* (REFP) status.

Specify the REFRESHPSB option to activate PSB changes in an IMSplex that were previously activated by the UPDATEPSB option in only a subset of the IMS systems in the IMSplex.

Program resources that might be affected by the changes must be stopped on the systems that are listed in the FOR() option.

PSB activation does not occur for systems that are not listed in the FOR(IMSID()) clause.

When OPTION(REFRESHPSB) is specified, the command master cannot be an IMS system that is specified on the IMSID() keyword, so route the IMPORT DEFN command to one or more IMS systems that are not specified on the IMSID() keyword.

To see a list of the IMS systems in your IMSplex, issue the QUERY IMSPLEX TYPE(IMS) SHOW(ALL) command.

While the REFRESHPSB option is being processed by an IMS system, the system status includes RFSHPSB in the Lc1Stat field of the QUERY MEMBER command output.

#### **UPDATEPSB**

In an IMSplex, activates PSB changes in the IMS catalog and causes the IMS systems that are listed in the FOR(IMSID()) option to reload the changed PSBs the next time the PSBs are scheduled. IMS systems that are omitted from the FOR(IMSID()) keyword are not notified by the command to reload the modified PSBs and continue to use the original PSB definition in online memory until the REFRESHPSB option is submitted or another event, like a system restart, causes the PSB to be reloaded.

If the staging data set of the IMS catalog contains any DBDs when the UPDATEPSB option is specified, the IMPORT command fails.

The FOR(IMSID()) keyword is not effective for DOPT PSBs. After you issue the UPDATEPSB option, all IMS system that schedule the PSB in the IMSplex load the new, modified PSB from the IMS directory.

The FOR(IMSID()) keyword is effective for non-resident PSBs only if the PSB is cached in the 64-bit storage pools of the IMS systems that are **not** specified on the FOR(IMSID()) keyword.

Program resources that might be affected by the changes must be stopped on the systems that are listed in the FOR(IMSID()) option. Stopping the resources is not necessary for systems that are not listed in the FOR(IMSID()) option, unless the PSB is a DOPT PSB or a non-resident PSB that is not cached in the 64-bit storage pool of the systems.

All pending PSB changes in the staging data set are activated in the IMS systems that are listed in the FOR(IMSID()) clause when the IMPORT command is issued with the UPDATEPSB option.

When OPTION(UPDATEPSB) is specified, one of the IMS systems that is specified on the IMSID() keyword must be the command master, so route the IMPORT DEFN command to only the specified IMS systems.

To see a list of the IMS systems in your IMSplex, issue the QUERY IMSPLEX TYPE(IMS) SHOW(ALL) command.

While OPTION(UPDATEPSB) is being processed by an IMS system, the system status includes UPDPBSB in the Lc1Stat field of the QUERY MEMBER command output.

#### **SOURCE()**

Specifies the source of the resource definitions to be imported.

#### **CATALOG**

A required parameter that indicates that the resources to be activated are to be imported into the online IMS systems from the staging data set of the IMS catalog.

### **Usage notes**

Before you issue the IMPORT DEFN SOURCE(CATALOG) command, stop all existing database and program resources that will be affected by the changes.

After the command processing completes, restart all databases and programs that were stopped for the change.

When the IMPORT DEFN SOURCE(CATALOG) command is issued, IMS takes the following actions:

1. Adds or replaces the ACBs in the directory data set of the IMS catalog.
2. Removes the ACBs from the staging data set of the IMS catalog.
3. Removes all old copies of the resources from memory and casts out DMBs and PSBs from their respective pools.
4. If the IMS catalog is shared by multiple IMS systems, coordinates the change across the IMSplex.

When you issue the IMPORT DEFN SOURCE(CATALOG) command, IMS uses an internal process that is similar to the online change function to activate the resources in online IMS systems. Consequently, if you issue the QUERY MEMBER command during import processing, the status reflects the same prepare and commit phases that online change uses. If errors occur, the messages might also reflect these phases.

The IMPORT DEFN SOURCE(CATALOG) command can be issued only through the Operations Manager (OM) API. This command applies to DB/DC and DBCTL systems in which the IMS management of ACBs is enabled by ACBMGMT=CATALOG in the CATALOG section of the DFSDFxxx member.

The IMPORT command is not valid on the XRF alternate, RSR tracker, or FDBR region.

The IMPORT DEFN SOURCE(CATALOG) command is valid for the following resource types:

- Database descriptors (DBDs), program views (PSBs), and their associated application control blocks (ACBs)

By default, when the IMPORT command is issued, OM routes the command to all active IMS systems (or to all IMS systems that are specified in the route list if the ROUTE parameter is specified on the CSLOMCMMD or CSLOMI request) that registered interest in the command. All IMS systems that receive the command attempt to process the IMPORT command and import the resource definitions from the specified source. If you need the IMPORT command to be processed by a specific IMS, the ROUTE parameter must be used to direct the IMPORT command to the specific IMS.

The value of the ACBSHR= parameter that is in effect on the command master IMS system can affect whether the other IMS systems in the IMSplex process an IMPORT DEFN SOURCE(CATALOG) command.

- If ACBSHR=Y is specified in the command master, the IMPORT DEFN SOURCE(CATALOG) command is processed by all IMS systems in the IMSplex that specify ACBSHR=Y and any IMS systems that specify ACBSHR=N, if they are specified on the ROUTE parameter when the command is issued. For IMS systems that specify ACBSHR=N, the IMPORT command succeeds only if the IMS systems have pending ACB changes in the IMS catalog instance that they use.
- If ACBSHR=N is specified on the command master, the command is processed by the command master and by any IMS systems that also specify ACBSHR=N and are specified on the ROUTE parameter. The command is rejected by IMS systems that specify ACBSHR=Y.

The IMS system that serves as the command master is selected by IMS from the IMS systems to which the IMPORT DEFN command was routed.

The UPDATEPSB and REFRESHPSB options each require you to route the command differently. When the UPDATEPSB option is specified, you must route the command to one or more of the IMS systems that are specified on the FOR(IMSID()) keyword. When the REFRESHPSB option is specified, you must route the command to one or more IMS systems that are **not** specified on the FOR(IMSID()) keyword.

The UPDATEPSB option is for use with PSBs that IMS systems store in online memory, such as PSBs defined as resident and non-resident PSBs that have been cached in a 64-bit storage pool. For dynamic option (DOPT) PSBs and for non-resident PSBs that are not cached in an online 64-bit storage pool of an IMS system, the effect of the UPDATEPSB option is no different than issuing the IMPORT DEFN SOURCE(CATALOG) command without OPTION(UPDATEPSB) specified.

The IMS systems that are not specified in the FOR(IMSID()) keyword when the UPDATEPSB option is specified do not process the IMPORT command so do not reload their in-memory PSB definitions after the modified PSB definitions are imported into the IMS directory. However, after command processing completes, if an IMS system that was not specified on the command is required to reload its in-memory PSB definition from the IMS directory for any other reason, the modified PSB definition will become the active PSB definition in that IMS system.

Usually, the IMS systems that are excluded from the UPDATEPSB option continue to use the original, in-memory PSB definitions until either the IMPORT DEFN SOURCE(CATALOG) OPTION(REFRESHPSB) command is issued for the IMS system or the IMS system is restarted. In-memory PSB definitions that are prevented from reloading when the UPDATEPSB option is specified have a *refresh-pending* (REFP) status in the IMPORT command response.

To activate the PSB modifications in an IMS system and remove the PSBs refresh-pending status, issue the IMPORT DEFN SOURCE(CATALOG) OPTION(REFRESHPSB) command with the ID of the IMS system specified on the FOR(IMSID()) keyword. The REFRESHPSB option reloads all PSB definitions that have a refresh-pending status in the IMS system.

## Output fields

In addition to the message, command response output is also returned in XML tag structure. The following table shows the IMPORT response output fields. The columns in the table are as follows:

### Short label

Contains the short label that is generated in the XML output.

### Long label

Contains the long label that is generated in the XML output.

### Keyword

Identifies keyword on the command that caused the field to be generated. N/A (not applicable) appears for output fields that are always returned. The value *error* appears for output fields that are returned only when an error occurs.

### Meaning

Provides a brief description of the output field.

Table 335. Output fields for the IMPORT command

Short label	Long label	Keyword	Meaning
ADD	ADD		The listed DBD or PSB was added as new to the IMS directory data set.  This output field is displayed only if it applies to one or more DBDs or PSBs.
CATS	CATSHR		<b>Y</b> The directory data sets are shared by IMS in the plex with ACBSHR=Y specified and with the same catalog alias specified.  <b>N</b> The directory data sets is accessed by this IMS only.
CC	CC	N/A	Completion code.
CCTXT	CCText	<i>error</i>	Completion code text that briefly explains the meaning of the nonzero completion code.
CHG	CHNG		The listed DBD or PSB was successfully modified in the IMS directory data sets.  This output field is displayed only if it applies to one or more DBDs or PSBs.
CPY	COPY		The listed DBD or PSB was added to the IMS directory data set, but cannot be used until its runtime control block is created in the online IMS system by either a CREATE DB or a CREATE PGM command.  This output field is displayed only if it applies to one or more DBDs or PSBs.
DBD	DBDName		Name of the DBD acted upon during IMPORT command processing.
DLT	DLET		The listed DBD or PSB was removed from the IMS directory data set because updates to another DBD removed the last or only logical relationship that required it.  This output field is displayed only if it applies to one or more DBDs or PSBs.
ERRT	ErrorText		Additional information about errors.

Table 335. Output fields for the IMPORT command (continued)

Short label	Long label	Keyword	Meaning
PHST	ErrPhase		A phase where an error occurred:  <b>OLCPREPF</b> The process failed at prepare phase. The process is aborted.  <b>OLCCMT1F</b> The process failed at commit phase 1. The process is aborted.  <b>OLCCMT2F</b> The process failed at commit phase 2. The process is aborted.  <b>OLCCMT3F</b> The process failed at commit phase 3. The process is aborted.  <b>OLCCMT4F</b> The process failed at commit phase 4. The process reached a commit point.
MBR	MbrName	N/A	IMSplex member that built the output line.
PSB	PSBName		Name of the imported PSB.
RFP	REFP		This IMS system is not yet using the most current version of the PSB that is in the IMS directory data set. To activate the most current version of the PSB in this IMS system, issue IMPORT DEFN SOURCE(REFRESHPSB) FOR( <i>this_ims</i> ).  This output field is displayed only if it applies to one or more DBDs or PSBs.
RFS	RFSH		The listed DBD or PSB was successfully imported into the IMS directory data set. The only difference between the imported version and the previous version is the timestamp.  This output field is displayed only if it applies to one or more DBDs or PSBs.

## Return and reason codes

The following table includes the return and reason codes and a brief explanation of the codes. The return or reason code returned for the command might also indicate an error from a CSL request.

Table 336. Return and reason codes for the IMPORT command

Return code	Reason code	Meaning
X'00000008'	X'00002004'	Invalid command keyword or invalid command keyword combination.
X'00000008'	X'00002008'	Insufficient number of keywords.
X'00000008'	X'00002009'	Multiple values for same attribute.
X'00000008'	X'00002028'	The command contains an invalid keyword.

Table 336. Return and reason codes for the IMPORT command (continued)

Return code	Reason code	Meaning
X'00000008'	X'00002176'	The Command (CMD) master was not provided.  If OPTION(UPDATEPSB) is specified, the IMS command master must be included in the FOR(IMSID(imsid)) list.
X'00000008'	X'00002177'	The Command (CMD) master was provided.  If OPTION(REFRESHPSB) is specified, the IMS command master must not be included in the FOR(IMSID(imsid)) list.
X'0000000C'	X'00003000'	Command was successful for some resource definitions but failed for others.
X'0000000C'	X'00003008'	Command was not successful for any of the resource definitions. The command output contains a line for each resource, accompanied by its completion code.
X'00000010'	X'00004100'	Resource structure is full.
X'00000010'	X'00004104'	No RM address space.
X'00000010'	X'00004108'	No SCI address space.
X'00000010'	X'0000400C'	Command is not valid on the XRF alternate.
X'00000010'	X'00004014'	Command is not valid on the RSR tracker.
X'00000010'	X'00004018'	No resource structure exists, or resource structure is not available.
X'00000010'	X'0000401C'	Command is not valid on the FDBR region.
X'00000010'	X'00004120'	Online change phase is in progress.
X'00000010'	X'00004124'	The IMPORT command is rejected, because a process of a similar type is in progress. Possible commands in progress are /DISPLAY MODIFY, CREATE DB, or DELETE DB commands. Wait until the other process completes and issue the IMPORT command again.
X'00000010'	X'00004509'	RM data version is lower than the IMS data version.
X'00000014'	X'00005000'	The IMPORT command was terminated, because IMODULE GETSTOR storage could not be obtained.
X'00000014'	X'00005004'	DFSOCMD response buffer could not be obtained.
X'00000014'	X'00005008'	DFSPOOL storage could not be obtained.
X'00000014'	X'0000500C'	AWE could not be obtained.
X'00000014'	X'00005010'	Unable to obtain latch.
X'00000014'	X'00005018'	Unable to obtain a 64-bit buffer.
X'00000014'	X'00005100'	RM request error.
X'00000014'	X'00005108'	SCI request error.
X'00000014'	X'00005FFF'	The IMPORT command could not be processed because of an internal error.
X'02000008'	X'0000203C'	The command contains an invalid keyword parameter value.
X'02000008'	X'00003004'	No requests were successful.

Errors unique to the processing of this command are returned as completion codes. The following table includes an explanation of the completion codes.

## Completion codes

Table 337. Completion codes for the *IMPORT* command

Completion code	Completion code text	Meaning
0		Command completed successfully for resource or descriptor definition.
1	COMMAND NOT APPLICABLE TO THIS IMS	The <i>IMPORT DEFN SOURCE(CATALOG)</i> command does not apply to this IMS.
2	COMMAND NOT ATTEMPTED BY THIS IMS	The <i>IMPORT DEFN SOURCE(CATALOG)</i> command was not attempted by this IMS for one of the following reasons: <ul style="list-style-type: none"> <li>• The <i>IMPORT DEFN SOURCE(CATALOG)</i> command master encountered an error and did not direct this IMS to process the <i>IMPORT DEFN SOURCE(CATALOG)</i> command.</li> <li>• The <i>IMPORT DEFN SOURCE(CATALOG)</i> command master rejected the command because one or more IMS systems are down. The <i>IMPORT DEFN SOURCE(CATALOG)</i> command master did not direct this IMS to process the command.</li> </ul>
3	IMS ALREADY IN REQUESTED OLC STATE	This IMS already processed the <i>IMPORT DEFN SOURCE(CATALOG)</i> command.
4	OLC PHASE INCOMPLETE ON THIS IMS	The <i>IMPORT DEFN SOURCE(CATALOG)</i> command is incomplete on this IMS. Look for other nonzero completion codes in the command output to determine why command processing is incomplete.
5	OLC COMPLETE, SOME MBRS UNUSABLE	The <i>IMPORT DEFN SOURCE(CATALOG)</i> command is complete; however, during internal commit processing after the IMS list was updated in the IMS directory, an error was encountered and one or more ACBs that were added or changed are unusable.
1D	XRF TAKEOVER IS IN PROGRESS	The <i>IMPORT DEFN SOURCE(CATALOG)</i> command cannot be processed because a takeover is in progress.
32	NOT ALLOWED FOR AN MSDB	The resident attribute in the imported database definition differs from the resident attribute in the runtime resource definition. The resident option of an MSDB cannot be changed.
51	NO RESOURCE STRUCTURE	The resource structure is not available. The resource could not be registered with RM. The resource definition was not imported.
52	RESOURCE STRUCTURE FULL	The resource structure is full. The resource could not be registered with RM. The resource definition was not imported.

Table 337. Completion codes for the IMPORT command (continued)

Completion code	Completion code text	Meaning
58	IMS NOT REGISTERED TO RM	IMS is not registered to RM.
60	IMODULE GETMAIN storage error.	
61	DFSBCB STORAGE ERROR	DFSBCB storage error. Could not get storage for DDIR, PDIR, RCTE, or SMB control block. The resource or descriptor definition was not imported.
62	HIOP storage error.	
63	WKAP storage error.	
64	GETSTOR STORAGE ERROR	DFSPOOL storage error. Could not get storage from the CMDP pool. The resource or descriptor was not imported.
73	PROGRAM SCHEDULED	Program is scheduled.
77	DEPENDENT REGIONS ACTIVE	One or more dependent regions are scheduled against the database, or one or more areas are open. The DEDB cannot be updated.
80	DATASET ERROR	Function (8 characters), ddname (8 characters), return code (8 bytes), and error detail (8 characters).  Function can be one of the following:  <b>BLDL</b> Data set BLDL error.  <b>BUSY</b> The data set is busy. The detail contains the job name that has the data set.  <b>DIR</b> Data set directory error.  <b>EMPTY</b> Library is empty.  <b>ENQUEUE</b> Data set enqueue error.  <b>EOF</b> Data set end-of-file (EOF) error.  <b>OPEN</b> Data set open error. Error detail can be 'DLS REG'.  <b>READ</b> Data set read error.  <b>WRITE</b> Data set write error.  Ddname can be the DFSDIRC, the ddname of the IMS directory data set or DFSSTDI, the ddname of the staging data set of the IMS directory.  Return code is the data set service return code.  Reason code is the data set service reason code.

Table 337. Completion codes for the IMPORT command (continued)

Completion code	Completion code text	Meaning
90	INTERNAL ERROR	An IMS internal error occurred. The definition was not imported.
91	TIMEOUT ERROR	An internal prepare phase, commit phase, or abort phase of the IMPORT DEFN SOURCE(CATALOG) command timed out before this IMS responded to the internal prepare or commit phase. The prepare phase or commit phase might have succeeded on this IMS.
94	RM REQUEST ERROR	An RM request error occurred. The resource could not be registered with RM. The resource definition was not imported.
B0	RESOURCE DEFINITION ERROR	The resource exists as another resource type in the RM structure. The resource definition was not imported.
B1	RESOURCE STATE ERROR	Resource type (8 characters), resource name (8 characters), error detail (16 characters).  The resource name is the name of the resource as it is defined to IMS, such as the database name, the program name, or the transaction name.  The error detail can be one of the following:  <b>AREA OPEN</b> An area associated with the FP DEDB is open.  <b>AREA STARTED</b> An area associated with the FP DEDB is started.  <b>DATABASE <i>dbname</i> ALTER INCOMPLETE</b> Alter processing is not complete for a HALDB database. After alter processing completes, retry the INITIATE OLC command.  <b>DATABASE <i>dbname</i> STG TOO SMALL</b> The IMPORT DEFN SOURCE(CATALOG) command cannot process the altered DDIR because there is not enough internal block storage.  The lack of storage could be caused by the start of another HALDB alter operation after the IMPORT command reaches the commit phase.  Retry the IMPORT DEFN SOURCE(CATALOG) command.

Error details continued in next row.

Table 337. Completion codes for the IMPORT command (continued)

Completion code	Completion code text	Meaning
B1	RESOURCE STATE ERROR (continued)	<p><b>DBD ACTIVE</b> A /DBDUMP command is active for the database.</p> <p><b>DBR ACTIVE</b> A /DBRECOVERY command is active for the database.</p> <p><b>DBR NEEDED</b> A /DBRECOVERY command is needed for the database resource. The IMPORT DEFN SOURCE(CATALOG) command was initiated to either change or delete one or more HALDB databases which have not been taken offline with a /DBR command.</p> <p>If the IMS catalog is enabled with an alias other than DFSC (the standard alias prefix), and the catalog databases have been modified with the IMPORT DEFN SOURCE(CATALOG) command, issue the /DBR DB command for the databases IMS1D000 and IMS1X000 instead of DFSCD000 and DFSCX000.</p> <p><b>DEP REGS ACTIVE</b> Dependent regions using the FP DEDB are active.</p> <p>Error details continued in next row.</p>

Table 337. Completion codes for the IMPORT command (continued)

Completion code	Completion code text	Meaning
B1	RESOURCE STATE ERROR (continued)	<p>Error detail (continued):</p> <p><b>INTERNAL ERROR</b> The resource control block is not found in the active IMS, or the resource control block exists. These are internal errors that should not occur.</p> <p><b>IN USE</b> The resource is in use.</p> <p>A transaction has queuing in progress (either terminal input program-to-program switch).</p> <p>A database is in use.</p> <p><b>PSB SCHEDULED</b> A program referencing a database to be changed or deleted is scheduled.</p> <p><b>QUEUEING</b> Messages are queued to the transaction or program.</p> <p><b>SCHEDULED</b> The named resource (a transaction or program to be changed or deleted, or a program referencing a database to be changed or deleted) is scheduled.</p> <p><b>SUSPENDED</b> The transaction to be changed or deleted is on the suspend queue.</p>
B2	IMS STATE ERROR	<p>IMS state error (32 characters). The IMS state can be one of the following:</p> <p><b>CHECKPOINT IN PROGRESS</b> This IMS has checkpoint in progress.</p> <p><b>NOT-REACHABLE</b> The online change phase is rejected because this IMS is NOT-REACHABLE. The SCI on the OS image where this IMS is active is down. Restart the SCI and reissue the IMPORT DEFN SOURCE(CATALOG) command.</p> <p><b>OLC ALREADY COMMITTED</b> The internal online change phase of IMPORT command processing is rejected because online change is already committed.</p>

Table 337. Completion codes for the *IMPORT* command (continued)

Completion code	Completion code text	Meaning
B2	IMS STATE ERROR (continued)	<p>IMS state error (32 characters). The IMS state can be one of the following:</p> <p><b>OLC PHASE IN PROGRESS</b> The IMPORT DEFN SOURCE(CATALOG) command is rejected because this IMS is processing either another IMPORT DEFN SOURCE(CATALOG) command or an online change request.</p> <p><b>RECOVERY IN PROGRESS</b> The IMPORT DEFN SOURCE(CATALOG) command is rejected because the online recovery service is already in progress.</p> <p><b>RESOLVE DB INDOUBTS</b> This IMS has DB indoubts. You must resolve the DB indoubts either by reconnecting the coordinator controller and IMS or by using an operator command.</p> <p><b>RESTART IN PROGRESS</b> This IMS has restart in progress.</p> <p><b>RESTART NOT COMPLETE</b> This IMS initialized before the IMPORT DEFN SOURCE(CATALOG) command was issued, but has not completed restart. The command is rejected as long as this IMS is in this state.</p>
D6	DATABASE IN USE	Database in use. The DEDB cannot be changed because the DEDB is in use in a region.
E1	DB MUST BE STOPPED AND OFFLINE	The database must be stopped and taken offline in order for the attribute to be changed. You might need to issue a /DBR DB command or an UPDATE DB STOP(ACCESS) command to stop the database and take it offline before issuing the IMPORT DEFN command.
120	STAGING ACBLIB IS EMPTY	The IMPORT DEFN SOURCE(CATALOG) command failed.
121	RESOURCE IN CMD HAS NO CHANGE	The IMPORT DEFN SOURCE(CATALOG) command failed.
122	NOT ALL PSBS REBUILT FOR THIS DMB	The IMPORT DEFN SOURCE(CATALOG) command failed.
123	STAGING LIBRARY LEVEL NOT COMPATIBLE WITH CURRENT IMS	The IMPORT DEFN SOURCE(CATALOG) command failed.

Table 337. Completion codes for the IMPORT command (continued)

Completion code	Completion code text	Meaning
125	NO PSB REBUILT	During processing of the IMPORT DEFN SOURCE(CATALOG) command, IMS found existing PSB resources in the IMS directory that must be rebuilt before a modified DBD resource can be activated.
1D0	NOT DONE DUE TO ERROR	The identified resource was not imported because an error occurred. Check for other completion codes that are greater than 4 to determine the cause of the error.
1F2	DIRECTORY BLDL FAILED	During processing of the IMPORT DEFN SOURCE(CATALOG) command, IMS encountered a BLDL error while reading the directory data sets.
1F3	BAD ACB FORMAT IN DIRECTORY	During processing of the IMPORT DEFN SOURCE(CATALOG) command, IMS encountered an ACB that contains a block with an unrecognized DBD or PSB format.
1F5	OPEN DIRECTORY FAILED	During processing of the IMPORT DEFN SOURCE(CATALOG) command, IMS was unable to open the IMS directory data set for update.  This is probably an internal error. Contact IBM Software Support.
1F6	CANNOT IMPORT IMS SYSTEM RSC	During processing of the IMPORT DEFN SOURCE(CATALOG) command, IMS encountered an ACB for the IMS catalog or another IMS system resource. The IMS catalog cannot be modified by using the IMPORT DEFN SOURCE(CATALOG) command.
1FB	PSB NOT REBUILT	During processing of the IMPORT DEFN SOURCE(CATALOG) command, IMS found existing PSB resources in the IMS directory that must be rebuilt before a modified DBD resource can be activated.
1FC	I/O ERROR READING DIRECTORY	During processing of the IMPORT DEFN SOURCE(CATALOG) command, IMS was unable to read from the IMS directory.  Determine the cause of the error and restore or rebuild the IMS directory. After the IMS directory is recovered, reissue the IMPORT command.
1FD	I/O ERROR READING STAGING	During processing of the IMPORT DEFN SOURCE(CATALOG) command, IMS was unable to read from the staging data set of the IMS catalog.  Resubmit the resource changes to IMS by using either DDL or one of the IMS catalog populate utilities. After the changes are resubmitted to IMS, reissue the IMPORT command.

Table 337. Completion codes for the IMPORT command (continued)

Completion code	Completion code text	Meaning
1FE	ADDED MBR NOT IN STAGING	This is probably an internal error. Contact IBM Software Support.
1FF	REPLACED MEMBER NOT IN STAGING	This is probably an internal error. Contact IBM Software Support.
227	ERROR CLEANUP DIRECTORY	IMPORT DEFN SOURCE(CATALOG) command processing attempted to clean up the directory data sets before initiating a new import activation process, but the cleanup process failed. Check the console for error messages and correct any errors before resubmitting the IMPORT command.
231	DLET MBR NOT IN DIRECTORY	During processing of the IMPORT DEFN SOURCE(CATALOG) command, IMS attempted to delete a resource from the IMS directory, but the resource was not found in the IMS directory.
235	EMPTY STAGING DIRECTORY	During processing of the IMPORT DEFN SOURCE(CATALOG) command, no resources were found to import in the staging data set.
236	I/O ERROR UPDATING DIRECTORY	During processing of the IMPORT DEFN SOURCE(CATALOG) command, IMS was unable to update the IMS directory.  This is probably an internal error. Contact IBM Software Support.
237	I/O ERROR READING DFSIMSL	During processing of the IMPORT DEFN SOURCE(CATALOG) command, IMS could not read the IMS list in the IMS directory.
238	I/O ERROR WRITING DFSIMSR	During processing of the IMPORT DEFN SOURCE(CATALOG) command, IMS could not update the resource list in the IMS directory.
23A	I/O ERROR WRITING DFSIMSL	During processing of the IMPORT DEFN SOURCE(CATALOG) command, IMS could not write to the list of IMS systems that share the IMS catalog in the IMS directory.
23C	THIS IMS CATALOG IS NO LONGER SHARED	The ACBSHR specification for this IMS system changed from ACBSHR=Y to ACBSHR=N. When processing the IMPORT DEFN SOURCE(CATALOG) command, IMS no longer includes this IMS system in the global coordination of changes to a shared IMS catalog in the IMSplex.
23D	DIRECTORY DATA SET FULL	The IMPORT DEFN SOURCE(CATALOG) command failed because the directory data set was full. Run the catalog recovery utility to get more spaces for the data sets.
23E	STAGING BLDL FAILED	During processing of the IMPORT DEFN SOURCE(CATALOG) command, IMS encountered a BLDL error while reading the staging data sets.

Table 337. Completion codes for the IMPORT command (continued)

Completion code	Completion code text	Meaning
240	IMSID NOT FOUND IN DFSIMSL	During processing of the IMPORT DEFN SOURCE(CATALOG) command, an imsid provided in the FOR(IMSID(imsid)) list does not exist in the DFSIMSL member of the IMS directory data set.
241	DMB FOUND IN STAGING DATA SET	During processing of the IMPORT DEFN SOURCE(CATALOG) command, a DMB was found in the staging data set.
242	NO RESOURCES IN REFRESH PENDING	During processing the IMPORT DEFN SOURCE(CATALOG) OPTION(REFRESHPSB) no PSBs were found to refresh.

## Examples

### Example 1 for IMPORT DEFN SOURCE(CATALOG)

The following example shows the output after issuing the IMPORT DEFN SOURCE(CATALOG) command in a single IMS system.

TSO SPOC input:

```
IMPORT DEFN SOURCE(CATALOG)
```

TSO SPOC output:

MbrName	Member	CC	CATSHR	DBDName	PSBName	CHNG	DLET
IMS1	IMS1	0	N				
IMS1	IMS1	0	N	DEDBGS1A		Y	
IMS1	IMS1	0	N	FPSI1ASA			Y
IMS1	IMS1	0	N		BMP255	Y	

OM API input:

```
CMD(IMPORT DEFN SOURCE(CATALOG))
```

OM API output:

```
<?xml version="1.0"?>
<!DOCTYPE imsout SYSTEM "imsout.dtd">
<imsout>
<ctl>
<omname>OM10M </omname>
<omvsn>1.7.0</omvsn>
<xmlvsn>20 </xmlvsn>
<statime>2015.276 06:24:56.083208</statime>
<stotime>2015.276 06:24:56.836962</stotime>
<staseq>CFA3D9F7A8308DE5</staseq>
<stoseq>CFA3D9F860362863</stoseq>
<rqsttkn1>USRT011 10232456</rqsttkn1>
<rc>00000000</rc>
<rsn>00000000</rsn>
</ctl>
<cmd>
<master>IMS1 </master>
<userid>USRT011 </userid>
<verb>IMP </verb>
<kwd>DEFN </kwd>
<input>IMPORT DEFN SOURCE(CATALOG) </input>
</cmd>
```

```

| <cmdrsphdr>
| <hdr s1bl="MBR" l1bl="MbrName" scope="LCL" sort="n" key="0"
|   scroll="yes" len="8" dtype="CHAR" align="left" />
| <hdr s1bl="MSMBR" l1bl="Member" scope="LCL" sort="a" key="4"
|   scroll="no" len="8" dtype="CHAR" align="left" />
| <hdr s1bl="CC" l1bl="CC" scope="LCL" sort="n" key="0" scroll="yes"
|   len="4" dtype="INT" align="right" />
| <hdr s1bl="CCTXT" l1bl="CCText" scope="LCL" sort="n" key="0"
|   scroll="yes" len="*" dtype="CHAR" align="left" skipb="yes" />
| <hdr s1bl="CATS" l1bl="CATSHR" scope="LCL" sort="n" key="0"
|   scroll="yes" len="1" dtype="CHAR" align="left" skipb="yes" />
| <hdr s1bl="RSC" l1bl="RSCName" scope="LCL" sort="a" key="3" scroll="no"
|   len="8" dtype="CHAR" align="left" skipb="yes" />
| <hdr s1bl="DBD" l1bl="DBDName" scope="LCL" sort="a" key="2" scroll="no"
|   len="8" dtype="CHAR" align="left" skipb="yes" />
| <hdr s1bl="PSB" l1bl="PSBName" scope="LCL" sort="a" key="1" scroll="no"
|   len="8" dtype="CHAR" align="left" skipb="yes" />
| <hdr s1bl="ADD" l1bl="ADD" scope="LCL" sort="n" key="0" scroll="yes"
|   len="1" dtype="CHAR" align="left" skipb="yes" />
| <hdr s1bl="CHG" l1bl="CHNG" scope="LCL" sort="n" key="0" scroll="yes"
|   len="1" dtype="CHAR" align="left" skipb="yes" />
| <hdr s1bl="CPY" l1bl="COPY" scope="LCL" sort="n" key="0" scroll="yes"
|   len="1" dtype="CHAR" align="left" skipb="yes" />
| <hdr s1bl="RFS" l1bl="RFSH" scope="LCL" sort="n" key="0" scroll="yes"
|   len="1" dtype="CHAR" align="left" skipb="yes" />
| <hdr s1bl="DLT" l1bl="DLET" scope="LCL" sort="n" key="0" scroll="yes"
|   len="1" dtype="CHAR" align="left" skipb="yes" />
| </cmdrsphdr>
| <cmdrspdata>
| <rsp>MBR(IMS1 ) IMSMBR(IMS1 ) CC( 0) CATS(N) </rsp>
| <rsp>MBR(IMS1 ) IMSMBR(IMS1 ) CC( 0) CCTXT() CATS(N)
|   DBD(DEDDBGS1A) CHG(Y) </rsp>
| <rsp>MBR(IMS1 ) IMSMBR(IMS1 ) CC( 0) CCTXT() CATS(N)
|   DBD(FPSI1ASA) DLT(Y) </rsp>
| <rsp>MBR(IMS1 ) IMSMBR(IMS1 ) CC( 0) CCTXT() CATS(N) PSB(BMP255
|   ) CHG(Y) </rsp>
| </cmdrspdata>
| </imsout>

```

**Example 2 for IMPORT DEFN SOURCE(CATALOG) in a shared ACB environment (ACBSHR=Y)**

The following example shows a scenario in which the processing of the IMPORT DEFN SOURCE(CATALOG) command fails in a multisystem environment where three IMS systems, IMS1, IMS2, and IMS3, share the ACBs in the IMS directory data set of the IMS catalog. In this scenario, IMS1, the command master, abends during phase 1 of commit processing. Later, IMS2 also abends.

The following example is of the QUERY MEMBER command that is routed to all IMS systems before IMS1, the command master, abends. The QUERY MEMBER command output shows that the processing of an IMPORT DEFN SOURCE(CATALOG) is in progress at phase 1 of commit processing in IMS1 and at phase 0 in both IMS2 and IMS3.

TSO SPOC input:

```
QUERY MEMBER TYPE(IMS) SHOW(ALL)
```

TSO SPOC output:

```

|   Response for: QUERY MEMBER TYPE(IMS) SHOW(ALL)
| MbrName   CC Type  Status LclAttr                LclStat
| IMS1      0 IMS    IMPCMD
| IMS1      0 IMS                GBLLOLC,NO-STM,DIRECTORY IMPPHC1I,SECCMD,SECMSG
| IMS2      0 IMS    IMPCMD

```

```

|           IMS2      0 IMS      GBLOLC,NO-STM,DIRECTORY IMPPHC01,SECCMD,SECMSG
|           IMS3      0 IMS      IMPCMD
|           IMS3      0 IMS      GBLOLC,NO-STM,DIRECTORY IMPPHC01,SECCMD,SECMSG

```

The following QUERY command is routed to all IMS systems and shows that IMS1, the command master for the IMPORT DEFN SOURCE(CATALOG) command, abended during phase 1 of commit processing. Because the commit point was not reached, the import is aborted.

Both IMS2 and IMS3 attempt to take over import processing to abort the updates that were made to the directory data sets and to cleanup RM process information.

TSO SPOC output:

```

|           Response for: QUERY MEMBER TYPE(IMS) SHOW(ALL)                               More:  >
| MbrName   CC Type  Status                               LclAttr                               LclStat>
| IMS2      0 IMS    IMPCMD,IMPPHC1T,IMPABORT
| IMS2      0 IMS    GBLOLC,NO-STM,DIRECTORY SECCMD,SECM
| IMS3      0 IMS    IMPCMD,IMPPHC1T,IMPABORT
| IMS3      0 IMS    GBLOLC,NO-STM,DIRECTORY SECCMD,SECM

```

In the following example, the QUERY MEMBER command was routed to all systems, but now IMS2, which became the new command master after the abend of IMS 1, also abended. IMS2 abended before it finished cleaning up the import process. IMS 3 cleans up its own local system, but not the directory or RM blocks, because IMS3 did not succeed in taking over processing after the first abend of IMS1.

```

|           Response for: QUERY MEMBER TYPE(IMS) SHOW(ALL)
| MbrName   CC Type  Status                               LclAttr                               LclStat
| IMS3      0 IMS    IMPCMD,IMPPHC1F,IMPABORT
| IMS3      0 IMS    GBLOLC,NO-STM,DIRECTORY SECCMD,SECM

```

To resolve import processing after the failures, the IMPORT DEFN SOURCE(CATALOG) command is issued again on IMS3. The new import command cleans up the prior import process before it attempts to import the resources again. After IMS1 and IMS2 are restarted, they will pick up the changed resources from the directory and will not process any IMPORT log records.

```

| Response for: IMPORT DEFN SOURCE(CATALOG)
| MbrName  Member   CC CText          ErrorText                               ErrPhase CATSHR DBDName  PSBName  CHNG
| -----
| IMS3     IMS1      4 IMS STATE ERROR ABENDED
| IMS3     IMS2      4 IMS STATE ERROR ABENDED
| IMS3     IMS3        0
| IMS3     IMS3        0                               Y                               DBOHIDK5          Y
| IMS3     IMS3        0                               Y                               DBVHDJ05          Y
| IMS3     IMS3        0                               Y                               D2XHDJ05          Y
| IMS3     IMS3        0                               Y                               D2XHIDK5          Y
| IMS3     IMS3        0                               Y                               PSBEJK05          Y

```

**Example 3 for the OPTION() keyword of the IMPORT DEFN SOURCE(CATALOG) command**

The following example shows a scenario in which a PSB is activated in only a subset of IMS systems in an IMSplex by issuing a sequence of IMPORT DEFN SOURCE(CATALOG) commands with OPTION(UPDATEPSB) or OPTION(REFRESHPSB) specified.

There are three IMS systems, IMS1, IMS2, and IMS3, in the IMSplex and they share the ACBs in the IMS directory data set of the IMS catalog.

The following command with OPTION(UPDATEPSB) activates PSB DDLTFPE1 on IMS1 only, which is also the command master. On IMS2 and IMS3, the PSB is placed in a refresh-pending state.

TSO SPOC input:

```
IMPORT DEFN SOURCE(CATALOG) OPTION(UPDATEPSB) FOR(IMSID(IMS1))
```

The log output for the preceding command shows that IMS1 is the command master.

```
Log for:  IMPORT DEFN SOURCE(CATALOG) OPTION(UPDATEPSB) FOR(IMSID(IMS1))
```

```

IMSpIex . . . . . : PLEX1
Routing . . . . . : IMS1
Start Time. . . . : 2016.264 15:18:59.26
Stop Time . . . . : 2016.264 15:18:59.86
Return code . . . : 00000000
Reason code . . . : 00000000
Reason text . . . :
Command master. . : IMS1

```

The TSO SPOC output shows that the active PSB on IMS1 changed:

```
Response for: IMPORT DEFN SOURCE(CATALOG) OPTION(UPDATEPSB) FOR(IMSID(IMS1))
```

Member	CC	CATSHR	PSBName	CHNG	REFP
IMS1	0	Y			
IMS1	0	Y	DDLTFPE1	Y	
IMS1	0	Y	DDLTFPE1		Y
IMS1	0	Y	DDLTFPE1		Y

Next, the IMPORT command is issued with OPTION(REFRESHPSB) and FOR(IMSID(IMS2)) specified to activate the PSB on IMS2. On IMS3, the PSB remains in a refresh-pending state. On IMS1, the refresh request is not applicable because the PSB is already active. The command response also shows that the refresh is not applicable for IMS1 (as it was the command master) and that the command was not attempted for IMS3 as it was not in the FOR() clause.

TSO SPOC input:

```
IMPORT DEFN SOURCE(CATALOG) OPTION(REFRESHPSB) FOR(IMSID(IMS2))
```

The log output for the preceding command shows that IMS1 is still the command master.

```
Log for:  IMPORT DEFN SOURCE(CATALOG) OPTION(REFRESHPSB) FOR(IMSID(IMS2))
```

```

IMSpIex . . . . . : PLEX1
Routing . . . . . : IMS1
Start Time. . . . : 2016.264 15:19:57.28
Stop Time . . . . : 2016.264 15:19:57.67
Return code . . . : 02000004
Reason code . . . : 00001014
Reason text . . . : At least one request completed with warning/s.
Command master. . : IMS1

```

The TSO SPOC output shows that the active PSB on IMS2 changed, the command was not applicable to IMS1, and not processed by IMS3:

```
Response for: IMPORT DEFN SOURCE(CATALOG) OPTION(REFRESHPSB) FOR(IMSID(IMS2))
```

Member	CC	CCText	CATSHR	PSBName	CHNG
IMS1	1	CMD NOT APPLICABLE TO THIS IMS			
IMS1	2	CMD NOT ATTEMPTED BY THIS IMS			
IMS1	0		Y	DDLTFPE1	Y

**Related concepts:**

 How to interpret CSL request return and reason codes (System Programming APIs)

**Related reference:**

Chapter 2, “Command keywords and their synonyms,” on page 63

---

## IMPORT DEFN SOURCE(REPO | RDDS) command

Use the IMPORT DEFN SOURCE(REPO | RDDS) command to create resource and descriptor definitions or replace existing resource and descriptor definitions in an online IMS system by using the definitions in a resource definition data set (RDDS) or the IMSRSC repository.

The IMPORT command can be issued through TSO SPOC, IBM Management Console for IMS and DB2 for z/OS, the Manage Resources options in the IMS Applications menu, or any user-written programs that issue commands through OM API. These commands can also be issued to an IMSplex by using the Batch SPOC utility.

**Subsections:**

- “Environment”
- “Syntax” on page 849
- “Keywords” on page 850
- “Usage notes” on page 855
- “Output fields” on page 857
- “Return and reason codes” on page 859
- “Completion codes” on page 861
- “Examples” on page 874

### Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

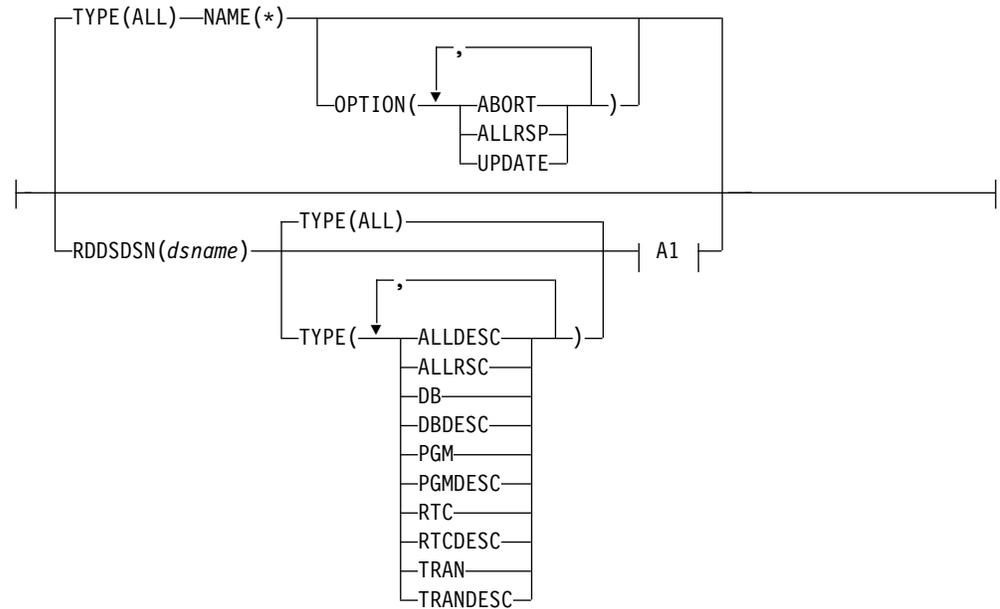
*Table 338. Valid environments for the IMPORT command and keywords*

Command / Keywords	DB/DC	DBCTL	DCCTL
IMPORT	X	X	X
DEFN	X	X	X
NAME	X	X	X
OPTION	X	X	X
RDDSDSN	X	X	X
SCOPE	X	X	X
SOURCE(RDDS), SOURCE(REPO)	X	X	X
TYPE	X	X	X

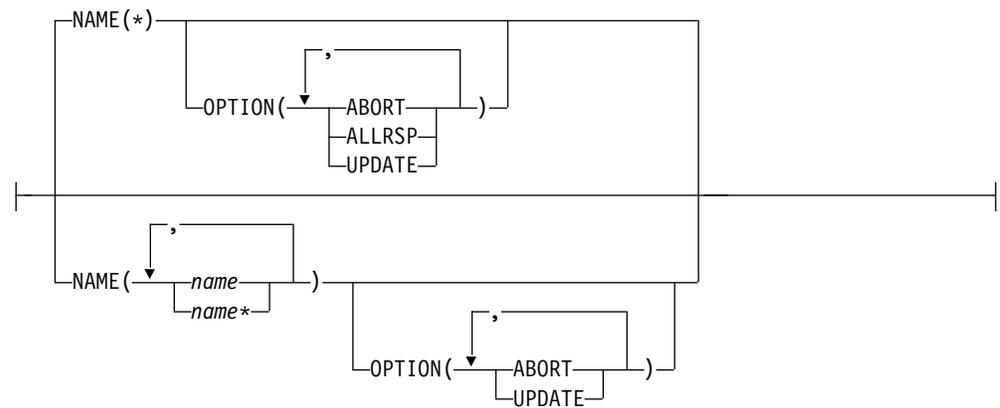
## Syntax



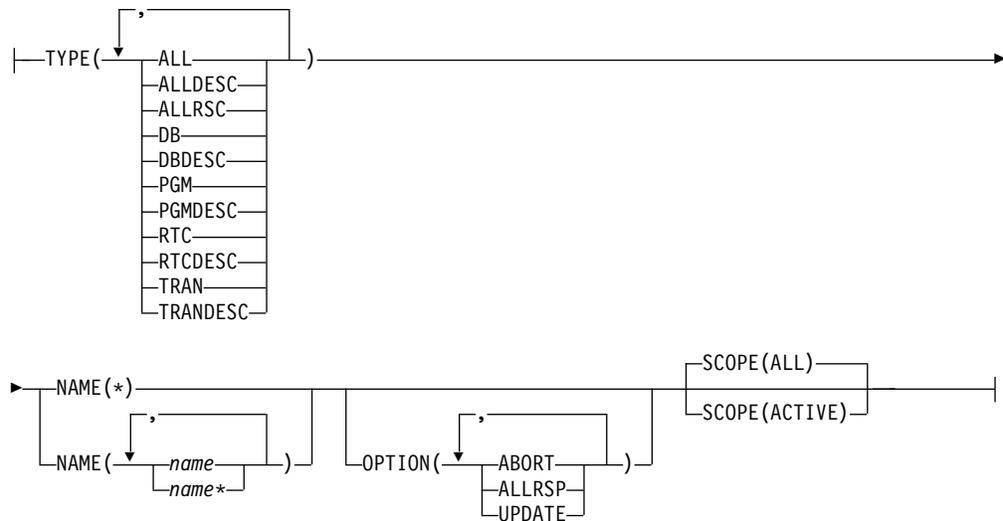
### A:



### A1:



### B:



## Keywords

The following keywords are valid for the IMPORT command:

### DEFN

A required keyword.

### NAME()

Specifies the 1-8 character name of a resource or descriptor for which you are importing a definition from either an RDDS or the repository. Wildcards can be specified in the name. The name is a repeatable parameter. If the NAME parameter specified is a specific name or a wildcard, command responses are returned for all the resource names that are processed. NAME is an optional parameter, and if not specified, it defaults to NAME(\*). For NAME(\*), command responses are returned only for the resource and descriptor names that resulted in an error. OPTION(ALLRSP) can be specified with NAME(\*) to obtain the command responses for all of the resource and descriptor names that are processed.

### OPTION()

Specifies additional functions to be performed along with the command.

You can specify the following options:

#### ABORT

Processes the definitions specified as a group. If an error occurs while processing any of the resource or descriptor definitions, the IMPORT command fails and none of the resource or descriptor definitions are imported.

If option ABORT is not specified and an error occurs while processing a particular resource or resource descriptor, that resource or descriptor is not imported. Import processing continues with the next resource or descriptor definition.

**Important:** Commands such as CREATE, DELETE, EXPORT, IMPORT, and UPDATE, which affect the definitional attributes of resources and descriptors, are not allowed while the IMPORT OPTION(ABORT) command is in progress.

## **ALLRSP**

Indicates that a response is to be returned for all resources and descriptors that are processed by this command. If NAME(\*) is specified, the default action is to return a response only for the resources and descriptors that result in an error.

OPTION(ALLRSP) can be used with NAME(\*) to request a response for all resources and descriptors processed by this command. If a name value other than NAME(\*) is specified, the default action is to return a response for all resources and descriptors processed by the command. If OPTION(ALLRSP) is specified with a name value other than NAME(\*) the ALLRSP parameter is ignored.

## **UPDATE**

Indicates that if the definition being imported is for a resource or descriptor that already exists in IMS, the imported definition should be used to replace the existing runtime resource or descriptor definition. If the definition being imported is for a resource or descriptor that does not exist, the imported definition should be used to create the runtime resource or descriptor definition. If the UPDATE option is not specified and a runtime definition already exists for the resource or descriptor, the import of the resource or descriptor definition fails.

In most cases, the affected resource must not be in use when the IMPORT OPTION(UPDATE) command is entered. If the resource is in use, the import of the stored definition fails. The following transaction attributes can be updated if the transaction is in use: CLASS, LCT, LPRI, NPRI, MAXRGN, PARLIM, PLCT, PLCTTIME, SEGNO, SEGSZ, and TRANSTAT. The following program attribute can be updated while the program is in use: TRANSTAT.

To minimize the likelihood that the import of a resource definition will fail, do the following steps before issuing the IMPORT command:

1. Stop the resource.
2. Query the resource to check for work in progress.
3. Complete the work, if any.

If the imported definition is for a resource or descriptor that already exists in IMS, the import time stamp in the existing runtime definition is replaced with the time the IMPORT command was received by OM. If one or more of the attributes in the existing runtime definition are different from the attributes in the imported definition, the update time stamp is also updated with the time the IMPORT command was received by OM. The access and create time stamps in the existing runtime definition are unchanged.

If the imported definition is for a resource or descriptor that does not exist in IMS, the import time stamp in the newly created runtime definition is set to the time the IMPORT command was received by OM. The create time stamp is obtained from the imported definition and stored in the new runtime definition.

If the imported definition is for a descriptor that has DEFAULT(N) defined and the runtime descriptor is the current default descriptor, the default value is not updated. The runtime descriptor remains the default descriptor. Other attributes are updated, but the default value remains unchanged. To change the default descriptor so that it is no longer the default descriptor, you must update another descriptor to be the default

descriptor. If the imported definition has DEFAULT(Y) defined, the updated runtime descriptor becomes the current default descriptor.

The DEFNTYPE of a newly created definition is set to IMPORT. The DEFNTYPE is set to IMPORT when an existing definition is replaced with a new definition.

When the ABORT option is also specified, the definitions are imported as a group. If an error occurs while processing any of the resource or descriptor definitions, the IMPORT command fails and none of the resource or descriptor definitions are imported.

If the ABORT option is not specified and an error occurs while processing a resource or descriptor definition, the resource or descriptor is not imported. Import processing continues with the next resource or descriptor definition. In an IMSplex environment, the IMPORT command might succeed on some IMS systems and fail on others.

#### **RDDSDSN(*dsname*)**

Specifies an optional RDDS from which resource and descriptor definitions are to be imported. The data set specified can be either a system RDDS or a non-system RDDS. If the RDDSDSN keyword is not specified, resource and descriptor definitions are imported from the most current system RDDS. If RDDS system data sets are not defined in the DFSDFxxx PROCLIB member, the RDDSDSN keyword is required on the IMPORT command.

#### **SCOPE**

Indicates the scope of the IMPORT DEFN SOURCE(REPO) command and whether any action is to be taken in Resource Manager (RM), along with the IMS systems that receive the command. The IMS systems that receive the command are identified by the ROUTE keyword on the OM API (such as TSO SPOC or Batch SPOC).

**Recommendation:** Specify ROUTE(ALL) so that all active IMS systems that have the resources defined will receive the command. If a ROUTE list is specified, the command is processed only by the IMS systems that are on the list. Other active IMS systems that have the resources defined but not on the ROUTE list will not be synchronized with the repository because they do not receive the command.

#### **ALL**

Indicates that the runtime resource definitions of the resource and descriptor names specified on the IMPORT command apply to all the IMS systems to which the resources and descriptors are defined in the repository.

When an IMPORT DEFN SOURCE(REPO) SCOPE(ALL) command is issued, resource definitions for the specified resource and descriptor names are read from the IMSRSC repository and imported at all active systems that receive the IMPORT command.

If SCOPE(ALL) is specified and an IMS system to which the resources and descriptors are defined is not active, the command master IMS calls RM to create or update an IMS change list in the repository with a list of the resource and descriptor names being imported. The resource definitions of all the resource names in the IMS change list are imported at warm or emergency restart so that the IMS is synchronized with all other IMS systems in the IMSplex. If a change list is created in the repository, a response line is returned on the IMPORT command for each resource name

in the change list with the IMSID of the IMS for which the change list is created, along with the import type (IMPTYPE) of the change list.

**Note:** A change list is created only if the command master IMS is IMS Version 13 or later and the RM that processes change list requests is at V13 (1.6) level or higher.

If the IMPORT DEFN SCOPE(ALL) command is issued without the OPTION(UPDATE) keyword and routed to an IMS system that has one or more resources or descriptors already defined, the IMPORT command results in a nonzero return code. The import might be successful at other active IMS systems, and the command might be successful in creating the change list in the IMSRSC repository for inactive IMS systems.

To avoid a nonzero return code and reason code from the IMPORT DEFN SCOPE(ALL) command, do one of the following:

- Specify the IMPORT DEFN SCOPE(ALL) command with the OPTION(UPDATE) keyword and route it to all IMS systems (ROUTE(\*))
- Route the IMPORT DEFN SCOPE(ALL) command to active IMS systems in which the resource or descriptors do not exist

#### **ACTIVE**

Indicates that the runtime resource definitions of the resource and descriptor names specified on the IMPORT command apply only to the active IMS systems to which the resources and descriptors are defined in the repository.

If an IMS system to which the resources and descriptors are defined is not active, that IMS does not obtain the resource definitions during warm or emergency restart because change lists are not created with the SCOPE(ACTIVE) keyword. To synchronize runtime resource definitions in all IMS systems in the IMSplex, you must issue the IMPORT command manually.

#### **SOURCE()**

Specifies the source of the resource and descriptor definitions to be imported.

#### **RDDS**

Specifies that resource and descriptor definitions are to be imported from an RDDS. If a data set name is not specified on the command with the RDDSDSN() keyword, the most current system RDDS is used.

Even if IMS is enabled with the repository, import from system RDDSs is allowed if the system RDDSs are defined. Import from a non-system RDDS is also valid when IMS is enabled to use the repository.

#### **REPO**

Indicates that the runtime resource definitions of the resource names specified on the NAME keyword are to be imported from the stored resource definitions in the repository. SOURCE(REPO) is valid only when IMS is enabled to use the repository.

The IMPORT command does not import resource and descriptor definitions from the repository if the resource or descriptor names do not exist in the IMS resource list for the IMS in the repository. You must first add the resource and descriptor names to the IMS resource list for the IMS before issuing the IMPORT command at the IMS. An EXPORT command with the SET(IMSID()) option can be used to add the resource definitions to an IMS resource list in the repository.

The resource definitions are imported from the repository and created in IMS if the resource definitions do not exist on the local IMS. If the resource definitions exist at the local IMS and OPTION(UPDATE) is specified, the resource definitions are replaced in the local IMS system.

#### **TYPE()**

Specifies the type of resource or descriptor definition to import.

TYPE() is not supported when SOURCE(CATALOG) is specified.

You can specify the following resource types:

#### **ALL**

ALL specifies that all types of resource and descriptor definitions (DB, DBDESC, PGM, PGMDESC, RTC, RTCDESC, TRAN, and TRANDESC) are to be imported. ALL is the only valid value when importing from a system RDDS.

#### **ALLDESC**

ALLDESC specifies that all definitions for all descriptor types (DBDESC, PGMDESC, RTCDESC, and TRANDESC) are to be imported. TYPE(ALLDESC) is not valid when importing from a system RDDS.

#### **ALLRSC**

ALLRSC specifies that all definitions for all resource types (DB, PGM, RTC, and TRAN) are to be imported. TYPE(ALLRSC) is not valid when importing from a system RDDS.

**DB** DB specifies that one or more database definitions are to be imported. TYPE(DB) is not valid when importing from a system RDDS.

#### **DBDESC**

DBDESC specifies that one or more database descriptor definitions are to be imported. TYPE(DBDESC) is not valid when importing from a system RDDS.

#### **PGM**

PGM specifies that one or more program definitions are to be imported. TYPE(PGM) is not valid when importing from a system RDDS.

#### **PGMDESC**

PGMDESC specifies that one or more program descriptor definitions are to be imported. TYPE(PGMDESC) is not valid when importing from a system RDDS.

#### **RTC**

RTC specifies that one or more route code definitions are to be imported. TYPE(RTC) is not valid when importing from a system RDDS.

#### **RTCDESC**

RTCDESC specifies that one or more route code descriptor definitions are to be imported. TYPE(RTCDESC) is not valid when importing from a system RDDS.

#### **TRAN**

TRAN specifies that one or more transaction definitions are to be imported. TYPE(TRAN) is not valid when importing from a system RDDS.

#### **TRANDESC**

TRANDESC specifies that one or transaction descriptor definitions are to be imported. TYPE(TRANDESC) is not valid when importing from a system RDDS.

## Usage notes

The IMPORT DEFN SOURCE(REPO | RDDS) command can be issued only through the Operations Manager (OM) API. This command applies to DB/DC, DBCTL and DCCTL systems.

The IMPORT command is not valid on the XRF alternate, RSR tracker, or FDBR region.

The IMPORT DEFN SOURCE(REPO | RDDS) command is valid for the following resource types:

- MODBLKS resources, such as the runtime resource attributes for databases, application programs, transactions, or Fast Path routing codes, if dynamic resource definition (DRD) is enabled by MODBLKS=DYN in either the COMMON\_SERVICE\_LAYER section of the DFSDFxxx or DFSCGxxx PROCLIB member)

The IMPORT DEFN SOURCE(REPO | RDDS) command is defined as ROUTE=ALL to OM. When the IMPORT command is issued, OM routes the IMPORT command to all active IMS systems (or to all IMS systems specified in the route list if the ROUTE parameter is specified on the CSLOMCMD or CSLOMI request) that have registered interest in the command. All IMS systems that receive the command attempt to process the IMPORT command and import the resource definitions from the specified source. If you want the IMPORT command to be processed by a specific IMS, the ROUTE parameter must be used to direct the IMPORT command to the specific IMS.

If the definition is for a resource or descriptor that is unknown to IMS, the IMS creates the internal control blocks that are needed to manage the resource. If the definition is for a resource or descriptor that exists in IMS and OPTION(UPDATE) is not specified, the definition is not imported. If the definition is for a resource or descriptor that exists in IMS and OPTION(UPDATE) is specified, the definition is imported and the existing runtime resource or descriptor definition is replaced with the attributes from the imported definition.

If the SOURCE(RDDS) keyword is specified for the IMPORT command, resource and descriptor definitions are created from or replaced by definitions from the system RDDS with the most current resource and descriptor definitions, or they are imported from an RDDS specified with the RDDS(SN) keyword on the IMPORT command. The RDDS specified with the RDDS(SN) keyword can be a system RDDS or a non-system RDDS. A system RDDS is one of the RDDSs defined with the RDDS(SN)= parameter in the DYNAMIC\_RESOURCES section of the DFSDFxxx PROCLIB member. A system RDDS contains all of the resource and descriptor definitions for a single IMS. If using automatic import or automatic export, each IMS must have its own set of system RDDSs.

If an RDDS contains multiple definition instances for the same resource or descriptor, the last definition instance imported is the definition used to create the resource or descriptor.

The definitions for IMS-defined resources, such as the Fast Path utility (DBF#FPU0) and the IMS-defined descriptors (DBFDSRT1, DFSDSDB1, DFSDSPG1 and DFSDSTR1), cannot be exported or imported. IMS-defined resources and descriptors are created at IMS restart. If the name of an IMS-defined resource or descriptor is specified in the NAME() parameter, the import of the resource or

descriptor definition fails with a completion code of 48 and a completion code text of "NOT ALLOWED FOR IMS RESOURCE".

**Rules and restrictions that apply to the IMPORT command:** The IMPORT DEFN SOURCE(REPO | RDDS) command imports resource and descriptor definitions from an RDDS or the repository and uses the definitions to create the runtime resource and descriptor definitions in the online IMS system. Therefore, the same rules and restrictions that apply to the CREATE commands for databases, programs, routing codes, and transactions also apply to the IMPORT command. For example, when importing a transaction definition, the definition for the program referenced by the transaction must either be imported at the same time, or the program must already be defined in the online IMS. The definitional attributes of the transaction must correspond to the definitional attributes of the program. The only exception is the CREATE DB command for a HALDB master database. The CREATE DB command for a HALDB master database creates the runtime resource definitions for both the HALDB master database and its associated HALDB partition databases. The IMPORT command for a HALDB master database creates the runtime resource definition for the HALDB master database only. The runtime resource definitions for the HALDB partition databases must be created with either a /START DB or UPDATE DB START(Access) command for the HALDB master database.

The runtime resource and descriptor definitions that are created by the IMPORT command exist in the online system until IMS terminates, unless they are deleted using a DELETE command. They are recoverable across an IMS warm start or emergency restart. To preserve the resource and descriptor definitions across a cold start, export the definitions to an RDDS or the repository before IMS terminates, and then import the stored definitions from the RDDS or the repository back into IMS either during cold start processing by using the automatic import function or, once IMS is up and running, by using the IMPORT command.

Runtime resource and descriptor definitions are lost if IMS is cold started (/NRE CHKPT 0 or /ERE COLDSYS), unless an automatic import from the current system RDDS or the repository is done during cold start.

When a transaction resource definition is imported from either an RDDS or the repository and a new runtime definition is created, the current priority (CPRI) is initialized to the normal priority (NPRI). When a transaction resource definition is imported from either an RDDS or the repository and an existing runtime definition is updated, the CPRI value is not updated.

When OPTION(ABORT) is not specified, IMPORT, EXPORT, and automatic export are processed serially. CREATE, DELETE, and UPDATE commands, which do not affect the imported resources, can run in parallel with the IMPORT command.

#### ***HALDB considerations***

The IMPORT command creates runtime resource and descriptor definitions in an online IMS system from definitions that have been exported to an RDDS or the repository. Whereas HALDB master database definitions can be exported to an RDDS or the repository, HALDB partition definitions cannot be exported to an RDDS or the repository. If the definition for a HALDB master database is imported, the IMPORT command creates only runtime resource definitions for the HALDB master database; it does not create runtime resource definitions for the

associated HALDB partitions. A completion code of “D” is returned for the HALDB master database and the completion code text of “RSC NEEDS STA DB ON HALDB MASTER” is returned.

After a HALDB master database is imported, you must issue a /START DB or UPDATE DB command on the HALDB master database to build its associated partition structures.

- /START DB *HALDB\_master*
- UPDATE DB NAME(*HALDB\_master*) START(ACCESS)

### ***MSDB considerations***

Like a CREATE DB command for an MSDB database, an IMPORT command for an MSDB database creates runtime resource definitions for the MSDB. If a CREATE DB or IMPORT command creates one or more MSDB databases, the MSDB resource control blocks are logged in the checkpoint log records. However, the MSDB databases cannot be used and they are not written to the MSDB checkpoint data sets (MSDBCPx) until they are loaded into online storage either at the next IMS cold start, or IMS warm start with MSDBLOAD (/NRESTART MSDBLOAD) from the MSDBINIT data set.

Like the UPDATE DB command, an IMPORT command that changes runtime resource definition attributes such as the RESIDENT attribute is rejected for MSDBs and DEDBs. An IMPORT command that changes the access type of an MSDB is also rejected.

If IMS is warm-started without MSDBLOAD (/NRESTART) or is emergency restarted (/ERESTART) without loading MSDB databases from the MSDBINIT data set, a new message, DFS2608W MSDBS NOT AVAILABLE - NO MSDBCPX DATA SET NAMES FOUND IN CHECKPOINT, is issued at IMS warm or emergency restart. IMS continues to run with no MSDB databases available until MSDB databases are loaded from the MSDBINIT data set during a subsequent cold start or warm restart with MSDBLOAD (/NRESTART MSDBLOAD).

## **Output fields**

When SOURCE(RDDS) is specified, message DFS3405I is returned by each IMS that processes the IMPORT command. It indicates whether all, some, or none of the specified resource or descriptor definitions were imported. It also contains the name of the data set from which the resource or descriptor definitions were imported. Each DFS3405I message is encapsulated in a single set of <msg> </msg> XML tags.

When SOURCE(REPO) is specified, the IMPORT command results in message DFS3406I, which indicates that the import from the repository was successful, or DFS3407E, which indicates that the import was not successful. These messages include the repository name and repository type. If the import from the repository is not successful because of a CSLRPQRY error, the DFS3407E message consists of the CSLRPQRY request return code and reason code. The CSLRPQRY return and reason codes are described in the CSLRRR macro.

In addition to the message, command response output is also returned in XML tag structure. The following table shows the IMPORT response output fields. The columns in the table are as follows:

**Short label**

Contains the short label generated in the XML output.

**Long label**

Contains the long label generated in the XML output.

**Keyword**

Identifies keyword on the command that caused the field to be generated. N/A (not applicable) appears for output fields that are always returned. The value *error* appears for output fields that are returned only in the case of an error.

**Meaning**

Provides a brief description of the output field.

Table 339. Output fields for the IMPORT command

Short label	Long label	Keyword	Meaning
CC	CC	N/A	Completion code.
CCTXT	CCText	<i>error</i>	Completion code text that briefly explains the meaning of the nonzero completion code.
IMPTYPE	ImpType	N/A	<p>Import type of the resource or descriptor. One of the following values might be set:</p> <p><b>CREATE</b> Indicates that a resource or descriptor definition is imported and a new resource or descriptor is created.</p> <p><b>UPDATE</b> Indicates that the attributes of an existing runtime resource or descriptor definition are updated by the import.</p> <p><b>CHGLIST</b> Indicates that the command master IMS is creating or updating the IMS change list for one or more IMS systems that are down.</p> <p>Indicates that the command master IMS created or updated the IMS change list for the IMS whose IMSID is in the IMSid column. The change list has been created or updated because the IMS is down and the IMPORT DEFN SCOPE(ALL) command is specified.</p> <p><b>(blank)</b> Indicates that a resource or descriptor definition is not imported due to an error, or that none of the attributes of an existing resource or descriptor definition are changed by the import.</p>

Table 339. Output fields for the IMPORT command (continued)

Short label	Long label	Keyword	Meaning
IMSID	IMSid	SCOPE(ALL)	IMSID of the IMS that is down and whose change list is updated with this resource. If the change list does not exist, it is created.
MBR	MbrName	N/A	IMSpIex member that built the output line.
NAME	Name	NAME	Descriptor or resource name.
OLDDEF	OldDefault	N/A	Old default descriptor name if this descriptor was made the default by the IMPORT command. The old default descriptor is no longer the default.
TYPE	Type	TYPE	Type of resource or descriptor, which can be DB, DBDESC, PGM, PGMDESC, RTC, RTCDESC, TRAN, or TRANDESC.

## Return and reason codes

The following table includes the return and reason codes and a brief explanation of the codes. The return or reason code returned for the command might also indicate an error from a CSL request.

Table 340. Return and reason codes for the IMPORT command

Return code	Reason code	Meaning
X'00000000'	X'00000000'	Command completed successfully. If NAME(*) is not specified (or defaulted to), the command output contains a line for each resource, accompanied by its completion code. If NAME(*) is specified and OPTION(ALLRSP) is not specified, no output lines are returned. If NAME(*) is specified and OPTION(ALLRSP) is specified, output lines are returned for each resource or descriptor definition imported.
X'00000008'	X'00002004'	Invalid command keyword or invalid command keyword combination.
X'00000008'	X'00002008'	Insufficient number of keywords.
X'00000008'	X'00002028'	The command contains an invalid keyword.
X'00000008'	X'0000211F'	Unable to determine the import data set name. The RDDSDSN() keyword was not specified on the command, and there are no system RDDDs defined.
X'00000008'	X'00002138'	Values other than NAME(*) and TYPE(ALL) were specified, and either the RDDSDSN() keyword was not specified or the data set specified with the RDDSDSN() keyword is a system RDDDs.
X'00000008'	X'00002139'	A value was specified for the RDDSDSN keyword, but SOURCE(RDDDs) was not specified.
X'00000008'	X'0000214E'	SOURCE(RDDDs) and the SCOPE() keyword conflict.
X'00000008'	X'00002176'	The Command (CMD) master was not provided.
X'00000008'	X'00002177'	The Command (CMD) master was provided.

Table 340. Return and reason codes for the IMPORT command (continued)

Return code	Reason code	Meaning
X'0000000C'	X'00003000'	Command was successful for some resource definitions but failed for others. If OPTION(ALLRSP) is specified, the command output contains a line for each resource, accompanied by its completion code. If OPTION(ALLRSP) is not specified, output lines are only returned for resources with nonzero completion codes.
X'0000000C'	X'00003008'	Command was not successful for any of the resource definitions. The command output contains a line for each resource, accompanied by its completion code.
X'00000010'	X'00004004'	No CQS address space.
X'00000010'	X'00004100'	Resource structure is full.
X'00000010'	X'00004104'	No RM address space.
X'00000010'	X'00004108'	No SCI address space.
X'00000010'	X'0000400C'	Command is not valid on the XRF alternate.
X'00000010'	X'00004014'	Command is not valid on the RSR tracker.
X'00000010'	X'00004018'	No resource structure exists, or resource structure is not available.
X'00000010'	X'0000401C'	Command is not valid on the FDBR region.
X'00000010'	X'00004038'	No requests were processed. OPTION(ABORT) was specified on the IMPORT command, and the number of database definitions to be imported exceeds 32K.
X'00000010'	X'00004114'	Unable to access the RDDDS. The RDDDS is either the system RDDDS with the most current data or the data set specified with the RDDSDSN() keyword.
X'00000010'	X'00004118'	Unable to allocate the RDDDS. The RDDDS is either the system RDDDS with the most current data or the data set specified with the RDDSDSN() keyword.
X'00000010'	X'0000411C'	The RDDDS from which the definitions are to be imported contains invalid data.
X'00000010'	X'00004120'	Online change phase is in progress.
X'00000010'	X'00004124'	The IMPORT command is rejected, because a process of a similar type is in progress. Possible commands in progress are /DISPLAY MODIFY, CREATE DB, or DELETE DB commands. Wait until the other process completes and issue the IMPORT command again.
X'00000010'	X'00004300'	Command is not allowed, because online change for MODBLKS is enabled (DFSDFxxx or DFSCGxxx defined with MODBLKS=OLC, or MODBLKS not defined).
X'00000010'	X'00004500'	IMS is not enabled with the repository.
X'00000010'	X'00004502'	Repository not available
X'00000010'	X'00004504'	Repository spare recovery is in progress.
X'00000010'	X'00004505'	No IMS resource list exists, or no resources for the resource type exist in the IMS resource list.
X'00000010'	X'00004507'	Repository access was denied.
X'00000010'	X'00004508'	Repository maximum put length exceeded.

Table 340. Return and reason codes for the IMPORT command (continued)

Return code	Reason code	Meaning
X'00000010'	X'00004509'	RM data version is lower than the IMS data version.
X'00000010'	X'0000450A'	Repository Server (RS) is being shut down.
X'00000010'	X'0000450B'	RS is not available.
X'00000010'	X'0000450C'	RS is busy.
X'00000010'	X'0000450D'	RM failed to define some of the internal fields related to the IMSRSC repository.
X'00000014'	X'00005000'	The IMPORT command was terminated, because IMODULE GETSTOR storage could not be obtained.
X'00000014'	X'00005004'	DFSOCMD response buffer could not be obtained.
X'00000014'	X'00005008'	DFSPPOOL storage could not be obtained.
X'00000014'	X'0000500C'	AWE could not be obtained.
X'00000014'	X'00005010'	Unable to obtain latch.
X'00000014'	X'00005018'	Unable to obtain a 64-bit buffer.
X'00000014'	X'00005100'	RM request error.
X'00000014'	X'00005104'	Unexpected CQS error.
X'00000014'	X'00005108'	SCI request error.
X'00000014'	X'00005110'	Repository error.
X'00000014'	X'00005115'	Import from the IMSRSC repository change list is in progress. Reissue the command after the change list processing is complete.
X'00000014'	X'0000551C'	A request to read data from the repository was rejected because a corrupted resource list was detected. The IMPORT DEFN command is rejected because the resource list is not usable. Contact IBM Software Support. Send in the output from the command in error, a dump of the local RM address space where the command executed or the dump of all RM address spaces in the IMSplex. Also, send a copy of the IMSRSC repository RID and RMD data sets.
X'00000014'	X'00005FFF'	The IMPORT command could not be processed because of an internal error.
X'02000008'	X'0000203C'	The command contains an invalid keyword parameter value.
X'02000008'	X'00003004'	No requests were successful.

## Completion codes

Errors unique to the processing of this command are returned as completion codes. The following table includes an explanation of the completion codes.

Table 341. Completion codes for the IMPORT command

Completion code	Completion code text	Meaning
0		Command completed successfully for resource or descriptor definition.

Table 341. Completion codes for the IMPORT command (continued)

Completion code	Completion code text	Meaning
6	RESOURCE NEEDS ONLINE CHANGE	The database definition was imported, but an online change is needed if the database is to be used.
9	NO IMPORT - OPTION ABORT	The resource or descriptor definition was not imported. An error occurred importing one or more resource or descriptor definitions and OPTION(ABORT) was specified.
A	NO IMPORT - SAME RSC FOUND LATER	There were multiple definitions for the same resource or descriptor on the RDDS. The last instance of the resource or descriptor definition on the RDDS was used to create the resource or descriptor.
D	RSC NEEDS STA DB ON HALDB MASTER	The HALDB master database definition was imported. In order to create the database partitions, a /STA DB or UPD DB START(ACCESS) command on the master HALDB is required.
E	IMPORT CANNOT SET DEFLT DESC	The descriptor definition was imported, but the descriptor could not be made the default descriptor as defined.
F	GLOBAL STATUS NOT APPLIED	The resource definition was imported, but global status could not be applied to the resource.
10	NO RESOURCES FOUND	Resource or descriptor name is invalid, or the wildcard parameter specified does not match any resource names.
11	RESOURCE ALREADY EXISTS	The resource or descriptor being imported already exists. No import was done.
16	INVALID CLASS VALUE	The transaction class value is invalid. The transaction definition was not imported. This should not occur if the definition was created by a valid export or by an IBM-supplied utility.
17	ANOTHER CMD IN PROGRESS	Another command (such as DELETE PGM or UPDATE PGM) is in progress for the program referenced by the transaction or routing code definition being imported.
19	CMTMODE=MULT/ WFI=Y CONFLICT	Wait-for-input WFI(Y) option conflicts with commit mode CMTMODE(MULT). The transaction definition was not imported. This should not occur if the definition was created by a valid export or by an IBM-supplied utility.
1A	IN CONVERSATION	Transaction is in conversation. The conversation ID and the terminal in conversation are returned separately. The terminal can be a static node, node and user, dynamic user, APPC luname, or OTMA tmember and tpipe.

Table 341. Completion codes for the IMPORT command (continued)

Completion code	Completion code text	Meaning
1B	CONV=Y/ CMTMODE=MULT CONFLICT	Conversation option CONV(Y) conflicts with commit mode CMTMODE(MULT). The transaction definition was not imported. This should not occur if the definition was created by a valid export or by an IBM-supplied utility.
1D	XRF TAKEOVER IS IN PROGRESS	The IMPORT DEFN SOURCE(CATALOG) command cannot be processed, because a takeover is in progress.
1E	CONV=N/SPASZ/ SPATRUNC CONFLICT	Non-conversation CONV(N) option conflicts with conversation attributes SPASZ > 0 or SPATRUNC. The transaction definition was not imported. This should not occur if the definition was created by a valid export or by an IBM-supplied utility.
1F	CONV=Y/RECOVER=N CONFLICT	Conversation CONV(Y) option conflicts with recovery RECOVER(N) option. The transaction definition was not imported. This should not occur if the definition was created by a valid export or by an IBM-supplied utility.
1F2	DIRECTORY BLDL FAILED	During processing of the IMPORT DEFN SOURCE(CATALOG) command, IMS encountered a BLDL error while reading the directory data sets.
21	CONV=Y REQUIRES SPASZ/SPATRUNC	Conversation CONV(Y) option requires conversation attributes SPASZ and SPATRUNC. The transaction definition was not imported. This should not occur if the definition was created by a valid export or by a utility supplied by IBM.
22	CPIC TRAN ALREADY EXISTS	CPI-C transaction already exists by the same name. The transaction definition was not imported.
23E	STAGING BLDL FAILED	During processing of the IMPORT DEFN SOURCE(CATALOG) command, IMS encountered a BLDL error while reading the staging data sets.
29	DOPT=Y/RESIDENT=Y CONFLICT	Dynamic DOPT(Y) option conflicts with resident RESIDENT(Y) option. The program definition was not imported. This should not occur if the definition was created by a valid export or by an IBM-supplied utility.
2A	DOPT=Y/PARALLEL CONFLICT	Dynamic DOPT(Y) option conflicts with parallel schedule SCHDTYPE(PARALLEL) option. The program definition was not imported. This should not occur if the definition was created by a valid export or by an IBM-supplied utility.
2B	INVALID EDITRTN NAME	EDITRTN name is invalid. The transaction definition was not imported. This should not occur if the definition was created by a valid export or by an IBM-supplied utility.

Table 341. Completion codes for the IMPORT command (continued)

Completion code	Completion code text	Meaning
2C	MAX 255 EDIT ROUTINES EXCEEDED	Transaction definition to be imported specifies a new edit routine name, which exceeds the limit of 255 edit routines. The transaction definition was not imported.
2D	INVALID EMHBSZ VALUE	EMH buffer size EMHBSZ value is invalid. The transaction definition was not imported. This should not occur if the definition was created by a valid export or by an IBM-supplied utility.
2F	FP=E/BMPTYPE=Y CONFLICT	Fast Path exclusive FP(E) option conflicts with BMP type BMPTYPE(Y). The program definition was not imported. This should not occur if the definition was created by a valid export or by an IBM-supplied utility.
31	NOT ALLOWED FOR A DEDB	The resident attribute in the imported database definition differs from the resident attribute in the runtime resource definition. The resident option of a DEDB cannot be changed.
32	NOT ALLOWED FOR AN MSDB	The resident attribute in the imported database definition differs from the resident attribute in the runtime resource definition. The resident option of an MSDB cannot be changed.
35	FP=E/EDITRTN CONFLICT	Fast Path exclusive FP(E) option conflicts with edit routine EDITRTN. The transaction definition was not imported. This should not occur if the definition was created by a valid export or by an IBM-supplied utility.
36	FP=E/FP=N PGM CONFLICT	Fast Path exclusive FP(E) option conflicts with program defined as non-Fast Path FP(N). The transaction definition was not imported.
37	FP=E/LANG=JAVA CONFLICT	Fast Path exclusive FP(E) option conflicts with Java language LANG(JAVA) option. The program definition was not imported. This should not occur if the definition was created by a valid export or by an IBM-supplied utility.
39	FP=E TRAN FOR RTC EXISTS	The routing code was created by IMS for a Fast Path exclusive transaction and cannot be updated with an IMPORT command. The attributes of the routing code can be updated with an UPDATE TRAN command or an IMPORT command that updates the Fast Path exclusive transaction.
3A	FP=E OR P/MS KEYWORD CONFLICT	Fast Path exclusive FP(E) or Fast Path potential FP(P) conflicts with MSC MSNAME or SIDR/SIDL value. The transaction definition was not imported. This should not occur if the definition was created by a valid export or by an IBM-supplied utility.

Table 341. Completion codes for the IMPORT command (continued)

Completion code	Completion code text	Meaning
3B	FP/MSGTYPE=MULTSEG CONFLICT	Fast Path exclusive FP(E) or Fast Path potential FP(P) conflicts with multiple segment MSGTYPE(MULTSEG) option. The transaction definition was not imported. This should not occur if the definition was created by a valid export or by an IBM-supplied utility.
3C	MAXRGN/ PARLIM=65535 CONFLICT	The maximum region count MAXRGN value conflicts with the parallel limit count PARLIM value 65535, which disables parallel processing. MAXRGN>0 is not allowed with PARLIM(65535). The transaction definition was not imported. This should not occur if the definition was created by a valid export or by an IBM-supplied utility.
3D	FP=N/EMHBSZ GT 0 CONFLICT	Non-Fast Path FP(N) conflicts with Fast Path attribute EMHBSZ>0. The transaction definition was not imported. This should not occur if the definition was created by a valid export or by an IBM-supplied utility.
3E	FP=N/FP=E PGM CONFLICT	Non-Fast Path FP(N) option conflicts with program defined as Fast Path exclusive FP(E). The transaction definition was not imported.
3F	FP=P/BMPTYPE=Y CONFLICT	Fast Path potential FP(P) option conflicts with program defined as BMP type BMPTYPE(Y). The transaction definition was not imported.
40	PARLIM/ SCHDTYPE=SERIAL CONFLICT	PARLIM value is something other than 65535 and the program is defined as SCHDTYPE(SERIAL). The transaction definition was not imported.
41	FP=E OR P/RECOVER=N CONFLICT	Fast Path exclusive FP(E) or Fast Path potential FP(P) option conflicts with recovery RECOVER(N) option. The transaction definition was not imported. This should not occur if the definition was created by a valid export or by an IBM-supplied utility.
42	FP=E OR P/RESP=N CONFLICT	Fast Path exclusive FP(E) or Fast Path potential FP(P) option conflicts with response mode RESP(N) option. The transaction definition was not imported. This should not occur if the definition was created by a valid export or by an IBM-supplied utility.
43	GPSB=Y/DOPT=Y CONFLICT	Generated PSB GPSB(Y) option conflicts with the dynamic DOPT(Y) option. The program definition was not imported. This should not occur if the definition was created by a valid export or by an IBM-supplied utility.

Table 341. Completion codes for the IMPORT command (continued)

Completion code	Completion code text	Meaning
44	TRANSACTION BUSY	The IMPORT command cannot be processed for the resource because the transaction is currently being scheduled.
45	INVALID SIDR VALUE	Remote system ID SIDR value is invalid. The transaction definition was not imported.
46	GPSB=N/LANG CONFLICT	Generated PSB option N GPSB(N) conflicts with the language option LANG(). The program definition is not imported. This should not occur if the definition was created by a valid export or by an IBM-supplied utility.
47	GPSB=Y/RESIDENT=Y CONFLICT	Generated PSB GPSB(Y) option conflicts with the resident RESIDENT(Y) option. The program definition is not imported. This should not occur if the definition was created by a valid export or by an IBM-supplied utility.
48	NOT ALLOWED FOR IMS RESOURCE	Command is not allowed for IMS descriptors or IMS resources. DFSDSDB1, DFSDSPG1, DBFDSRT1, and DFSWSTR1 are examples of IMS descriptors. DBF#FPU0 is an example of an IMS resource.
49	INQ=N/RECOVER=N CONFLICT	Inquiry INQ(N) option conflicts with recover RECOVER(N) option. The transaction definition was not imported. This should not occur if the definition was created by a valid export or by an IBM-supplied utility.
4A	IN USE	Transaction is in use. Queuing is in progress, either terminal input or a program-to-program switch.
4B	INVALID LCT VALUE	Limit count LCT value is invalid. The transaction definition was not imported. This should not occur if the definition was created by a valid export or by an IBM-supplied utility.
4D	INVALID LPRI INVALID	Limit priority LPRI value is invalid. The transaction definition was not imported. This should not occur if the definition was created by a valid export or by an IBM-supplied utility.
4E	LTERM ALREADY EXISTS	An LTERM already exists by the same name. The transaction definition was not imported.
4F	INVALID MAXRGN VALUE	Maximum region count value is invalid. The transaction definition was not imported. This should not occur if the definition was created by a valid export or by an IBM-supplied utility.
51	NO RESOURCE STRUCTURE	The resource structure is not available. The resource could not be registered with RM. The resource definition was not imported.

Table 341. Completion codes for the IMPORT command (continued)

Completion code	Completion code text	Meaning
52	RESOURCE STRUCTURE FULL	The resource structure is full. The resource could not be registered with RM. The resource definition was not imported.
55	NO FAST PATH INSTALLED	Fast Path is not installed. The resource or descriptor definition was not imported.
58	IMS NOT REGISTERED TO RM	IMS is not registered to RM.
59	MSC KEYWORD/FP=E PGM CONFLICT	MSC MSNAME or SIDR/SIDL value conflicts with program defined as Fast Path exclusive FP(E). The transaction definition was not imported.
5A	INVALID MSNAME	MSNAME is invalid. The transaction definition was not imported.
5B	MSNAME ALREADY EXISTS	An MSNAME with the same name exists. The transaction definition was not imported.
5E	MAXRGN GT 0/SERIAL=Y CONFLICT	Serial SERIAL(Y) option conflicts with maximum region count MAXRGN > 0 value. The transaction definition was not imported. This should not occur if the definition was created by a valid export or by an IBM-supplied utility.
5F	INVALID CHARACTERS IN NAME	Resource or descriptor name is invalid because it contains an invalid character. The resource or descriptor definition was not imported.
60	IMODULE GETMAIN storage error.	
61	DFSBCB STORAGE ERROR	DFSBCB storage error. Could not get storage for DDIR, PDIR, RCTE, or SMB control block. The resource or descriptor definition was not imported.
62	HIOP storage error.	
63	WKAP storage error.	
64	GETSTOR STORAGE ERROR	DFSPPOOL storage error. Could not get storage from the CMDP pool. The resource or descriptor was not imported.
65	DMB POOL STORAGE ERROR	DMB pool storage error. The command failed because of DMB pool storage request failure.
67	NO PGM DEFINED	No program PGM() defined. PGM() is required for transactions that are not remote REMOTE(N) and for routing codes. The definition was not imported. This should not occur if the definition was created by a valid export or by an IBM-supplied utility.
68	INVALID NPRI VALUE	Normal priority NPRI value is invalid. The transaction definition was not imported. This should not occur if the definition was created by a valid export or by an IBM-supplied utility.

Table 341. Completion codes for the IMPORT command (continued)

Completion code	Completion code text	Meaning
69	INVALID PARLIM VALUE	Parallel limit count PARLIM value is invalid. The transaction definition was not imported. This should not occur if the definition was created by a valid export or by an IBM-supplied utility.
6A	FP=P/FP=E PGM CONFLICT	Fast Path potential FP(P) transaction definition conflicts with program already defined as Fast Path exclusive FP(E). The transaction definition was not imported.
6B	PARLIM/SERIAL=Y CONFLICT	Serial SERIAL(Y) option conflicts with parallel limit PARLIM value. The transaction definition was not imported. This should not occur if the definition was created by a valid export or by an IBM-supplied utility.
6C	NOT ALLOWED FOR HALDB PARTITION	When processing a database definition, the IMPORT command attempted to change the resident attribute of a HALDB partition. HALDB partition definitions are not exported to an RDDS or the IMSRSC repository. When the database definition was exported, the database must not have been a HALDB partition.
6D	INVALID PROGRAM NAME	Program specified does not exist. The transaction or route code definition was not imported.
73	PROGRAM SCHEDULED	Program is scheduled.
75	INVALID PLCTIME VALUE	Parallel limit count time is invalid. The transaction definition was not imported. This should not occur if the definition was created by a valid export or by a utility supplied by IBM.
77	DEPENDENT REGIONS ACTIVE	One or more dependent regions are scheduled against the database, or one or more areas are open. The DEDB cannot be updated.
79	REMOTE/SIDR/SIDL/MSNAME CONFLICT	Remote REMOTE(Y) option requires MSC MSNAME or SIDR/SIDL value. The transaction definition was not imported. This should not occur if the definition was created by a valid export or by an IBM-supplied utility.
7A	RTC/FP=N PGM CONFLICT	The program specified is defined as non-Fast Path FP(N). The route code definition was not imported.
7E	INVALID SPASZ VALUE	SPA size is invalid. The transaction definition was not imported. This should not occur if the definition was created by a valid export or by an IBM-supplied utility.

Table 341. Completion codes for the IMPORT command (continued)

Completion code	Completion code text	Meaning
80	DATASET ERROR	<p>Function (8 characters), ddname (8 characters), return code (8 bytes), and error detail (8 characters).</p> <p>Function can be one of the following:</p> <p><b>BLDL</b> Data set BLDL error.</p> <p><b>BUSY</b> The data set is busy. The detail contains the job name that has the data set.</p> <p><b>DIR</b> Data set directory error.</p> <p><b>EMPTY</b> Library is empty.</p> <p><b>ENQUEUE</b> Data set enqueue error.</p> <p><b>EOF</b> Data set end-of-file (EOF) error.</p> <p><b>OPEN</b> Data set open error. Error detail can be 'DLS REG'.</p> <p><b>READ</b> Data set read error.</p> <p><b>WRITE</b> Data set write error.</p> <p>Ddname can be the DFSDIRC, the ddname of the IMS directory data set or DFSSTDI, the ddname of the staging data set of the IMS directory.</p> <p>Return code is the data set service return code.</p> <p>Reason code is the data set service reason code.</p>
83	INVALID SIDL VALUE	Local system ID SIDL value is invalid. The transaction definition was not imported.
85	SUSPENDED	Transaction is on the suspend queue.
87	TRAN QUEUEING	Transaction has messages queued in a non-shared-queues environment.
89	TRAN SCHEDULED	Transaction is scheduled.
8A	WILDCARD PARAMETER NOT SUPPORTED	Resource or descriptor name is invalid because it contains a wildcard parameter. The definition was not imported. This should not occur if the definition was created by a valid export or by an IBM-supplied utility.
90	INTERNAL ERROR	An IMS internal error occurred. The definition was not imported.

Table 341. Completion codes for the IMPORT command (continued)

Completion code	Completion code text	Meaning
93	RESERVED NAME	Resource or descriptor name is reserved. For example, name starts with DFS (except for DFSSAMxx or DFSIVPxx), or is an IMS reserved name such as BASICEDT or WTOR. The definition was not imported. This should not occur if the definition was created by a valid export or by an IBM-supplied utility.
94	RM REQUEST ERROR	An RM request error occurred. The resource could not be registered with RM. The resource definition was not imported.
97	DOPT=Y/LANG CONFLICT	The program was defined as DOPT(Y) and a language other than Java was defined. The program definition was not imported. This should not occur if the definition was created by a valid export or by an IBM-supplied utility.
98	CQS REQUEST ERROR	A CQS request error occurred. The resource could not be registered with RM. The resource definition was not imported.
B0	RESOURCE DEFINITION ERROR	The resource exists as another resource type in the RM structure. The resource definition was not imported.
B1	RESOURCE STATE ERROR	Resource type (8 characters), resource name (8 characters), error detail (16 characters).  The resource name is the name of the resource as it is defined to IMS, such as the database name, the program name, or the transaction name.  The error detail can be one of the following:  <b>AREA OPEN</b> An area associated with the FP DEDB is open.  <b>AREA STARTED</b> An area associated with the FP DEDB is started.  <b>DATABASE <i>dbname</i> ALTER INCOMPLETE</b> Alter processing is not complete for a HALDB database. After alter processing completes, retry the INITIATE OLC command.  Error details continued in next row.

Table 341. Completion codes for the IMPORT command (continued)

Completion code	Completion code text	Meaning
B1	RESOURCE STATE ERROR (continued)	<p><b>DBD ACTIVE</b> A /DBDUMP command is active for the database.</p> <p><b>DBR ACTIVE</b> A /DBRECOVERY command is active for the database.</p> <p><b>DEP REGS ACTIVE</b> Dependent regions using the FP DEDB are active.</p> <p>Error details continued in next row.</p>
B1	RESOURCE STATE ERROR (continued)	<p>Error detail (continued):</p> <p><b>INTERNAL ERROR</b> The resource control block is not found in the active IMS, or the resource control block exists. These are internal errors that should not occur.</p> <p><b>IN USE</b> The resource is in use.  A transaction has queuing in progress (either terminal input program-to-program switch).  A database is in use.</p> <p><b>PSB SCHEDULED</b> A program referencing a database to be changed or deleted is scheduled.</p> <p><b>QUEUEING</b> Messages are queued to the transaction or program.</p> <p><b>SCHEDULED</b> The named resource (a transaction or program to be changed or deleted, or a program referencing a database to be changed or deleted) is scheduled.</p> <p><b>SUSPENDED</b> The transaction to be changed or deleted is on the suspend queue.</p>

Table 341. Completion codes for the IMPORT command (continued)

Completion code	Completion code text	Meaning
B2	IMS STATE ERROR	<p>IMS state error (32 characters). The IMS state can be one of the following:</p> <p><b>CHECKPOINT IN PROGRESS</b> This IMS has checkpoint in progress.</p> <p><b>NOT-REACHABLE</b> The online change phase is rejected because this IMS is NOT-REACHABLE. The SCI on the OS image where this IMS is active is down. Restart the SCI and reissue the IMPORT DEFN SOURCE(CATALOG) command.</p>
B2	IMS STATE ERROR (continued)	<p>IMS state error (32 characters). The IMS state can be one of the following:</p> <p><b>RESOLVE DB INDOUBTS</b> This IMS has DB indoubts. You must resolve the DB indoubts either by reconnecting the coordinator controller and IMS or by using an operator command.</p> <p><b>RESTART IN PROGRESS</b> This IMS has restart in progress.</p>
B3	TRAN ELIGIBLE FOR SCHEDULING	Transaction is eligible for scheduling and cannot be updated. You might need to stop the transaction with an UPDATE TRAN STOP(Q,SCHD) command before attempting the import again.
B5	ROUTING CODE ALREADY EXISTS	Transaction update failed because the Fast Path exclusive FP(E) option conflicts with a routing code that already exists by that transaction name.
B6	LATCH ERROR	Unable to obtain latch. The resource or descriptor definition was not imported.
B9	LANG REQUIRED	Program is defined as GPSB(Y) but a language LANG() value was not defined. The program definition was not imported. This should not occur if the definition was created by a valid export or by an IBM-supplied utility.
BA	NOT ALLOWED FOR MRQ PSB	An attribute was defined that is not supported for the MRQ PSB. The program definition was not imported. This should not occur if the definition was created by a valid export or by an IBM-supplied utility.
BC	MAX 32767 DB EXCEEDED	32767 databases already exist. A maximum of 32767 databases are allowed. The database definition was not imported.

Table 341. Completion codes for the IMPORT command (continued)

Completion code	Completion code text	Meaning
BD	INVALID DB ACCESS TYPE	The database access type is invalid. The database definition was not imported. This should not occur if the definition was created by a valid export or by an IBM-supplied utility.
BE	INVALID CPRI VALUE	The current priority CPRI value is invalid. The transaction definition was not imported. This should not occur if the definition was created by a valid export or by an IBM-supplied utility.
D6	DATABASE IN USE	Database in use. The DEDB cannot be changed because the DEDB is in use in a region.
DE	ACBLIB READ FAILURE	An error occurred trying to read ACBLIB. The definition was not imported.
E1	DB MUST BE STOPPED AND OFFLINE	The database must be stopped and taken offline in order for the attribute to be changed. You might need to issue a /DBR DB command or an UPDATE DB STOP(ACCESS) command to stop the database and take it offline before issuing the IMPORT DEFN command.
E2	PARALLEL DB OPEN NOT COMPLETE	Restart parallel DB open is not complete. The command is not processed because the restart parallel DB open is not complete for the database.
140	NO TRAN INPUT EDIT RTN TBL	An edit routine was defined for the transaction, but an edit routine table was never built. The transaction definition was not imported.
141	LOAD FOR EDIT ROUTINE FAILED	An edit routine was defined for the transaction, but the edit routine could not be loaded. The transaction definition was not imported.
143	NO IMPORT - FPE TRAN ERROR	The routing code is associated with a Fast Path exclusive transaction. An error occurred importing the transaction, so the routing code definition was not imported.
144	NO STORAGE FOR RM REG TRAN LIST	No storage was available to build the list of transactions to register with RM. The transaction definition was not imported.
145	MESSAGE IN PROGRESS ACROSS LINK	Transaction update failed because a message for the transaction is in progress across the MSC link.
1D1	NO REPOSITORY DEFINED	No repository is defined.
1D4	REPOSITORY MEMBER IN USE	The repository member for the specified resource is in use.
1D6	REPOSITORY ERROR	Repository error.

Table 341. Completion codes for the IMPORT command (continued)

Completion code	Completion code text	Meaning
1D8	CHANGE LIST IN USE	The IMPORT DEFN SOURCE(REPO) command could not be completed because the change list for an IMS is in use and not accessible to be updated. If it is a residual change list, issue the DELETE DEFN TARGET(REPO) TYPE(CHGLIST) command to delete the change list and then retry the command, or issue the IMPORT command with the SCOPE(ACTIVE) keyword. If the IMS whose change list is not accessible is being restarted, issue the IMPORT command again.

## Examples

The following are examples of the IMPORT command:

### Example 1 for IMPORT command

TSO SPOC input:

```
IMPORT DEFN SOURCE(RDDS) RDDSDSN(IMSplex1.RDDS.DEFN)
```

TSO SPOC output:

```
Log for: IMPORT DEFN SOURCE(RDDS) RDDSDSN(IMSplex1.RDDS.DEFN)
```

```
IMSplex . . . . . : PLEX1
Routing . . . . . :
Start time . . . . : 2007.074 12:40:02.30
Stop time . . . . . : 2007.074 12:40:03.32
Return code . . . . : 00000000
Reason code . . . . : 00000000
Reason text . . . . :
Command master . . : IMS2
```

```
MbrName  Messages
-----  -
IMS1     DFS3405I IMPORT COMMAND COMPLETE FOR ALL , DSN=IMSplex1.RDDS.DEFN
IMS2     DFS3405I IMPORT COMMAND COMPLETE FOR ALL , DSN=IMSplex1.RDDS.DEFN
```

**Explanation:** The IMPORT command is routed to two IMS systems: IMS1 and IMS2. Each IMS successfully imports all of the resource and descriptor definitions from RDDS IMSplex1.RDDS.DEFN. Because OPTION(ALLRSP) is not specified, the list of imported resources is not returned.

**Note:** Each DFS3405I message is encapsulated in a single set of <msg> </msg> tags and is shown on the same TSO SPOC screen as the return and reason codes. If the message does not fit on one screen, you might need to scroll right to view the entire message.

### Example 2 for IMPORT command

TSO SPOC input:

```
IMPORT DEFN SOURCE(RDDS) RDDSDSN(IMSplex1.RDDS.DEFN) OPTION(ALLRSP)
```

TSO SPOC output:

Name	Type	MbrName	CC	ImpType
DEDBJN01	DB	IMS2	0	CREATE
DBDFLT	DBDESC	IMS2	0	CREATE
CDEBS	PGM	IMS2	0	CREATE
PGMDFLT	PGMDESC	IMS2	0	CREATE
SMQFP5A	RTC	IMS2	0	CREATE
RTCDFLT	RTCDESC	IMS2	0	CREATE
CDEBTRN1	TRAN	IMS2	0	CREATE
TRANDFLT	TRANDESC	IMS2	0	CREATE

When the **PF4** key is pressed, the following is displayed:

Log for: IMPORT DEFN SOURCE(RDDS) RDDSDSN(IMSPLEX1.RDDS.DEFN) OPTION(ALLRSP)

```

IMSpIex . . . . . : PLEX1
Routing . . . . . : IMS2
Start time. . . . : 2007.074 12:40:02.30
Stop time . . . . : 2007.074 12:40:03.32
Return code . . . : 00000000
Reason code . . . : 00000000
Reason text . . . :
Command master. . : IMS2

```

MbrName Messages

```

-----
IMS2      DFS3405I IMPORT COMMAND COMPLETE FOR ALL , DSN=IMSPLEX1.RDDS.DEFN

```

**Explanation:** The IMPORT command is sent to IMS2. All resources from RDDSDSN(IMSPLEX1.RDDS.DEFN) are imported successfully by IMS2. Because OPTION(ALLRSP) is specified, a list of imported resources is returned.

### *Example 3 for IMPORT command*

TSO SPOC input:

IMPORT DEFN SOURCE(RDDS) RDDSDSN(IMSPLEX1.RDDS.DEFN) OPTION(ALLRSP)

TSO SPOC output:

Name	Type	MbrName	CC	CCText	ImpType
PART	DB	IMS1	0		CREATE
PART	DB	IMS2	0		CREATE
PARTMAST	DB	IMS1	0		CREATE
PARTMAST	DB	IMS2	0		CREATE
TESTDB	DB	IMS1	0		CREATE
TESTDB	DB	IMS2	0		CREATE
TRND01	TRAN	IMS1	6D	INVALID PROGRAM NAME	
TRND01	TRAN	IMS2	0		CREATE
TRND11	TRAN	IMS1	0		CREATE
TRND11	TRAN	IMS2	0		CREATE
TRND123	TRAN	IMS1	0		CREATE
TRND123	TRAN	IMS2	0		CREATE
TRND01	TRANDESC	IMS1	0		CREATE
TRND01	TRANDESC	IMS2	0		CREATE

When the **PF4** key is pressed, the following is displayed:

Log for: IMPORT DEFN SOURCE(RDDS) RDDSDSN(IMSPLEX1.RDDS.DEFN)

```

IMSpIex . . . . . : PLEX1
Routing . . . . . :
Start time. . . . : 2007.074 12:40:02.30
Stop time . . . . : 2007.074 12:40:03.32
Return code . . . : 0200000C
Reason code . . . : 00003000
Reason text . . . : At least one request successful

```

Command master. . : IMS2

MbrName	Return Code	Reason Code	Description
IMS1	0000000C	00003000	At least one request successful

MbrName	Messages
IMS1	DFS3405I IMPORT COMMAND COMPLETE FOR SOME, DSN=IMSPLEX1.RDDS.DEFN
IMS2	DFS3405I IMPORT COMMAND COMPLETE FOR ALL , DSN=IMSPLEX1.RDDS.DEFN

**Explanation:** The IMPORT command is routed to two IMS systems: IMS1 and IMS2. Each IMS imports the resource and descriptor definitions from RDDS IMSPLEX1.RDDS.DEFN. IMS1 failed to import one or more resource or descriptor definitions. IMS2 imported all of the resource and descriptor definitions successfully. Transaction resource TRND01 is not imported at IMS1, because its program name does not exist at IMS1. Because OPTION(ALLRSP) is specified, a list of imported resources at IMS1 and IMS2 is returned.

#### *Example 4 for IMPORT command*

TSO SPOC input:

```
IMPORT DEFN SOURCE(RDDS) RDDS(SN(MYDSN.DEFN) OPTION(ABORT,ALLRSP)
```

TSO SPOC output:

Name	Type	MbrName	CC	CCText
PART	DB	IMS1	9	NO IMPORT - OPTION ABORT
PARTMAST	DB	IMS1	9	NO IMPORT - OPTION ABORT
TESTDB	DB	IMS1	9	NO IMPORT - OPTION ABORT
TRND01	TRAN	IMS1	6D	INVALID PROGRAM NAME
TRND11	TRAN	IMS1	9	NO IMPORT - OPTION ABORT
TRND123	TRAN	IMS1	9	NO IMPORT - OPTION ABORT
TRND01	TRANDESC	IMS1	9	NO IMPORT - OPTION ABORT

When the PF4 key is pressed, the following is displayed:

```
Log for . . : IMPORT DEFN SOURCE(RDDS) RDDS(SN(MYDSN.DEFN) OPTION(...
```

```
IMSPlex . . . . . : PLEX1
Routing . . . . . :
Start time . . . . : 2009.027 19:34:42.27
Stop time . . . . . : 2009.027 19:34:42.34
Return code . . . . : 0200000C
Reason code . . . . : 00003008
Reason text . . . . : None of the clients were successful.
Command master. . : IMS1
```

MbrName	Return Code	Reason Code	Reason text
IMS1	0000000C	00003004	No requests were successful

MbrName	Messages
IMS1	DFS3405I IMPORT COMMAND COMPLETE FOR NONE, DSN=MYDSN.DEFN

**Explanation:** The IMPORT command is sent to IMS1. IMS1 encountered an error when importing resource and descriptor definitions from RDDS MYDSN.DEFN. Transaction resource TRND01 is not imported, because its associated program name does not exist at IMS1. Because OPTION(ABORT,ALLRSP) is specified, no resources are imported at IMS1 because of the ABORT option, and a response line

is returned for each resource processed by the command because of the ALLRSP option. The resources that are not imported because of the ABORT option have a completion code of 9 and completion code text of "NO IMPORT - OPTION ABORT". If OPTION(ALLRSP) is not specified, response lines are only returned for those resources that encountered an import error, such as TRND01 in this example, and response lines are not returned for those resources that have a completion code of 9.

*Example 5 for IMPORT command*

TSO SPOC input:

```
IMPORT DEFN SOURCE(RDDS) RDDSDSN(IMSplex1.RDDS.DEFN) OPTION(ALLRSP,UPDATE)
```

TSO SPOC output:

Name	Type	MbrName	CC	ImpType
DEDBJN01	DB	IMS2	0	CREATE
DBDFLT	DBDESC	IMS2	0	CREATE
CDEBS	PGM	IMS2	0	UPDATE
PGMDFLT	PGMDESC	IMS2	0	CREATE
SMQFP5A	RTC	IMS2	0	CREATE
RTCDFLT	RTCDESC	IMS2	0	CREATE
CDEBTRN1	TRAN	IMS2	0	UPDATE
TRANDFLT	TRANDESC	IMS2	0	CREATE

When the PF4 key is pressed, the following is displayed:

```
Log for: IMPORT DEFN SOURCE(RDDS) RDDSDSN(IMSplex1.RDDS.DEFN) OPTION(ALLRSP,UPDATE)
IMSplex . . . . . : PLEX1
Routing . . . . . :
Start time. . . . : 2007.074 12:40:02.30
Stop time . . . . : 2007.074 12:40:03.32
Return code . . . : 00000000
Reason code . . . : 00000000
Reason text . . . :
Command master. . : IMS2
```

MbrName Messages

```
-----
IMS2 DFS3405I IMPORT COMMAND COMPLETE FOR ALL , DSN=IMSplex1.RDDS.DEFN
```

OM API input:

```
CMD(IMPORT DEFN SOURCE(RDDS) RDDSDSN(IMSplex1.RDDS.DEFN) OPTION(ALLRSP,UPDATE))
```

OM API output:

```
<imsout>
<ctl>
<omname>OM10M </omname>
<omvsn>1.3.0</omvsn>
<xmlvsn>20 </xmlvsn>
<statime>2009.099 00:26:38.776367</statime>
<stotime>2009.099 00:26:38.829881</stotime>
<staseq>C402495EBA42FFA0</staseq>
<stoseq>C402495EC7539267</stoseq>
<rqsttkn1>USRT011 10172638</rqsttkn1>
<rc>00000000</rc>
<rsn>00000000</rsn>
</ctl>
<cmd>
<master>IMS2 </master>
<userid>USRT011 </userid>
<verb>IMP </verb>
<kwd>DEFN </kwd>
```

```

<input>IMPORT DEFN SOURCE(RDDS) RDDSDSN(IMSPLEX1.RDDS.DEFN)
  OPTION(ALLRSP,UPDATE) </input>
</cmd>
<cmdsphdr>
<hdr s1b1="NAME" l1b1="Name" scope="LCL" sort="a" key="2" scroll="no"
  len="8" dtype="CHAR" align="left" />
<hdr s1b1="TYPE" l1b1="Type" scope="LCL" sort="a" key="1" scroll="no"
  len="8" dtype="CHAR" align="left" skipb="yes" />
<hdr s1b1="MBR" l1b1="MbrName" scope="LCL" sort="a" key="3" scroll="no"
  len="8" dtype="CHAR" align="left" />
<hdr s1b1="CC" l1b1="CC" scope="LCL" sort="n" key="0" scroll="yes"
  len="4" dtype="INT" align="right" />
<hdr s1b1="CCTXT" l1b1="CCText" scope="LCL" sort="n" key="0"
  scroll="yes" len="32" dtype="CHAR" align="left" skipb="yes" />
<hdr s1b1="IMPTYPE" l1b1="ImpType" scope="LCL" sort="n" key="0"
  scroll="yes" len="8" dtype="CHAR" align="left" skipb="yes" />
<hdr s1b1="OLDDEF" l1b1="OldDefault" scope="LCL" sort="n" key="0"
  scroll="yes" len="8" dtype="CHAR" align="left" skipb="yes" />
</cmdsphdr>
<cmdrspdata>
<rsp>NAME(CDEBTRN1) TYPE(TRAN) MBR(IMS2) CC( 0) IMPTYPE(UPDATE) </rsp>
<rsp>NAME(TRANDFLT) TYPE(TRANDESC) MBR(IMS2) CC( 0) IMPTYPE(CREATE)
</rsp>
<rsp>NAME(DEDBJN01) TYPE(DB) MBR(IMS2) CC( 0) IMPTYPE(CREATE) </rsp>
<rsp>NAME(DBDFLT ) TYPE(DBDESC) MBR(IMS2) CC( 0) IMPTYPE(CREATE)
</rsp>
<rsp>NAME(CDEBS ) TYPE(PGM) MBR(IMS2) CC( 0) IMPTYPE(UPDATE) </rsp>
<rsp>NAME(PGMDFLT ) TYPE(PGMDESC) MBR(IMS2) CC( 0) IMPTYPE(CREATE)
</rsp>
<rsp>NAME(SMQFP5A ) TYPE(RTC) MBR(IMS2) CC( 0) IMPTYPE(CREATE) </rsp>
<rsp>NAME(RTCDFLT ) TYPE(RTCDESC) MBR(IMS2) CC( 0) IMPTYPE(CREATE)
</rsp>
</cmdrspdata>
<msgdata>
<mbr name="IMS2 ">
<msg>DFS3405I IMPORT COMMAND COMPLETE FOR ALL , DSN=IMSPLEX1.RDDS.DEFN
  </msg>
</mbr>
</msgdata>
</imsout>

```

**Explanation:** The IMPORT command is sent to IMS IMS2. All resources from RDDS IMSPLEX1.RDDS.DEFN are imported successfully by IMS2. Because OPTION(ALLRSP) is specified, a list of imported resources is returned. In this example, runtime resource definitions for program CDEBS and transaction CDEBTRAN already exist. Because OPTION(UPDATE) is specified, the exiting runtime resource definitions for those resources are replaced. New runtime resource definitions are created for the remaining resources and descriptors.

### *Example 6 for IMPORT command*

TSO SPOC input:

```
IMPORT DEFN SOURCE(RDDS) RDDSDSN(MYDSN.DEFN) OPTION(UPDATE,ABORT)
```

TSO SPOC output:

Name	Type	MbrName	CC	CCText
TRND01	TRAN	SYS3	6D	INVALID PROGRAM NAME

When the PF4 key is pressed, the following is displayed:

```
Log for . . : IMPORT DEFN SOURCE(RDDS) RDDSDSN(MYDSN.DEFN) OPT... More: +>
```

```
IMSp1ex . . . . . : PLEX1
Routing . . . . . :
```

```

Start time. . . . : 2011.124 15:59:03.26
Stop time . . . . : 2011.124 15:59:03.41
Return code . . . : 0200000C
Reason code . . . : 00003000
Reason text . . . : At least one request was successful.
Command master. . : IMS2

```

MbrName	Return Code	Reason Code	Reason text
IMS1	0000000C	00003004	No requests were successful

```

MbrName Messages
-----
IMS2 DFS3405I IMPORT COMMAND COMPLETE FOR ALL , DSN=MYDSN.DEFN
IMS1 DFS3405I IMPORT COMMAND COMPLETE FOR NONE, DSN=MYDSN.DEFN

```

### OM API input:

```

CMD(IMPORT DEFN SOURCE(RDDS) RDDSDSN(MYDSN.DEFN) OPTION(UPDATE,ABORT))

```

### OM API output:

```

<imsout>
<ctl>
<omname>OM10M </omname>
<omvsn>1.5.0</omvsn>
<xmlvsn>20 </xmlvsn>
<statime>2011.124 22:59:03.358498</statime>
<stotime>2011.124 22:59:03.407697</stotime>
<staseq>C7B8B7CB3D422A2C</staseq>
<stoseq>C7B8B7CB49451F2C</stoseq>
<rqsttkn1>USRT011 10155903</rqsttkn1>
<rc>0200000C</rc>
<rsn>00003000</rsn>
<rsnmsg>CSLN023I</rsnmsg>
<rsntxt>At least one request was successful.</rsntxt>
</ctl>
<cmderr>
<mbr name="IMS1 ">
<typ>IMS </typ>
<styp>DBDC </styp>
<rc>0000000C</rc>
<rsn>00003004</rsn>
<rsntxt>No requests were successful</rsntxt>
</mbr>
</cmderr>
<cmd>
<master>IMS2 </master>
<userid>USRT011 </userid>
<verb>IMP </verb>
<kwd>DEFN </kwd>
<input>IMPORT DEFN SOURCE(RDDS) RDDSDSN(MYDSN.DEFN)
OPTION(UPDATE,ABORT) </input>
</cmd>
<cmdrsphdr>
<hdr s1b1="NAME" l1b1="Name" scope="LCL" sort="a" key="2" scroll="no"
len="8" dtype="CHAR" align="left" />
<hdr s1b1="TYPE" l1b1="Type" scope="LCL" sort="a" key="1" scroll="no"
len="8" dtype="CHAR" align="left" skipb="yes" />
<hdr s1b1="MBR" l1b1="MbrName" scope="LCL" sort="a" key="3" scroll="no"
len="8" dtype="CHAR" align="left" />
<hdr s1b1="CC" l1b1="CC" scope="LCL" sort="n" key="0" scroll="yes"
len="4" dtype="INT" align="right" skipb="no" />
<hdr s1b1="CCTXT" l1b1="CCText" scope="LCL" sort="n" key="0"
scroll="yes" len="32" dtype="CHAR" align="left" skipb="yes" />
<hdr s1b1="ERRT" l1b1="ErrorText" scope="LCL" sort="n" key="0"
scroll="yes" len="*" dtype="CHAR" skipb="yes" align="left" />
<hdr s1b1="IMPTYPE" l1b1="ImpType" scope="LCL" sort="n" key="0"
scroll="yes" len="8" dtype="CHAR" align="left" skipb="yes" />
<hdr s1b1="OLDDEF" l1b1="OldDefault" scope="LCL" sort="n" key="0"
scroll="yes" len="8" dtype="CHAR" align="left" skipb="yes" />
</cmdrsphdr>
<cmdrspdata>
<rsp>NAME(TRND01 ) TYPE(TRAN) MBR(SYS3 ) CC( 6D) CCTXT(INVALID PROGRAM NAME) </rsp>
</cmdrspdata>

```

```

<msgdata>
<mbr name="IMS2  ">
<msg>DFS3405I IMPORT COMMAND COMPLETE FOR ALL , DSN=MYDSN.DEFN
      </msg>
</mbr>
<mbr name="IMS1  ">
<msg>DFS3405I IMPORT COMMAND COMPLETE FOR NONE, DSN=MYDSN.DEFN
      </msg>
</mbr>
</msgdata>
</imsout>

```

**Explanation:** The IMPORT command is sent to two IMS subsystems: IMS1 and IMS2. IMS1 encountered an error when importing resource and descriptor definitions from RDDS MYDSN.DEFN. Transaction resource TRND01 is not imported at IMS1 because the program specified in the transaction definition does not exist on IMS1. Because OPTION(ABORT) is specified, no resources are imported at IMS1. The command is successful at IMS2. Because OPTION(ALLRSP) is not specified, a list of resource names is not returned for IMS2.

### *Example 7 for IMPORT command*

The following example consists of three separate commands.

TSO SPOC input:

```
QRY TRAN NAME(SKS1,SKS2,TRANA) SHOW(TIMESTAMP)
```

TSO SPOC output:

```

Response for: QRY TRAN NAME(SKS1,SKS2,TRANA) SHOW(TIMESTAMP)
Trancode MbrName  CC CText          LTimeAccess
SKS1     IMS1       0                2011.130 11:55:16.02
SKS2     IMS1       0
TRANA    IMS1       10 NO RESOURCES FOUND

```

```

Response for: QRY TRAN NAME(SKS1,SKS2,TRANA) SHOW(TIMESTAMP)      More: <
Trancode MbrName  LTimeUpdate      LTimeCreate      LTimeImport
SKS1     IMS1       2011.130 12:02:17.44 2011.130 11:49:11.01
SKS2     IMS1                               2011.130 11:49:11.01
TRANA    IMS1

```

**Explanation:** The QRY TRAN SHOW(TIMESTAMP) command is issued to display the current time stamp values for transactions SKS1, SKS2, and TRANA. Transactions SKS1 and SKS2 exist on IMS IMS1. An access time, an update time, and a create time are displayed for SKS1. A create time is displayed for SKS2. The import time fields are blank for both SKS1 and SKS2. Transaction TRANA does not exist on IMS IMS1.

TSO SPOC input:

```
IMPORT DEFN SOURCE(RDDS) NAME(SKS1,SKS2,TRANA) RDDSDSN(NONSYS.RDDS1) OPTION(UPDATE)
```

TSO SPOC output:

```

Response for: IMPORT DEFN SOURCE(RDDS) NAME(SKS1,SKS2,TRANA) ...
Name      Type      MbrName  CC ImpType
SKS1     TRAN      IMS1     0  UPDATE
SKS2     TRAN      IMS1     0
TRANA    TRAN      IMS1     0  CREATE

```

When the PF4 key is pressed, the following is displayed:

```

Log for . . : IMPORT DEFN SOURCE(RDDS) NAME(SKS1,SKS2,TRANA) ... More:
IMSp1ex . . . . . : PLEX1

```

```

Routing . . . . . :
Start time. . . . : 2011.130 12:07:27.27
Stop time . . . . : 2011.130 12:07:27.32
Return code . . . : 00000000
Reason code . . . : 00000000
Reason text . . . :
Command master. . : IMS1

```

MbrName Messages

```

-----
IMS1      DFS3405I IMPORT COMMAND COMPLETE FOR ALL , DSN=NONSYS.RDDS1

```

OM API input:

```

CMD(IMPORT DEFN SOURCE(RDDS) NAME(SKS1,SKS2,TRANA) RDDS(SN(NONSYS.RDDS1) OPTION(UPDATE))

```

OM API output:

```

<imsout>
<ctl>
<omname>OM10M </omname>
<omvsn>1.5.0</omvsn>
<xmlvsn>20 </xmlvsn>
<stime>2011.130 19:07:27.297021</stime>
<stotime>2011.130 19:07:27.323045</stotime>
<staseq>C7C00F37B01FDCE7</staseq>
<stoseq>C7C00F37B67A5CAC</stoseq>
<rqsttkn1>USRT011 10120727</rqsttkn1>
<rc>00000000</rc>
<rsn>00000000</rsn>
</ctl>
<cmd>
<master>IMS1 </master>
<userid>USRT011 </userid>
<verb>IMP </verb>
<kwd>DEFN </kwd>
<input>IMPORT DEFN SOURCE(RDDS) NAME(SKS1,SKS2,TRANA)
      RDDS(SN(NONSYS.RDDS1) OPTION(UPDATE) </input>
</cmd>
<cmdrsphdr>
<hdr s1b1="NAME" l1b1="Name" scope="LCL" sort="a" key="2" scroll="no"
  len="8" dtype="CHAR" align="left" />
<hdr s1b1="TYPE" l1b1="Type" scope="LCL" sort="a" key="1" scroll="no"
  len="8" dtype="CHAR" align="left" skipb="yes" />
<hdr s1b1="MBR" l1b1="MbrName" scope="LCL" sort="a" key="3" scroll="no"
  len="8" dtype="CHAR" align="left" />
<hdr s1b1="CC" l1b1="CC" scope="LCL" sort="n" key="0" scroll="yes"
  len="4" dtype="INT" align="right" skipb="no" />
<hdr s1b1="CCTXT" l1b1="CCText" scope="LCL" sort="n" key="0"
  scroll="yes" len="32" dtype="CHAR" align="left" skipb="yes" />
<hdr s1b1="ERRT" l1b1="ErrorText" scope="LCL" sort="n" key="0"
  scroll="yes" len="*" dtype="CHAR" skipb="yes" align="left" />
<hdr s1b1="IMPTYPE" l1b1="ImpType" scope="LCL" sort="n" key="0"
  scroll="yes" len="8" dtype="CHAR" align="left" skipb="yes" />
<hdr s1b1="OLDDEF" l1b1="OldDefault" scope="LCL" sort="n" key="0"
  scroll="yes" len="8" dtype="CHAR" align="left" skipb="yes" />
</cmdrsphdr>
<cmdrspdata>
<rsp>NAME(SKS1 ) TYPE(TRAN) MBR(IMS1 ) CC( 0) IMPTYPE(UPDATE)
</rsp>
<rsp>NAME(SKS2 ) TYPE(TRAN) MBR(IMS1 ) CC( 0) IMPTYPE( )
</rsp>
<rsp>NAME(TRANA ) TYPE(TRAN) MBR(IMS1 ) CC( 0) IMPTYPE(CREATE)
</rsp>
</cmdrspdata>
<msgdata>
<mbr name="IMS1 ">
<msg>DFS3405I IMPORT COMMAND COMPLETE FOR ALL , DSN=NONSYS.RDDS1

```

```

</msg>
</mbr>
</msgdata>
</imsout>

```

**Explanation:** The IMPORT DEFN command is issued to import definitions for transactions SKS1, SKS2, and TRANA from RDDS NONSYS.RDDS1. A runtime resource definition for SKS1 already exists in IMS IMS1. One or more of the attribute values in the imported definition differ from the attribute values in the existing runtime definition, so the runtime definition is replaced with the attribute values from the imported definition, and the import type is set to UPDATE. A runtime resource definition for SKS2 already exists in IMS IMS1. The attribute values in the imported definition are the same as the attribute values in the existing runtime definition, so the import type is set to UPDATE. A runtime resource definition does not exist for TRANA, so a new runtime definition is created and the import type is set to CREATE.

TSO SPOC input:

```
QRY TRAN NAME(SKS1,SKS2,TRANA) SHOW(TIMESTAMP)
```

TSO SPOC output:

```

Response for: QRY TRAN NAME(SKS1,TRANA) SHOW(TIMESTAMP)
  File Action Manage resources SPOC View Options Help
-----
PLEX1                      IMS Single Point of Control
Command ===>

----- Plex . .      Route . .      Wait . .
Response for: QRY TRAN NAME(SKS1,SKS2,TRANA) SHOW(TIMESTAMP)
Trancode MbrName      CC LTimeAccess      LTimeUpdate
SKS1     IMS1          0 2011.130 11:55:16.02 2011.130 12:07:27.29
SKS2     IMS1          0
TRANA    IMS1          0

Response for: QRY TRAN NAME(SKS1,SKS2,TRANA) SHOW(TIMESTAMP)      More: <
Trancode MbrName      LTimeCreate      LTimeImport
SKS1     IMS1          2011.130 11:49:11.01 2011.130 12:07:27.29
SKS2     IMS1          2011.130 11:49:11.01 2011.130 12:07:27.29
TRANA    IMS1          2011.130 11:55:16.02 2011.130 12:07:27.29

```

**Explanation:** After the IMPORT command is completed, the QRY TRAN SHOW(TIMESTAMP) command is issued again to display the current time stamp values for transactions SKS1, SKS2, and TRANA. The existing runtime definition for SKS1 was replaced by the IMPORT command. Both the import time stamp and the update time stamp were set to the time that the IMPORT command was received by OM. The access time stamp and create time stamp were not changed. The existing runtime definition for SKS2 was unchanged. The import time stamp for SKS2 was set to the time that the IMPORT command was received by OM. The update time stamp, access time stamp, and create time stamp were unchanged. The IMPORT command created a runtime definition for TRANA. The import time stamp was set to the time that the IMPORT command was received by OM. The create time stamp was obtained from the imported definition and stored in the new runtime definition.

#### *Example 8 for IMPORT command*

TSO SPOC input:

```
IMPORT DEFN SOURCE(REPO) NAME(TRANA,DBA,RTCA,PGMA) TYPE(ALL) SCOPE(ACTIVE)
```

TSO SPOC output:

Name	Type	MbrName	CC	ImpType
DBA	DB	IMS2	0	CREATE
DBA	DB	IMS1	0	CREATE
PGMA	PGM	IMS2	0	CREATE
PGMA	PGM	IMS1	0	CREATE
RTCA	RTC	IMS2	0	CREATE
RTCA	RTC	IMS1	0	CREATE
TRANA	TRAN	IMS2	0	CREATE
TRANA	TRAN	IMS1	0	CREATE

When the **PF4** key is pressed, the following is displayed:

Log for . . : IMPORT DEFN SOURCE(REPO) NAME(TRANA,DBA,RTCA,PGM... More: >

IMSpIex . . . . . : PLEX1  
Routing . . . . . :  
Start time . . . . : 2011.125 11:09:18.82  
Stop time . . . . . : 2011.125 11:09:19.09  
Return code . . . . : 00000000  
Reason code . . . . : 00000000  
Reason text . . . . :  
Command master . . : IMS2

MbrName Messages

-----  
IMS2 DFS3406I IMPORT COMMAND COMPLETE FOR ALL, REPOTYPE=IMSRSC, REPONAME=IMS\_REPOS  
IMS1 DFS3406I IMPORT COMMAND COMPLETE FOR ALL, REPOTYPE=IMSRSC, REPONAME=IMS\_REPOS

OM API input:

CMD(IMPORT DEFN SOURCE(REPO) NAME(TRANA,DBA,RTCA,PGMA) TYPE(ALL) SCOPE(ACTIVE))

OM API output:

```
<imsout>
<ctl>
<omname>OM10M </omname>
<omvsn>1.5.0</omvsn>
<xmlvsn>20 </xmlvsn>
<statime>2011.125 18:09:18.842449</statime>
<stotime>2011.125 18:09:19.093317</stotime>
<stoseq>C7B9B8E588A517AE</stoseq>
<stoseq>C7B9B8E5C5E45638</stoseq>
<rqsttkn1>USRT011 10110918</rqsttkn1>
<rc>00000000</rc>
<rsn>00000000</rsn>
</ctl>
<cmd>
<master>IMS2 </master>
<userid>USRT011 </userid>
<verb>IMP </verb>
<kwd>DEFN </kwd>
<input>IMPORT DEFN SOURCE(REPO) NAME(TRANA,DBA,RTCA,PGMA) TYPE(ALL)
SCOPE(ACTIVE) </input>
</cmd>
<cmdrsphdr>
<hdr s1bl="NAME" l1bl="Name" scope="LCL" sort="a" key="2" scroll="no"
len="8" dtype="CHAR" align="left" />
<hdr s1bl="TYPE" l1bl="Type" scope="LCL" sort="a" key="1" scroll="no"
len="8" dtype="CHAR" align="left" skipb="yes" />
<hdr s1bl="MBR" l1bl="MbrName" scope="LCL" sort="a" key="3" scroll="no"
len="8" dtype="CHAR" align="left" />
<hdr s1bl="CC" l1bl="CC" scope="LCL" sort="n" key="0" scroll="yes"
len="4" dtype="INT" align="right" skipb="no" />
<hdr s1bl="CCTXT" l1bl="CCText" scope="LCL" sort="n" key="0"
scroll="yes" len="32" dtype="CHAR" align="left" skipb="yes" />
<hdr s1bl="ERRT" l1bl="ErrorText" scope="LCL" sort="n" key="0"
```

```

    scroll="yes" len="*" dtype="CHAR" skipb="yes" align="left" />
<hdr s1b1="IMPTYPE" l1b1="ImpType" scope="LCL" sort="n" key="0"
    scroll="yes" len="8" dtype="CHAR" align="left" skipb="yes" />
<hdr s1b1="OLDDEF" l1b1="OldDefault" scope="LCL" sort="n" key="0"
    scroll="yes" len="8" dtype="CHAR" align="left" skipb="yes" />
</cmdrsphdr>
<cmdrspdata>
<rsp>NAME(DBA      ) TYPE(DB) MBR(IMS2  ) CC(  0) IMPTYPE(CREATE)
</rsp>
<rsp>NAME(PGMA    ) TYPE(PGM) MBR(IMS2  ) CC(  0) IMPTYPE(CREATE)
</rsp>
<rsp>NAME(RTCA    ) TYPE(RTC) MBR(IMS2  ) CC(  0) IMPTYPE(CREATE)
</rsp>
<rsp>NAME(TRANA   ) TYPE(TRAN) MBR(IMS2  ) CC(  0) IMPTYPE(CREATE)
</rsp>
<rsp>NAME(DBA      ) TYPE(DB) MBR(IMS1  ) CC(  0) IMPTYPE(CREATE)
</rsp>
<rsp>NAME(PGMA    ) TYPE(PGM) MBR(IMS1  ) CC(  0) IMPTYPE(CREATE)
</rsp>
<rsp>NAME(RTCA    ) TYPE(RTC) MBR(IMS1  ) CC(  0) IMPTYPE(CREATE)
</rsp>
<rsp>NAME(TRANA   ) TYPE(TRAN) MBR(IMS1  ) CC(  0) IMPTYPE(CREATE)
</rsp>
</cmdrspdata>
<msgdata>
<mbr name="IMS2  ">
<msg>DFS3406I  IMPORT COMMAND COMPLETE FOR ALL, REPOTYPE=IMSRSC,
    REPONAME=IMS_REPOS                                </msg>
</mbr>
<mbr name="IMS1  ">
<msg>DFS3406I  IMPORT COMMAND COMPLETE FOR ALL, REPOTYPE=IMSRSC,
    REPONAME=IMS_REPOS                                </msg>
</mbr>
</msgdata>
</imsout>

```

**Explanation:** The runtime resource definitions for the database named DBA, transaction named TRANA, program named PGMA, and routing code named RTCA are imported from the repository as the runtime resource definitions for IMS systems IMS2 and IMS3. The command is processed by all IMS systems that receive the command.

#### *Example 9 for IMPORT command*

TSO SPOC input:

```
IMPORT DEFN SOURCE(REPO) NAME(PGM002,PGM003,PGM004,PGM005,PGM006) SCOPE(ALL)
```

TSO SPOC output:

Name	Type	MbrName	CC	ImpType	IMSid
PGM002	PGM	IMS1	0	CREATE	
PGM002	PGM	IMS1	0	CHGLIST	IMS2
PGM002	PGM	IMS1	0	CHGLIST	IMS3
PGM002	PGM	IMS1	0	CHGLIST	IMS4
PGM002	PGM	IMS1	0	CHGLIST	IMS5
PGM003	PGM	IMS1	0	CHGLIST	IMS3
PGM003	PGM	IMS1	0	CHGLIST	IMS2
PGM003	PGM	IMS1	0	CHGLIST	IMS4
PGM003	PGM	IMS1	0	CREATE	
PGM003	PGM	IMS1	0	CHGLIST	IMS5
PGM004	PGM	IMS1	0	CHGLIST	IMS2
PGM004	PGM	IMS1	0	CHGLIST	IMS4
PGM004	PGM	IMS1	0	CHGLIST	IMS3
PGM004	PGM	IMS1	0	CREATE	
PGM004	PGM	IMS1	0	CHGLIST	IMS5
PGM005	PGM	IMS1	0	CHGLIST	IMS4

```

PGM005  PGM      IMS1      0 CHGLIST  IMS2
PGM005  PGM      IMS1      0 CREATE
PGM005  PGM      IMS1      0 CHGLIST  IMS3
PGM005  PGM      IMS1      0 CHGLIST  IMS5
PGM006  PGM      IMS1      0 CHGLIST  IMS2
PGM006  PGM      IMS1      0 CHGLIST  IMS3
PGM006  PGM      IMS1      0 CREATE
PGM006  PGM      IMS1      0 CHGLIST  IMS4
PGM006  PGM      IMS1      0 CHGLIST  IMS5

```

When the **PF4** key is pressed, the following is displayed:

```

Log for . . . : IMPORT DEFN SOURCE(REPO) NAME(PGM002,PGM003,PGM004,PGM005,PGM006) SCOPE(ALL)

IMSpIex . . . . . : PLEX1
Routing . . . . . :
Start time . . . . : 2013.189 16:06:50.58
Stop time . . . . . : 2013.189 16:06:50.59
Return code . . . . : 00000000
Reason code . . . . : 00000000
Reason text . . . . :
Command master . . : IMS1

MbrName  Messages
-----  -
IMS1     DFS3406I IMPORT COMMAND COMPLETE FOR ALL, REPOTYPE=IMSRSC, REPONAME=IMS_REPOS

```

OM API input:

```

CMD(IMPORT DEFN SOURCE(REPO) NAME(PGM002,PGM003,PGM004,PGM005,PGM006) SCOPE(ALL))

```

OM API output:

```

<?xml version="1.0"?>
<!DOCTYPE imsout SYSTEM "imsout.dtd">
<imsout>
<ctl>
<omname>OM10M </omname>
<omvsn>1.6.0</omvsn>
<xmlvsn>20 </xmlvsn>
<statime>2013.189 23:00:48.499819</statime>
<stotime>2013.189 23:00:48.545610</stotime>
<staseq>CBA184CA6A86B806</staseq>
<stoseq>CBA184CA75B4AD94</stoseq>
<rqsttkn1>USRT005 10160048</rqsttkn1>
<rc>00000000</rc>
<rsn>00000000</rsn>
</ctl>
<cmd>
<master>IMS1 </master>
<userid>USRT005 </userid>
<verb>IMP </verb>
<kwd>DEFN </kwd>
<input>IMPORT DEFN SOURCE(REPO)
NAME(PGM002,PGM003,PGM004,PGM005,PGM006) TYPE(PGM) SCOPE(ALL) </input>
</cmd>
<cmdsphdr>
<hdr s1b1="NAME" l1b1="Name" scope="LCL" sort="a" key="2" scroll="no"
len="8" dtype="CHAR" align="left" />
<hdr s1b1="TYPE" l1b1="Type" scope="LCL" sort="a" key="1" scroll="no"
len="8" dtype="CHAR" align="left" skipb="yes" />
<hdr s1b1="MBR" l1b1="MbrName" scope="LCL" sort="a" key="3" scroll="no"
len="8" dtype="CHAR" align="left" />
<hdr s1b1="CC" l1b1="CC" scope="LCL" sort="n" key="0" scroll="yes"
len="4" dtype="INT" align="right" skipb="no" />
<hdr s1b1="CCTXT" l1b1="CCText" scope="LCL" sort="n" key="0"
scroll="yes" len="32" dtype="CHAR" align="left" skipb="yes" />
<hdr s1b1="ERRT" l1b1="ErrorText" scope="LCL" sort="n" key="0"
scroll="yes" len="*" dtype="CHAR" skipb="yes" align="left" />
<hdr s1b1="IMPTYPE" l1b1="ImpType" scope="LCL" sort="n" key="0"

```

```

    scroll="yes" len="8" dtype="CHAR" align="left" skipb="yes" />
<hdr s1b1="IMSID" l1b1="IMSid" scope="LCL" sort="n" key="0"
    scroll="yes" len="4" dtype="CHAR" align="left" skipb="yes" />
</cmdrsphdr>
<cmdrspdata>
<rsp>NAME(PGM002 ) TYPE(PGM) MBR(IMS1 ) CC( 0) IMPTYPE(CREATE)
</rsp>
<rsp>NAME(PGM003 ) TYPE(PGM) MBR(IMS1 ) CC( 0) IMPTYPE(CREATE)
</rsp>
<rsp>NAME(PGM004 ) TYPE(PGM) MBR(IMS1 ) CC( 0) IMPTYPE(CREATE)
</rsp>
<rsp>NAME(PGM005 ) TYPE(PGM) MBR(IMS1 ) CC( 0) IMPTYPE(CREATE)
</rsp>
<rsp>NAME(PGM006 ) TYPE(PGM) MBR(IMS1 ) CC( 0) IMPTYPE(CREATE)
</rsp>
<rsp>NAME(PGM002 ) TYPE(PGM) MBR(IMS1 ) CC( 0) IMSID(IMS2 )
IMPTYPE(CHGLIST) </rsp>
<rsp>NAME(PGM003 ) TYPE(PGM) MBR(IMS1 ) CC( 0) IMSID(IMS2 )
IMPTYPE(CHGLIST) </rsp>
<rsp>NAME(PGM004 ) TYPE(PGM) MBR(IMS1 ) CC( 0) IMSID(IMS2 )
IMPTYPE(CHGLIST) </rsp>
<rsp>NAME(PGM005 ) TYPE(PGM) MBR(IMS1 ) CC( 0) IMSID(IMS2 )
IMPTYPE(CHGLIST) </rsp>
<rsp>NAME(PGM006 ) TYPE(PGM) MBR(IMS1 ) CC( 0) IMSID(IMS2 )
IMPTYPE(CHGLIST) </rsp>
<rsp>NAME(PGM002 ) TYPE(PGM) MBR(IMS1 ) CC( 0) IMSID(IMS3 )
IMPTYPE(CHGLIST) </rsp>
<rsp>NAME(PGM003 ) TYPE(PGM) MBR(IMS1 ) CC( 0) IMSID(IMS3 )
IMPTYPE(CHGLIST) </rsp>
<rsp>NAME(PGM004 ) TYPE(PGM) MBR(IMS1 ) CC( 0) IMSID(IMS3 )
IMPTYPE(CHGLIST) </rsp>
<rsp>NAME(PGM005 ) TYPE(PGM) MBR(IMS1 ) CC( 0) IMSID(IMS3 )
IMPTYPE(CHGLIST) </rsp>
<rsp>NAME(PGM006 ) TYPE(PGM) MBR(IMS1 ) CC( 0) IMSID(IMS3 )
IMPTYPE(CHGLIST) </rsp>
<rsp>NAME(PGM002 ) TYPE(PGM) MBR(IMS1 ) CC( 0) IMSID(IMS4 )
IMPTYPE(CHGLIST) </rsp>
<rsp>NAME(PGM003 ) TYPE(PGM) MBR(IMS1 ) CC( 0) IMSID(IMS4 )
IMPTYPE(CHGLIST) </rsp>
<rsp>NAME(PGM004 ) TYPE(PGM) MBR(IMS1 ) CC( 0) IMSID(IMS4 )
IMPTYPE(CHGLIST) </rsp>
<rsp>NAME(PGM005 ) TYPE(PGM) MBR(IMS1 ) CC( 0) IMSID(IMS4 )
IMPTYPE(CHGLIST) </rsp>
<rsp>NAME(PGM006 ) TYPE(PGM) MBR(IMS1 ) CC( 0) IMSID(IMS4 )
IMPTYPE(CHGLIST) </rsp>
<rsp>NAME(PGM002 ) TYPE(PGM) MBR(IMS1 ) CC( 0) IMSID(IMS5 )
IMPTYPE(CHGLIST) </rsp>
<rsp>NAME(PGM003 ) TYPE(PGM) MBR(IMS1 ) CC( 0) IMSID(IMS5 )
IMPTYPE(CHGLIST) </rsp>
<rsp>NAME(PGM004 ) TYPE(PGM) MBR(IMS1 ) CC( 0) IMSID(IMS5 )
IMPTYPE(CHGLIST) </rsp>
<rsp>NAME(PGM005 ) TYPE(PGM) MBR(IMS1 ) CC( 0) IMSID(IMS5 )
IMPTYPE(CHGLIST) </rsp>
<rsp>NAME(PGM006 ) TYPE(PGM) MBR(IMS1 ) CC( 0) IMSID(IMS5 )
IMPTYPE(CHGLIST) </rsp>
</cmdrspdata>
<msgdata>
<mbr name="IMS1 ">
<msg>DFS3406I IMPORT COMMAND COMPLETE FOR ALL, REPOTYPE=IMSRSC,
REPONAME=IMS_REPOS </msg>
</mbr>
</msgdata>
</imsout>

```

**Explanation:** IMS1 is active. IMS2, IMS3, IMS4, and IMS5 are not active. IMS1 through IMS5 have programs PGM002 through PGM006 defined in their resource

lists. Because SCOPE(ALL) is specified, the change lists of the IMS systems that are down are updated with the resources that are being imported. The IMPORT command imports five programs, which are created for IMS1 and added to the change lists of IMS2, IMS3, IMS4, and IMS5. The IMSID column identifies the IMS whose change list was created or updated.

*Example 10 for IMPORT command*

TSO SPOC input:

```
IMPORT DEFN SOURCE(REPO) NAME(DB0010) TYPE(DB) SCOPE(ALL) OPTION(ALLRSP)
```

TSO SPOC output:

Name	Type	MbrName	CC	CCText	ImpType	IMSid
DB0010		IMS1	10	NO RESOURCES FOUND		
DB0010	DB	IMS1	0		CHGLIST	IMS5

When the PF4 key is pressed, the following is displayed:

```
Log for . . . : IMPORT DEFN SOURCE(REPO) NAME(DB0010) TYPE(DB) SCOPE(ALL) OPTION(ALLRSP)
```

```
IMSpIex . . . . . : PLEX1
Routing . . . . . :
Start time. . . . : 2013.189 16:18:35.27
Stop time . . . . : 2013.189 16:18:35.33
Return code . . . : 0200000C
Reason code . . . : 00003008
Reason text . . . : None of the clients were successful.
Command master. . : IMS1
```

MbrName	Return Code	Reason Code	Reason text
IMS1	0000000C	00003008	At least one request successful

```
MbrName Messages
-----
IMS1 DFS3407E IMPORT COMMAND COMPLETE FOR SOME, REPOTYPE=IMSRSC, REPONAME=IMS_REPOS
```

OM API input:

```
CMD(IMPORT DEFN SOURCE(REPO) NAME(DB0010) TYPE(DB) SCOPE(ALL) OPTION(ALLRSP))
```

OM API output:

```
<?xml version="1.0"?>
<!DOCTYPE imsout SYSTEM "imsout.dtd">
<imsout>
<ctl>
<omname>OM10M </omname>
<omvsn>1.6.0</omvsn>
<xmlvsn>20 </xmlvsn>
<statime>2013.189 23:18:35.272816</statime>
<stotime>2013.189 23:18:35.332778</stotime>
<staseq>CBA188C3C5270282</staseq>
<stoseq>CBA188C3D3CAA981</stoseq>
<rqsttkn1>USRT005 10161835</rqsttkn1>
<rc>0200000C</rc>
<rsn>00003008</rsn>
<rsnmsg>CSLN054I</rsnmsg>
<rsntxt>None of the clients were successful.</rsntxt>
</ctl>
<cmderr>
<mbr name="IMS1 ">
<typ>IMS </typ>
<styp>DBDC </styp>
<rc>0000000C</rc>
```

```

<rsn>00003000</rsn>
<rsntxt>At least one request successful</rsntxt>
</mbr>
</cmderr>
<cmd>
<master>IMS1 </master>
<userid>USRT005 </userid>
<verb>IMP </verb>
<kwd>DEFN </kwd>
<input>IMPORT DEFN SOURCE(REPO) NAME(DB0010) TYPE(DB) SCOPE(ALL)
        OPTION(ALLRSP) </input>
</cmd>
<cmdrsphdr>
<hdr s1bl="NAME" l1bl="Name" scope="LCL" sort="a" key="2" scroll="no"
    len="8" dtype="CHAR" align="left" />
<hdr s1bl="TYPE" l1bl="Type" scope="LCL" sort="a" key="1" scroll="no"
    len="8" dtype="CHAR" align="left" skipb="yes" />
<hdr s1bl="MBR" l1bl="MbrName" scope="LCL" sort="a" key="3" scroll="no"
    len="8" dtype="CHAR" align="left" />
<hdr s1bl="CC" l1bl="CC" scope="LCL" sort="n" key="0" scroll="yes"
    len="4" dtype="INT" align="right" skipb="no" />
<hdr s1bl="CCTXT" l1bl="CCText" scope="LCL" sort="n" key="0"
    scroll="yes" len="32" dtype="CHAR" align="left" skipb="yes" />
<hdr s1bl="ERRT" l1bl="ErrorText" scope="LCL" sort="n" key="0"
    scroll="yes" len="*" dtype="CHAR" skipb="yes" align="left" />
<hdr s1bl="IMPTYPE" l1bl="ImpType" scope="LCL" sort="n" key="0"
    scroll="yes" len="8" dtype="CHAR" align="left" skipb="yes" />
<hdr s1bl="IMSID" l1bl="IMSid" scope="LCL" sort="n" key="0"
    scroll="yes" len="4" dtype="CHAR" align="left" skipb="yes" />
</cmdrsphdr>
<cmdrspdata>
<rsp>NAME(DB0010 ) TYPE(DB) MBR(IMS1 ) CC( 0) IMSID(IMS5 )
    IMPTYPE(CHGLIST) </rsp>
<rsp>NAME(DB0010 ) MBR(IMS1 ) CC( 10) CCTXT(NO RESOURCES FOUND)
</rsp>
</cmdrspdata>
<msgdata>
<mbr name="IMS1 ">
<msg>DFS3407E IMPORT COMMAND COMPLETE FOR SOME, REPOTYPE=IMSRSC,
    REPONAME=IMS_REPOS </msg>
</mbr>
</msgdata>
</imsout>

```

**Explanation:** IMS1 is active, and IMS5 is down. The IMPORT command is routed to IMS1, which does not have database DB0010 in its resource list. The repository has database DB0010, which is defined for IMS5 but not IMS1. Because SCOPE(ALL) is specified, the change list of the IMS systems that are down are updated with the resources that are being imported. IMS1, the command master, reports that IMS5's change list is updated for DB0010, but it does not import anything itself. A response line with a completion code of X'10', no resources found, is returned for IMS1 because DB0010 is not defined for IMS1 in the repository.

**Related concepts:**

-  How to interpret CSL request return and reason codes (System Programming APIs)
-  CSL RM, IMS, and Repository Server termination (System Administration)
-  IMSRSC repository administration (System Administration)
-  Change lists for the IMSRSC repository (System Definition)

**Related reference:**

Chapter 2, "Command keywords and their synonyms," on page 63



---

## Chapter 36. INITIATE commands

Use the INITIATE commands to initiate the online change process or online reorganization of the specified HALDB partitions.

INITIATE commands are:

- “INITIATE OLC command”
- “INITIATE OLREORG command” on page 936

---

### INITIATE OLC command

Use the INITIATE OLC command to initiate the global online change process. The resources that can be changed online are areas, databases, DMBs, Fast Path DEDB databases, MFS formats, programs, routing codes, and transactions. The correct online change command sequence is INITIATE OLC PHASE(PREPARE) followed by INITIATE OLC PHASE(COMMIT).

#### Restrictions:

- The INITIATE OLC command is not supported if local online change is enabled.
- This command is invalid on XRF alternate, RSR tracker, and FDBR systems.
- INIT OLC TYPE(MODBLKS) does not apply or is not supported when dynamic resource definition (DRD) is enabled by MODBLKS=DYN in the <COMMON\_SERVICE\_LAYER> section of the DFSDFxxx PROCLIB member. To change MODBLKS resources in an online system when DRD is enabled, use the IMS type-2 commands CREATE, UPDATE, and DELETE.
- INIT OLC TYPE(ACBLIB) and INIT OLC TYPE(ACBMBR) do not apply or are not supported when the IMS management of ACBs is enabled by ACBGMT=CATALOG in the <CATALOG> section of the DFSDFxxx PROCLIB member. To change ACBs in an online system when IMS manages ACBs, submit the changes by using either DDL or one of the IMS catalog population utilities and, if the changes are not activated automatically, issuing the IMPORT DEFN SOURCE(CATALOG).

#### Subsections:

- “Environment”
- “Syntax” on page 892
- “Keywords” on page 892
- “Usage notes” on page 900
- “Output fields” on page 903
- “Return, reason, and completion codes” on page 905
- “Examples” on page 922

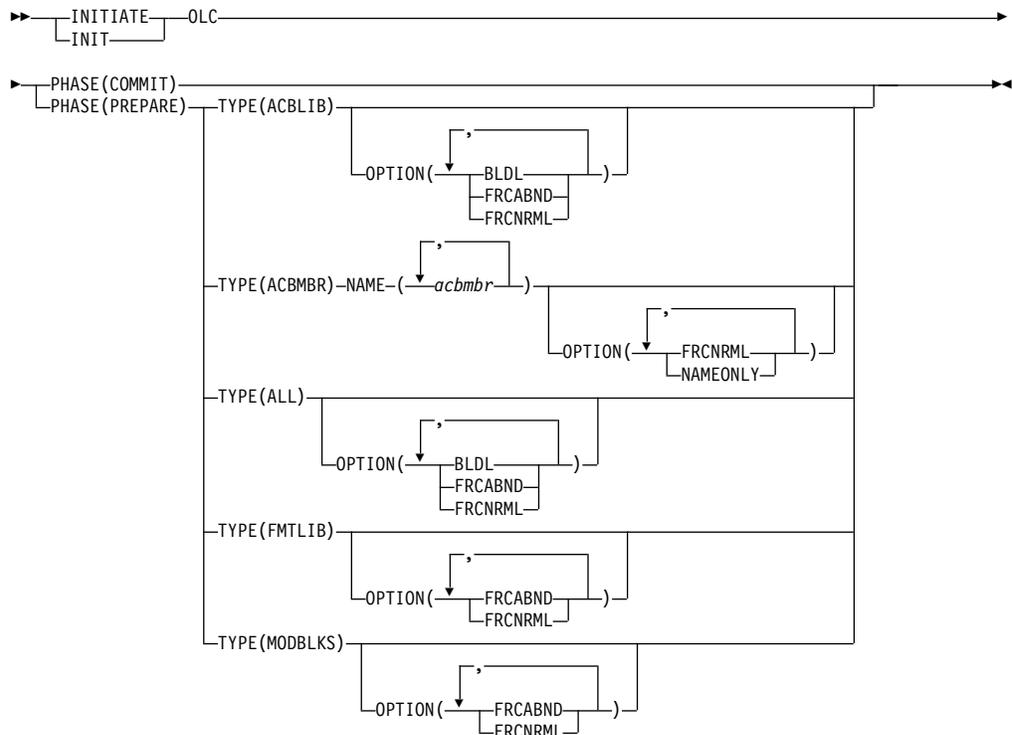
#### Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) from which the INITIATE command and keywords can be issued.

Table 342. Valid environments for the INITIATE OLC command and keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
INITIATE OLC	X	X	X
OPTION	X	X	X
PHASE	X	X	X
TYPE	X	X	X

## Syntax



## Keywords

The following keywords are valid for the INITIATE OLC command:

### OPTION()

Specifies an additional function to be performed during an online change. Following is a list of additional functions:

**BLDL** Sends asynchronous information message DFS3498I to the system console for each PSB or DMB not found in ACBLIB. This message does not affect the success of the command.

### FRCABND

Forces the online change prepare phase even if one or more required IMS systems in the IMSplex have ended abnormally. An IMS is required to participate in the global online change if it is listed in the OLCSTAT data set as being current with the online change libraries. An IMS is listed in the OLCSTAT data set because it either participated in the last global online change or cold started since the last global online change. A successful global online

change specified with the FRCABND removes an abnormally ended IMS from the list of required IMS systems in the OLCSTAT data set.

If the INITIATE OLC PHASE(PREPARE) command is specified with the FRCABND keyword and the command fails for any IMS, you can proceed with an INITIATE OLC PHASE(COMMIT) command, if you want, after canceling those IMS systems where prepare failed. Otherwise, you must cancel (abort) the online change and start over.

#### **FRCNRML**

Forces the online change prepare phase even if one or more required IMS systems in the IMSplex have terminated normally. An IMS is required to participate in the global online change if it is listed in the OLCSTAT data set as being current with the online change libraries. An IMS is listed in the OLCSTAT data set because it either participated in the last global online change or cold started since the last global online change. A successful global online change specified with the FRCNRML keyword removes an IMS that shut down normally from the list of required IMS systems in the OLCSTAT data set.

If the INITIATE OLC PHASE(PREPARE) command is specified with the FRCNRML keyword and the command fails for any IMS, you can proceed with an INITIATE OLC PHASE(COMMIT) command, if you want, after shutting down those IMS systems where prepare failed. Otherwise, you must terminate the online change and start over.

#### **NAMEONLY**

Specifies that only the DBDs and PSBs that are specified in the NAME keyword are to be processed.

By default, when a DBD is specified on the NAME keyword of the INITIATE OLC PHASE(PREPARE) TYPE(ACBMBR) command, the ACB member online change (OLC) function automatically copies all associated PSBs and externally referenced DBDs from the staging ACB library to active ACB library. Depending on the number of PSBs and DBDs, the amount of time required for the copying process can be significant.

In many cases, you can significantly reduce the amount of time the ACB member online change function takes by specifying OPTION(NAMEONLY) on the INITIATE OLC PHASE(PREPARE) TYPE(ACBMBR) command. The NAMEONLY option limits the processing of the ACB member online change function to only the DBDs and PSBs that are specified in NAME() keyword of the command.

If a PSB is specified in the NAME keyword, the following rules apply:

- Regardless of whether a PSB is new to the active ACB library or not, if the DBDs in the intent list have not been modified, this PSB is copied to the active ACB library.
- If a PSB is not new to the active ACB library and the DBDs in the intent list have been modified, this PSB is not copied to the active ACB library and a completion code 12B is returned.

If a DBD is specified in the NAME keyword, the following rules apply:

- If a DBD is new to the active ACB library and the DBDs in the reference list have not been modified, this DBD is copied to the active ACB library.
- If a DBD is new to the active ACB library and the DBDs in the reference list have been modified, this DBD is not copied to the active ACB library and a completion code 12C is returned.
- If a DBD is not new to the active ACB library and has been modified, this DBD is not copied to the active ACB library and a completion code 12A is returned.

#### **NAME(*acmbr*)**

Specifies that one or more ACB library members are to be copied from the staging ACB library to the active ACB library. *acmbr* can be a DBD or a PSB that is being changed or added.

Wildcard characters are not supported in the member names.

The maximum number of DBDs and PSBs that can be specified on the *acmbr* parameter is limited to 100.

If a DBD is being changed and OPTION(NAMEONLY) is not specified, you do not have to specify the associated PSBs on the command because all of the PSBs that are associated with the changed DBD are copied automatically from the staging ACB library to the active ACB library.

If a DBD that is being changed or added has external references and OPTION(NAMEONLY) is not specified, the secondary index DBD does not have to be specified on the *acmbr* parameter. The INIT OLC TYPE(ACBMBR) command processing copies all externally referenced members of the DBD from the staging ACB library to the active ACB library.

#### **PHASE()**

Specifies the online change phase to start. The PREPARE phase must be performed first, followed by the COMMIT phase.

#### **PREPARE**

Performs the online change prepare phase on each IMS system listed in the OLCSTAT data set by specifying which resources are to be added, changed, and deleted. PREPARE is not recovered across an emergency restart and must be reentered after restart if the COMMIT phase did not complete before IMS failure.

The prepare phase is rejected if any IMS systems in the IMSplex current with the online change libraries are down, unless the FRCABND or the FRCNRML option is used. The OLCSTAT data set lists the IMS systems that are current with the online change libraries. These IMS systems either participated in the last global online change or cold started after the last global online change. To force a global online change in spite of abended IMS systems, use the FRCABND option. To force a global online change in spite of IMS systems that are shut down normally, use the FRCNRML option. The prepare phase fails if any IMS participating in the global online change prepare phase fails, detects an error, or fails to respond in time. Zero, one, or more of the IMS systems remain in an online change prepare complete state. Issue the QUERY MEMBER TYPE(IMS) SHOW(STATUS) command to display the

online change state of all the IMS systems in the IMSplex. Evaluate the QUERY MEMBER TYPE(IMS) output to help you decide whether to issue the INITIATE OLC PHASE(PREPARE) command to try the PREPARE phase again, or to issue the TERMINATE OLC command to terminate the online change. The IMS systems that are in an online change state remain in an online change state until you terminate or commit the online change. IMS does not automatically terminate online change because of a failure. IMS leaves the IMS systems in their online change states. You must issue the QUERY MEMBER TYPE(IMS) SHOW(STATUS) command and evaluate the output to decide whether to terminate the online change or attempt the online change phase again.

**Restriction:** The INITIATE OLC PHASE(PREPARE) command is rejected if the IMS to which the command is routed does not support global online change. If this situation occurs and there is an IMS that supports global online change, you must route the command to a specific IMS that supports global online change.

## COMMIT

Performs the online change commit phase on each IMS listed in the OLCSTAT data set, which commits the online changes by bringing all the newly defined resources online, updating changed resources, and removing deleted resources. The commit phase consists of commit phase 1, the OLCSTAT data set update, commit phase 2, and commit phase 3. The OLCSTAT data set is updated with the new current online change libraries and the list of IMS systems that are current with the current online change libraries. The commit phase 2 switches the online environment from the active ACBLIB, FORMAT, or MODBLKS libraries to the inactive libraries containing the new or changed resource descriptions.

**Recommendation:** Use the /DISPLAY MODIFY command to display the work in progress for resources to be changed or deleted, before attempting the COMMIT phase.

When COMMIT is successful, the modifications persist across all IMS restarts, unless global online change occurs while this IMS is down.

The commit phase fails if any IMS participating in the global online change commit phase fails, detects an error, or fails to respond in time. Zero, one, or more of the IMS systems might be in an online change prepare complete state, a commit phase 1 complete state, or no longer in an online change state because commit phase 2 succeeded. Issue the QUERY MEMBER TYPE(IMS) SHOW(STATUS) command to display the online change state of all the IMS systems in the IMSplex. Evaluate the QUERY MEMBER TYPE(IMS) output to help you decide whether to issue the INITIATE OLC PHASE(COMMIT) command to try the commit again, or issue the TERMINATE OLC command to terminate the online change. The IMS systems that are in an online change state remain in an online change state until you terminate or commit the online change.

IMS does not automatically terminate online change because of a failure. IMS leaves the IMS systems in their online change states. You must issue the QUERY MEMBER TYPE(IMS) SHOW(STATUS)

command and evaluate the output to decide whether to terminate the online change or attempt the online change phase again.

If a database is deleted, IMS closes the database and makes it unavailable to programs. Also, if the database is authorized for DBRC, COMMIT deauthorizes it.

If FMTLIB is being changed, most new and in-progress requests are queued for processing after the INITIATE OLC PHASE(COMMIT) command is completed.

Most new and in-progress requests are queued for processing.

An INITIATE OLC PHASE(COMMIT) command is rejected if:

- An INITIATE OLC PHASE(PREPARE) command was not previously entered.
- Commit phase 1 fails on one or more IMS systems in the IMSplex.
- Transactions to be deleted have messages queued.
- Transactions to be changed by the following system definition keywords have messages queued:
  - MSGTYPE
  - INQUIRY
  - FPATH
  - EDIT
  - SPA
  - SERIAL
- Transactions to be changed by keywords not shown in the previous bullet have messages queues and the transactions have not been the object of a /STOP or /PSTOP command.
- Transactions with access to programs or databases to be changed or deleted are prevented from updating and the transactions have not been the object of a /STOP or /PSTOP command. When a transaction shows a status of USTOP during a /DISPLAY TRAN command, the /STO command might have to be entered because the /PSTOP command might not enable the INITIATE OLC command to complete the transaction.
- Any program (PSB) that is currently scheduled has a program, database, or transaction change or deletion. This includes Wait for Input (WFI) and Fast Path transactions. All WFI and Fast Path regions that reference changed or deleted routing codes, programs, or databases must be stopped before entering the INITIATE OLC PHASE(COMMIT).
- An I/O error occurs while reading the inactive MFS FORMAT library directory or the resident index (\$\$IMSDIR) member.
- A /START DATABASE command is in progress for any database that is changed or deleted.
- A /DBDUMP or /DBRECOVERY command is in progress for any database that is changed or deleted.
- An IMS that participated in the prepare phase was shut down normally or ended abnormally. The online change must be aborted and started over.

- OM routed the command to an IMS where local online change is enabled.
- The OLCSTAT data set is not locked for global online change. The INITIATE OLC PHASE(PREPARE) command locks the OLCSTAT data set, so that no IMS systems can initialize during the global online change. The commit command is rejected if the lock is not set. The INITIATE OLC PHASE(PREPARE) command might not have completed successfully or the OLCSTAT data set contents might be invalid.
- An IFP region is running. The INITIATE OLC PHASE(COMMIT) command is rejected because an active route code exists. All IFP regions must be terminated before issuing a COMMIT.

IMS does not reject the INITIATE OLC PHASE(COMMIT) command if the online change indirectly affects transactions (having work in progress) that reference resources in the ACB library or in the MODBLKS data set that are directly affected by the online change. These transactions themselves are not being changed (no attribute on the TRANSACT macro is being changed), but they reference a PDIR, PSB, or DMB that is affected by the online change. A transaction can be indirectly affected by an online change in one of the following situations:

- The transaction references a PSB directory (PDIR) control block in the IMS.MODBLKS data set that is being changed.
- The transaction references a PSB in the ACB library that is being changed.
- The transaction references a PSB directory (PDIR) control block in the IMS.MODBLKS data set which in turn references a database that is being changed online.
- The transaction references a PSB in the ACB library (IMS.ACBLIB) that references a database that is being changed online.

Operator action might be required to prevent the preceding conditions. For example, if a program or database is scheduled, the operator must either wait until the program or database is finished before the INITIATE OLC PHASE(COMMIT) command is entered or issue the /STOP, /PSTOP, or UPDATE TRAN command for the associated transaction.

If IMS is in an IMSplex and Resource Manager (RM) is using a resource structure, the transactions being added by the online change are registered to RM if they are not already registered to RM. If any error occurs during the CSLRMUPD request to register the transactions, the /MODIFY or INITIATE OLC command fails and IMS will be in a COMMIT2F state.

If an INITIATE OLC PHASE(COMMIT) command results in the IMS remaining in a COMMIT2F state, a completion code of the error that caused it is returned to OM. If the error can be fixed, the INITIATE OLC PHASE(COMMIT) can be reissued and CSLRMUPD tried again and the online change completed. If the error cannot be fixed, the IMS must be canceled and warm started as the online change is not complete.

## TYPE()

Specifies the types of resources that are changed online. The resources are as follows:

### ACBLIB

Causes IMS to prepare to add, change, or delete database descriptors (DMB) and program descriptors (PSB) to ACBLIB. The online environment is switched from the active ACBLIB to the inactive ACBLIB containing new or changed DMB and PSBs. ACBLIB members that are not found are displayed as CHANGED because a DDIR or PDIR exists in MODBLKS. Additions or deletions to ACBLIB require the MODBLKS parameter. Quiesces transactions to be changed or deleted and transactions that access databases or programs to be changed or deleted. Fast Path DEDBs are supported but Fast Path MSDBs are not supported.

ACBLIB is disabled if IMS management of ACBs is enabled for IMS. If IMS management of ACBs is enabled on all of the IMS systems, the INITIATE OLC TYPE(ACBLIB) command fails. If IMS management of ACBs is enabled on some of the IMS systems, the INITIATE OLC PHASE(PREPARE) TYPE(ACBLIB) command succeeds with a warning return code indicating that the command was not applicable to some IMS systems. The IMS systems where online change for ACBLIB is enabled, perform the ACBLIB online change. The IMS systems where IMS management of ACBs is enabled, return a completion code of 1 (not applicable).

If the INITIATE OLC PHASE(PREPARE) command is specified, with type ACBLIB and MODBLKS and/or FMTLIB, the OLC is performed as follows:

- IMS systems where ACBLIB online change is enabled, perform the ACBLIB OLC.
- IMS systems where IMS management of ACBs is enabled, ignore the ACBLIB keyword.
- IMS systems where MODBLKS online change is enabled, perform the MODBLKS OLC.
- IMS systems where DRD is enabled, ignore the MODBLKS keyword.
- IMS systems where FMTLIB is enabled, perform FMTLIB OLC.
- DBCTL systems ignore the FMTLIB keyword. DBCTL IMS systems with DRD and IMS management of ACBs enabled ignore the MODBLKS, FMTLIB and ACBLIB keywords and no OLC is performed.

### ACBMBR

Specifies that an ACB library member online change process is to be performed for the specified ACB library member name or names. The command response lists all the members that are to be copied from the staging ACB library to the active ACB library during the COMMIT command processing.

When you issue either an INIT OLC PHASE(PREPARE) TYPE(ALL) command or an INIT OLC PHASE(PREPARE) TYPE(ACBLIB) command, IMS initiates a full online change process for either the ACB library or the ACB, FMT, and MODBLKS libraries. In either case, an ACB library member online change process is not involved.

After an INITIATE OLC PHASE(COMMIT) command, when the TYPE(ACBMBR) online change completes successfully, the active and inactive ACB library suffixes and the MODID are not updated in the OLCSTAT data set or in the IMS control blocks because ACB library member online change processing changes or adds ACB members directly into the active ACB library.

ACBMBR OLC is disabled if IMS management of ACBs is enabled for IMS. If IMS management of ACBs is enabled on all the IMS systems, the INITIATE OLC TYPE(ACBMBR) command fails. If IMS management of ACBs is enabled on some IMS systems, the INITIATE OLC PHASE(PREPARE) TYPE(ACBMBR) command succeeds with a warning return code indicating that the command was not applicable to some IMS systems. The IMS systems where online change for ACBLIB is enabled, perform the ACBMBR online change. The IMS systems where IMS management of ACBs is enabled return a completion code of 1 (not applicable).

**ALL** Prepares to add, change, and delete resources for the ACBLIB, FMTLIB, or MODBLKS libraries. The BLDL function is performed for ACBLIB. ALL does not include the ACBMBR online change.

If the INITIATE OLC PHASE(PREPARE) command is specified with ALL, the OLC is performed as follows:

- IMS systems where ACBLIB online change is enabled, perform the ACBLIB OLC.
- IMS systems where IMS management of ACBs is enabled, ignore the ACBLIB keyword.
- IMS systems where MODBLKS online change is enabled, perform the MODBLKS OLC.
- IMS systems where DRD is enabled, ignore the MODBLKS keyword.
- IMS systems where FMTLIB is enabled, perform FMTLIB OLC.
- DBCTL systems ignore the FMTLIB keyword. DBCTL IMS systems with DRD and IMS management of ACBs enabled ignore the MODBLKS, FMTLIB and ACBLIB keywords and no OLC is performed.

### **FMTLIB**

Causes IMS to prepare to add, change, or delete MFS maps in the FMTLIB library that are produced by the MFS Language utility and Service utility. The online environment is switched from the active FMTLIB to the inactive FMTLIB containing new MFS maps produced by the MFS Language utility and Service utility. The FMTLIB parameter is ignored with DBCTL. There are no restrictions on the use of MFS format blocks until the INITIATE OLC PHASE(COMMIT) command is entered.

If the INITIATE OLC PHASE(PREPARE) command is specified with type FMTLIB and MODBLKS and/or ACBLIB, the OLC is performed as follows:

- IMS systems where ACBLIB online change is enabled, perform the ACBLIB OLC.
- IMS systems where IMS management of ACBs is enabled, ignore the ACBLIB keyword.

- IMS systems where MODBLKS online change is enabled, perform the MODBLKS OLC.
- IMS systems where DRD is enabled, ignore the MODBLKS keyword.
- IMS systems where FMTLIB is enabled, perform FMTLIB OLC.
- DBCTL systems ignore the FMTLIB keyword. DBCTL IMS systems with DRD and IMS management of ACBs enabled ignore the MODBLKS, FMTLIB and ACBLIB keywords and no OLC is performed.

## MODBLKS

Causes IMS to prepare to add, change, or delete database, transaction, program, or RTCODE resources in the MODBLKS library.

The online environment is switched from the active MODBLKS library to the inactive MODBLKS library. The MODBLKS library contains a subset of control blocks produced by IMS system definition. Transactions to be changed or deleted and transactions that access databases or programs to be changed or deleted are quiesced.

MODBLKS is disabled if DRD is enabled for IMS. If DRD is enabled on all the IMS systems, the INITIATE OLC TYPE(MODBLKS) command fails. If DRD is enabled on some IMS systems, the INITIATE OLC PHASE(PREPARE) TYPE(MODBLKS) command succeeds with a warning return code indicating that the command was not applicable to some IMS systems. The IMS systems where online change for MODBLKS is enabled perform the MODBLKS online change. The IMS systems where DRD is enabled return a completion code of 1 (not applicable).

If the INITIATE OLC PHASE(PREPARE) command is specified, with type MODBLKS and FMTLIB and/or ACBLIB, the OLC is performed as follows:

- IMS systems where ACBLIB online change is enabled perform the ACBLIB OLC.
- IMS systems where IMS management of ACBs is enabled ignore the ACBLIB keyword.
- IMS systems where MODBLKS online change is enabled perform the MODBLKS OLC.
- IMS systems where DRD is enabled ignore the MODBLKS keyword.
- IMS systems where FMTLIB is enabled perform FMTLIB OLC.
- DBCTL systems ignore the FMTLIB keyword. DBCTL IMS systems with DRD and IMS management of ACBs enabled ignore the MODBLKS, FMTLIB and ACBLIB keywords and no OLC is performed.

## Usage notes

The INITIATE OLC command can be issued to an IMSplex by using the Batch SPOC utility.

If an IMSplex is running with a Resource Manager (RM) environment (RMENV=Y in the DFSCGxxx PROCLIB member), the INITIATE OLC command initiates a

phase of global online change of resources and coordinates the phase with all of the IMS systems in the IMSplex. The INITIATE OLC command is similar to the command sequence /MODIFY PREPARE and /MODIFY COMMIT, except that it applies to an IMSplex-wide global online change. OM sends the INITIATE OLC command to one master IMS in the IMSplex. The command master IMS uses RM services to coordinate the online change with all participating IMS systems.

If the command master IMS in an IMSplex is running without an RM environment (RMENV=N in the DFSCGxxx PROCLIB member), the INITIATE OLC command is performed locally at the IMS to which the command was sent. The OLCSTAT data set must be unique for each IMS in the IMSplex that is running without an RM environment. The OLCSTAT data set cannot be shared between IMS systems and must be unique. If the OLCSTAT data set contains the name of an IMS other than the one processing the online change, the INITIATE OLC command is rejected. You can use the QUERY OLC command to display the contents of the OLCSTAT data set to determine which IMS member name is invalid. To correct the data set, you can use the Global Online Change utility, DFSUOLC0.

If the INITIATE OLC command is issued with a route list containing either the default Route All or multiple IMS systems that do not have RM running, the INITIATE OLC command must be issued separately to each IMS. To perform online change for IMS systems that do not have RM running, you must issue the INITIATE OLC command to each IMS. To determine which IMS systems are defined with RMENV=N in the DFSCGxxx PROCLIB member, you can issue the QUERY MEMBER SHOW(ATTRIB) command.

If some IMS systems in an IMSplex use RM services and some do not, you must ensure that the OLCSTAT data sets are properly defined to the IMS systems. The IMS systems that use RM services and global online change must share one OLCSTAT data set. Each IMS system that does not use RM services must have its own unique OLCSTAT data set.

| When all of the IMS systems return the completion code of 1 (CMD NOT  
| APPLICABLE), the online change cannot be committed as there is nothing to  
| commit. You must issue a TERM OLC to terminate the OLC phase in progress.  
| IMS systems remain in OLCPREPC state until the OLC is terminated.

If the INITIATE OLC PHASE(PREPARE) command is specified without an FRCABND or FRCNRML keyword and the command fails because one or more IMS systems are down or go down before the online change is committed, the online change must be aborted and started over. Issue the TERMINATE OLC command to terminate the online change.

The INITIATE OLC command master usually performs the online change phase locally first. If the online change phase fails locally, the command master usually skips sending the online change phase to the other IMS systems, sets a completion code for each other IMS indicating that the online change phase was not attempted, and terminates command processing. However, if the INITIATE OLC PHASE(COMMIT) command fails on the local IMS because of work in progress for resources that are directly affected by the online change, the command master still sends the commit phase 1 to the other IMS systems. The purpose is to report work in progress for all the IMS systems in the IMSplex, to facilitate completion of the work in progress.

In a mixed IMSplex, you might have some variations of IMS systems that support a particular type of online change and some that do not. At least one IMS in the

IMSplex must successfully perform the online change phase for the command to be considered successful. If no IMS in the IMSplex supports the type of online change that the command is attempting, the command reason code indicates that none of the IMS systems performed the online change phase. If you enter an INITIATE OLC PHASE(PREPARE) command that does not apply to any IMS in the IMSplex, you must terminate the online change with a TERMINATE OLC command.

If the INITIATE OLC PHASE(COMMIT) command fails for any IMS before the OLCSTAT data set is updated, either correct the errors and try the commit again or terminate the online change with a TERMINATE OLC command. If the INITIATE OLC PHASE(COMMIT) command fails for any IMS after the OLCSTAT data set has been updated, correct the errors and try the commit again. The online change cannot be terminated.

If an IMS ends abnormally during online change and the INITIATE OLC PHASE(PREPARE) command was not specified with FRCABND, then issue the TERMINATE OLC command to terminate the online change. The INITIATE OLC PHASE(COMMIT) command is not permitted in this case. If an IMS ends abnormally during online change and the INITIATE OLC PHASE(PREPARE) command was specified with FRCABND, then the INITIATE OLC PHASE(COMMIT) command is permitted.

Type-1 and type-2 commands that come from the OM interface are rejected during the commit phase, if the command changes resources. Commands that change resources could interfere with the online change of the resources. Type-1 and type-2 commands that come from the OM interface are permitted during the commit phase, if the command displays resources such as QUERY or /DISPLAY. Type-1 commands that are entered from the system console or an IMS terminal are queued during the online change commit phase. These commands run after the online change is committed or aborted.

Each IMS participating in the global online change does not issue the same synchronous online change messages to the master terminal or system console that it does for a local online change. The OM command response contains information equivalent to the online change messages that appear for the local online change, such as the DFS3499 message contents.

Each IMS participating in the global online change might issue asynchronous online change messages to the system console, such as DFS3400, DFS3445, and DFS3498.

The INITIATE OLC PHASE(PREPARE) TYPE(ALL) command ignores the following resource definitions on IMS systems where dynamic resource definition is enabled: database directories (DDIRs), program directories (PDIRs), Fast Path routing codes (RCTEs), and scheduler message blocks (SMBs, which are also known as transactions). INITIATE OLC PHASE(PREPARE) specified with TYPE MODBLKS alone or with other keywords is rejected on the IMS systems where dynamic resource definition is enabled, but processed on the IMS systems where dynamic resource definition is not enabled. The INITIATE OLC PHASE(COMMIT) command master updates the OLCSTAT data set to switch the MODBLKS data set, even in the case where the master has dynamic resource definition enabled and the MODBLKS online change does not apply. The MODBLKS data set is switched in the OLCSTAT data set in case another IMS in the IMSplex does not have dynamic resource definition enabled. If the INITIATE OLC PHASE(PREPARE) command is

specified with TYPE MODBLKS and there is only one IMS in the IMSplex and it has dynamic resource definition enabled, the command is rejected and the OLCSTAT data set is not updated.

The OM command timeout default of 300 seconds (5 minutes) might not be enough time for the online change phase to complete. You might have to specify a timeout value on the command based on the needs of the installation. To change the timeout value, use the TIMEOUT parameter of the CSLOMCMD command request or specify a Wait (timeout) value when you issue the INITIATE OLC command from TSO SPOC.

An online change of a PSB or DBD removes the member from the 31-bit pool as well as from the 64-bit pool.

If a resident program is changed, it becomes nonresident locally because its definition in the resident PSB pool is no longer valid. If its resident definition attribute was not changed, it becomes resident again at the next IMS restart. If a nonresident program is changed to be resident, the residency takes effect at the next IMS restart.

## Output fields

The following table shows the output fields for an INITIATE OLC command. The columns in the table are as follows:

### Short label

Contains the short label generated in the XML output.

### Keyword

Identifies the keyword on the command that caused the field to be generated. N/A appears for output fields that are always returned.

### Meaning

Provides a brief description of the output field.

*Table 343. INITIATE OLC output field descriptions*

Short label	Keyword	Meaning
ACBL	TYPE(ACBLIN,FMTLIB,MODBLKS,ALL)	The current ACBLIB library suffix (if prepare succeeded or commit succeeded that did not include ACBLIB), or the new ACBLIB library suffix (if commit succeeded for ACBLIB). The ACBLIB suffix is returned if the online change prepare phase or commit phase is successful. Suffix A means that ACBLIBA is current. Suffix B means that ACBLIBB is current.
ACBS	TYPE(ACBMBR)	Value of an IMS system.
ADD	TYPE(ACBMBR)	Resource added. The ACB member is found in the staging ACBLIB and not in the active ACBLIB. The ACB member is to be added to the active ACBLIB.
CC	N/A	Completion code returned by IMS member that performed the global online change phase. Completion code is always returned.
CCTXT	<i>error</i>	Completion code text that briefly explains the meaning of the nonzero completion code.

Table 343. INITIATE OLC output field descriptions (continued)

Short label	Keyword	Meaning
CHG	TYPE(ACBMBR)	Resource changed. The ACB member is found in the staging ACBLIB and the active ACBLIB. The ACB member is to be changed in the active ACBLIB.
CPY	TYPE(ACBMBR)	Resource copied. The ACB member is found in the staging ACBLIB and not in the active ACBLIB. The ACB member does not have a PDIR for a PSB or a DDIR for a DBD in an IMS system. The ACB member is to be copied from the staging to the active ACBLIB. The ACB resource is not available until a PDIR for a PSB or a DDIR for a DBD is created.
DBD	TYPE(ACBMBR)	DBD resource name
ERRT	N/A	Error text returned by IMS member that failed performing the global online change phase. Error text might be returned if the completion code is nonzero.
FMTL	TYPE(ACBLIN,FMTLIB,MODBLKS,ALL)	The current FMTLIB library suffix (if prepare succeeded or commit succeeded that did not include FMTLIB), or the new FMTLIB library suffix (if commit succeeded for FMTLIB). The FMTLIB suffix is returned if the online change prepare phase or commit phase is successful. Suffix A means that FMTLIBA is current. Suffix B means that FMTLIBB is current.
IMSMBR	N/A	IMS member that performed the global online change phase. The IMS member name is always returned.
MBR	N/A	IMSplex member that built the line of output. IMS identifier of the IMS that is the master of this online change phase. IMS identifier is always returned.
MODB	TYPE(ACBLIN,FMTLIB,MODBLKS,ALL)	The current MODBLKS library suffix (if prepare succeeded or commit succeeded that did not include MODBLKS), or the new MODBLKS library suffix (if commit succeeded for MODBLKS). The MODBLKS suffix is returned if the online change prepare phase or commit phase is successful.  Suffix A means that MODBLKSA is current. Suffix B means that MODBLKSB is current.
MODI	TYPE(ACBLIN,FMTLIB,MODBLKS,ALL)	The current modify ID (if prepare succeeded), or new modify ID (if commit succeeded). The modify ID - 1 indicates the number of global online changes that have been performed. The modify ID is returned if the online change prepare phase or commit phase is successful.
PSB	TYPE(ACBMBR)	PSB resource name
RSC	TYPE(ACBSHR)	Resource type is not known.
RFS	TYPE(ACBSHR)	Resource is refreshed in the active ACBLIB.

## Return, reason, and completion codes

Two sets of return and reason codes are returned when a command is issued through the OM API. One set of codes is for the OM request and the other set of codes is for the command itself. The OM request return and reason codes that can be returned as a result of the INITIATE OLC command are standard for all commands entered through the OM API.

The return and reason codes returned by the INITIATE OLC command are passed from the OM API on to the SPOC application. These codes are encapsulated in XML tags similar to how the response to the command is encapsulated in XML tags. The IBM-supplied TSO SPOC displays these codes, whereas a user-written SPOC application might choose not to display these codes.

Some reason codes are accompanied by a complete list of IMS systems and return codes. The reason code meaning indicates whether a list is returned. A partial list of IMS systems and return codes might be returned with any INITIATE OLC error reason code, if any output was built before the error was detected.

The following table includes the return and reason codes and a brief explanation of the codes. The return or reason code returned for the command might also indicate an error from a CSL request.

*Table 344. Return and reason codes for INITIATE OLC*

Return code	Reason code	Meaning
X'00000000'	X'00000000'	<p>The INITIATE OLC command completed successfully. The INITIATE OLC command applies to all of the IMS systems listed in the OLCSTAT data set. Each IMS listed in the OLCSTAT data set is current with the online change libraries and required to participate in the online change. The FRCABND option was not specified. The FRCNRML option was not specified, either.</p> <p>If the INITIATE OLC PHASE(PREPARE) command was specified, all of the IMS systems listed in the OLCSTAT data set are now in an online change prepare state.</p> <p>If the INITIATE OLC PHASE(COMMIT) command was specified, all of the IMS systems listed in the OLCSTAT data set successfully committed the online change.</p> <p>An output line is built for each IMS listed in the OLCSTAT data set. Each output line contains the IMS member name and a completion code of zero.</p>

Table 344. Return and reason codes for INITIATE OLC (continued)

Return code	Reason code	Meaning
X'00000004'	X'0000100C'	<p>The command completed successfully, but was not applicable to one or more IMS systems for acceptable reasons. The INITIATE OLC command applies to all of the IMS systems listed in the OLCSTAT data set. Each IMS listed in the OLCSTAT data set is current with the online change libraries and required to participate in the online change (unless FRCABND or FRCNRML is specified).</p> <p>An output line is built for each IMS listed in the OLCSTAT data set. Each output line contains the IMS member name and a completion code. A nonzero completion code might be accompanied by error text. One or more of the IMS systems contain a completion code indicating that the online change phase did not apply to this IMS, such as the IMS state is shutdown and FRCSHUT was specified, the IMS state is abended and FRCABND was specified, the online change type does not apply to this IMS, or this IMS is already in the correct online change state. The INITIATE OLC completion code table contains the list of completion codes and error text that can be returned by the INITIATE OLC command.</p>
X'00000004'	X'0000100D'	<p>The specified online change type is not applicable to all the IMS systems in the OLCSTAT. Because the online change cannot be committed, use the TERMINATE OLC command to terminate the online change in progress. This error return and reason code combination is returned in the following situations:</p> <ul style="list-style-type: none"> <li>• If a MODBLKS type of online change process is attempted in an IMSplex where all IMS systems are DRD enabled</li> <li>• If an FMMLIB type of online change process is attempted in an IMSplex where all IMS systems are configured as DBCTL systems</li> </ul>
X'00000008'	X'00002040'	<p>The INITIATE OLC command is rejected because of an invalid filter. The INITIATE OLC PHASE(PREPARE) OPTION(BLDL) command can be specified only with TYPE(ACBLIB) or TYPE(ALL). The INITIATE OLC PHASE(COMMIT) command does not support TYPE or OPTION.</p>
X'00000008'	X'00004300'	<p>The command is not allowed, because online change for the MODBLKS data set is disabled (DFSDFxxx or DFSCGxxx is defined with MODBLKS=DYN). Dynamic resource definition is enabled.</p>

Table 344. Return and reason codes for INITIATE OLC (continued)

Return code	Reason code	Meaning
X'0000000C'	X'00003000'	<p>The INITIATE OLC command is successful for at least one IMS but not all IMS systems. The INITIATE OLC command applies to all of the IMS systems listed in the OLCSTAT data set. Each IMS listed in the OLCSTAT data set is current with the online change libraries and required to participate in the online change (unless FRCABND or FRCNRML is specified).</p> <p>An output line is built for each IMS listed in the OLCSTAT data set. Each output line contains the IMS member name and a completion code. A nonzero completion code might be accompanied by error text. One or more of the IMS systems returned an error completion code. The INITIATE OLC completion code table contains the list of completion codes and error text that can be returned by the INITIATE OLC command.</p> <p>If INITIATE OLC PHASE(PREPARE) fails, the online change must be aborted and started over.</p> <p>If INITIATE OLC PHASE(COMMIT) fails and you want to try the commit again, correct the problem, and issue the INITIATE OLC PHASE(COMMIT) command again.</p> <p>To abort the online change, issue the TERMINATE OLC command. If the INITIATE OLC PHASE(PREPARE) command failed or the INITIATE OLC PHASE(COMMIT) command failed before the commit master updated the OLCSTAT data set, the online change is aborted.</p> <p>See 'INITIATE OLC error handling' for more details.</p>

Table 344. Return and reason codes for INITIATE OLC (continued)

Return code	Reason code	Meaning
X'0000000C'	X'00003004'	<p>The INITIATE OLC command failed for all of the IMS systems. The INITIATE OLC command applies to all of the IMS systems listed in the OLCSTAT data set. Each IMS listed in the OLCSTAT data set is current with the online change libraries and required to participate in the online change (unless FRCABND or FRCNRML is specified).</p> <p>An output line is built for each IMS listed in the OLCSTAT data set. Each output line contains the IMS member name and a completion code. A nonzero completion code might be accompanied by error text. The INITIATE OLC completion code table contains the list of completion codes and error text that can be returned by the INITIATE OLC command.</p> <p>If INITIATE OLC PHASE(PREPARE) fails for all of the IMS systems listed in the OLCSTAT data set, no IMS is in an online change state. You can try the online change again with another INITIATE OLC PHASE(PREPARE) command.</p> <p>If INITIATE OLC PHASE(COMMIT) fails and you want to try the commit again, correct the problem, and then issue the INITIATE OLC PHASE(COMMIT) command again.</p> <p>To terminate the online change, issue the TERMINATE OLC command. If the INITIATE OLC PHASE(PREPARE) command failed or the INITIATE OLC PHASE(COMMIT) command failed before the commit master updated the OLCSTAT data set, the online change is terminated.</p> <p>For more details, see "INITIATE OLC error handling" on page 933.</p>
X'00000010'	X'00004004'	<p>The INITIATE OLC command failed because there is no CQS. RM attempted to access the process resource on the resource structure, but it failed because CQS is not available. The online change phase might have succeeded on one or more IMS systems.</p> <p>For more details, see "INITIATE OLC error handling" on page 933.</p>
X'00000010'	X'0000400C'	The INITIATE OLC command failed because it is invalid for an XRF alternate.
X'00000010'	X'00004014'	The INITIATE OLC command failed because it is invalid for an RSR tracker.
X'00000010'	X'0000401C'	The INITIATE OLC command failed because it is invalid for an FDBR region.
X'00000010'	X'00004018'	<p>The INITIATE OLC command failed because the RM resource structure is not available. The online change phase might have succeeded on one or more IMS systems.</p> <p>For more details, see "INITIATE OLC error handling" on page 933.</p>
X'00000010'	X'00004100'	The INITIATE OLC PHASE(PREPARE) command is rejected because the resource structure is full. RM failed trying to create the process resource on the resource structure.

Table 344. Return and reason codes for INITIATE OLC (continued)

Return code	Reason code	Meaning
X'00000010'	X'00004104'	<p>The INITIATE OLC command failed because RM is not available. The online change phase might have succeeded on one or more IMS systems. Either there is no RM address space, or RM is active but not registered to SCI because CQS or the resource structure is not available.</p> <p>For more details, see "INITIATE OLC error handling" on page 933.</p>
X'00000010'	X'00004108'	<p>The INITIATE OLC command failed because SCI is not available. The online change phase might have succeeded on one or more IMS systems.</p> <p>For more details, see "INITIATE OLC error handling" on page 933.</p>
X'00000010'	X'0000410C'	<p>The INITIATE OLC command is rejected, because global online change is not enabled. Local online change is enabled. Use the /MODIFY command for local online change. If your IMSplex is made up of some IMS systems that support global online change and some that support local online change, route the INITIATE OLC command to an IMS that is enabled for global online change.</p>
X'00000010'	X'0000410D'	<p>Online change prepare has already been done. Another prepare command is not allowed. The only commands that are allowed when IMS is in this state are INITIATE OLC PHASE(COMMIT) or TERMINATE OLC.</p>
X'00000010'	X'0000410E'	<p>Online change prepare has not been done. An online change prepare command must complete successfully before a commit command is attempted.</p>
X'00000010'	X'0000410F'	<p>The online change has already been committed, so it cannot be terminated. However, an error occurred that prevented the commitment from being fully completed. For example, one of the IMS systems that participates in the online change was unable to complete commit phase 2 or commit phase 3, or an IMS response to RM timed out and the commit master could not determine whether the online change commit succeeded on the other IMS systems. Another commit command is required to complete the online change commitment.</p>
X'00000010'	X'00004110'	<p>The INITIATE OLC command is rejected, because the command does not apply to the online change state of the command master.</p> <p>INITIATE OLC PHASE(COMMIT) is rejected if the command master is not in an online change prepare state.</p> <p>INITIATE OLC PHASE(COMMIT) is rejected if routed to an IMS that already successfully completed commit phase 3.</p> <p>INITIATE OLC PHASE(PREPARE) is rejected if the command master is already in an online change prepare state.</p> <p>For more details, see "INITIATE OLC error handling" on page 933.</p>

Table 344. Return and reason codes for INITIATE OLC (continued)

Return code	Reason code	Meaning
X'00000010'	X'00004111'	The INITIATE OLC command failed because the command master is not in the OLCSTAT data set.
X'00000010'	X'00004112'	The PREPARE command cannot be processed, because the OLCSTAT is locked.
X'00000010'	X'00004114'	The INITIATE OLC command failed because of an error accessing the OLCSTAT data set. The online change phase might have succeeded on one or more IMS systems.  A DFS2843 message is sent to the OM output exit as unsolicited output.  For more details, see "INITIATE OLC error handling" on page 933.
X'00000010'	X'00004118'	The INITIATE OLC command failed because of an error allocating the OLCSTAT data set. The online change phase might have succeeded on one or more IMS systems.  A DFS2848 message is sent to the OM output exit as unsolicited output.  For more details, see "INITIATE OLC error handling" on page 933.
X'00000010'	X'0000411C'	The INITIATE OLC command failed because of an error in the OLCSTAT data set contents. One or more of the values is invalid.  A DFS2844 message is sent to the OM output exit as unsolicited output.
X'00000010'	X'00004120'	The INITIATE OLC command is rejected because an online change command is already in progress on this IMS. The command in progress might be INITIATE OLC, TERMINATE OLC, or /DISPLAY MODIFY.
X'00000010'	X'00004124'	The INITIATE OLC PHASE(PREPARE) command is rejected, because another process of the same type is already in progress. Processes include another OLC command or a DRD command such as CREATE DB or DELETE DB for a DEDB. Wait until the other process has completed.
X'00000010'	X'0000412C'	The OLCSTAT data set contains the name of an IMS other than the IMS processing the online change. Use DFSUOLC0 to correct the data set.
X'00000010'	X'00004303'	At least one IMS in the OLCSTAT is at a lower level than required for the TYPE(ACBMBR) online change.
X'00000010'	X'00004304'	Data set version is invalid. The OLCSTAT is initialized to an invalid version for the type of online change requested. For example, version 1 is invalid for TYPE(ACBMBR) member online change. Or, the data set is initialized to version 2 and one or more IMS systems are lower than IMS Version 10.
X'00000010'	X'00004305'	IMS version mismatch. There is a mismatch between the version of the IMS in the OLCSTAT and the actual version of IMS that is currently running. The actual version of IMS is returned from the CSLSCQRY request. The OLCSTAT must be reinitialized so that the IMS is at the correct version.

Table 344. Return and reason codes for INITIATE OLC (continued)

Return code	Reason code	Meaning
X'00000010'	X'0000501C'	The INITIATE OLC command is rejected because an IMODULE GETMAIN storage request failed.
X'00000014'	X'00005000'	The INITIATE OLC command is rejected because an IMODULE GETSTOR storage request failed.
X'00000014'	X'00005004'	The INITIATE OLC command failed because a DFSOCMD response buffer could not be obtained. The online change phase might have succeeded on one or more IMS systems.  For more details, see "INITIATE OLC error handling" on page 933.
X'00000014'	X'0000500C'	The INITIATE OLC command failed because an AWE could not be obtained.
X'00000014'	X'00005100'	The INITIATE OLC command failed because of an RM error. The online change phase might have succeeded on one or more IMS systems. The RM error might be caused by a resource structure failure that causes RM to lose knowledge of an IMSplex-wide process.  For more details, see "INITIATE OLC error handling" on page 933.
X'00000014'	X'00005104'	The INITIATE OLC command failed because of a CQS error. The online change phase might have succeeded on one or more IMS systems.  For more details, see "INITIATE OLC error handling" on page 933.
X'00000014'	X'00005108'	The INITIATE OLC command failed because of an SCI error. The online change phase might have succeeded on one or more IMS systems.  For more details, see "INITIATE OLC error handling" on page 933.
X'00000014'	X'0000510C'	Another RM process step is in progress. If this reason code is returned after the previous COMMIT command timed out, try the COMMIT command again.
X'00000014'	X'00005FFF'	The INITIATE OLC command failed because of an internal IMS error. The online change phase might have succeeded on one or more IMS systems.  For more details, see "INITIATE OLC error handling" on page 933.

The INITIATE command (with the OLC keyword) can result in errors that leave one or more of the IMS systems in the IMSplex in various online change states. The following table contains the possible completion codes that can be returned as a result of issuing an INITIATE OLC command. The table also includes a brief explanation of the code and provides the error text if applicable.

Table 345. Completion codes for the INITIATE OLC command

Completion code	Meaning	ERROR TEXT (uppercase)
0	The online change prepare phase or commit phase completed successfully.	

Table 345. Completion codes for the INITIATE OLC command (continued)

Completion code	Meaning	ERROR TEXT (uppercase)
1	The online change type does not apply to this IMS. For example, an FMTLIB online change does not apply to a DBCTL IMS. This IMS keeps status indicating that it is participating in the online change. This IMS is included in the OLCSTAT data set IMS list after a successful commit phase 1. For another example, a MODBLKS online change does not apply to an IMS where dynamic resource definition is enabled (DFSDfxxx or DFSCGxxx is defined with MODBLKS=DYN).	
2	The online change phase was not attempted by this IMS for one of the following reasons: <ul style="list-style-type: none"> <li>• The online change phase master encountered an error and did not direct this IMS to perform the online change phase.</li> <li>• The online change phase master rejected the online change because one or more IMS systems are down and the FRCNRML or FRCABND keyword was not specified on the INITIATE OLC PHASE(PREPARE) command. The online change phase master did not direct this IMS to perform the online change phase.</li> </ul>	
3	This IMS is already in the correct online change state. The INITIATE OLC command must have been tried again after a previous INITIATE OLC command resulted in a mix of successes and failures. This IMS keeps status indicating that it is participating in the online change. This IMS is included in the OLCSTAT data set IMS list after a successful commit phase 1.	
4	The online change commit phase is incomplete on this IMS. One or more online change commit phases have been completed on this IMS. All online change commit phases were not sent to this IMS due to errors.	
5	The ACBMBR OLC is complete; however, during COMMIT processing after the OLCSTAT was updated, an error was encountered and one or more members that have been added or changed with the ACBMBR OLC are unusable. Any programs that are not usable are marked bad with a reason code of MOLCCOMMIT. Any databases that are not usable are marked bad with a reason code of MOLCFAIL.	OLC COMPLETE, SOME MBRS UNUSABLE
1D	The OLC directive cannot be processed, because a takeover is in progress.	XRF TAKEOVER IS IN PROGRESS
57	The queue structure is unavailable. IMS is unable to check if there is a global queue count for transactions to be added, changed, or deleted. Commit is not permitted to proceed. The structure might be unavailable for one of the following reasons: <ul style="list-style-type: none"> <li>• CQS is unavailable</li> <li>• The MSGQ structure failed and it has not been rebuilt yet</li> <li>• CQS lost connectivity to the MSGQ structure</li> </ul>	THE QUEUE STRUCTURE IS UNAVAILABLE.

Table 345. Completion codes for the INITIATE OLC command (continued)

Completion code	Meaning	ERROR TEXT (uppercase)
58	<p>An IMS is not registered to RM. An OLCSTAT data set contains an IMS that is not registered to RM. The INITIATE command fails for that IMS.</p> <p>If the INITIATE OLC PHASE(PREPARE) command is issued to an IMS that is registered to RM and the OLCSTAT data set contains an IMS system that is not registered to RM, the command fails. The unregistered IMS command-line response is cc=58. To initiate online change, you must remove the unregistered IMS from the OLCSTAT data set. To remove the IMS, perform the following steps:</p> <ol style="list-style-type: none"> <li>1. Issue QRY MEMBER TYPE(IMS) SHOW(ATTRIB) or SHOW(ALL) to verify the IMS that received the cc=58 was initialized with RMENV=N.</li> <li>2. Unlock the OLCSTAT data set using the DFSUOLC0 utility. When online change is in progress it is necessary for the utility to delete the invalid member name.</li> <li>3. Delete the invalid IMSID from the OLCSTAT data set using the DFSUOLC0 utility.</li> <li>4. Issue TERM OLC to terminate the OLC with the valid IMS systems.</li> <li>5. Reissue the OLC.</li> </ol>	
60	IMODULE GETMAIN storage error.	
61	DFSBCB storage error.	<p><b>INTERNALBLOCK NOT BUILD</b> IMS was unable to get the CSA storage that is required to build the DBRC work area.</p>
62	HIOP storage error.	
63	WKAP storage error.	
64	GETSTOR storage error,	<p><b>INTERNALBLOCK NOT BUILD</b> IMS was unable to get the CSA storage that is required to build an internal block that holds a list of altered HALDB master databases.</p>
70	Module load error.	<p>Module name (8 characters), module type (8 characters).</p> <p>The module type can be 'SECURITY'.</p>
71	Module locate error.	<p>Module name (8 characters), module type (8 characters).</p> <p>The module type can be 'SECURITY'.</p>
72	Randomizer load error.	FP area randomizer name (8 characters).

Table 345. Completion codes for the INITIATE OLC command (continued)

Completion code	Meaning	ERROR TEXT (uppercase)
80	Data set error.	<p>Function (8 characters), ddname (8 characters), return code (8 bytes), and error detail (8 characters).</p> <p>Function can be one of the following:</p> <p><b>BLDL</b> Data set BLDL error.</p> <p><b>BUSY</b> The data set is busy. The detail contains the job name that has the data set.</p> <p><b>DIR</b> Data set directory error.</p> <p><b>DIRSIZE</b> <i>ddname nm mbrname</i>            Where:  <i>ddname</i> The 7-character FMTLIB DD name, which is either FORMATA or FORMATB.  <i>nm</i> The 2-byte FMTLIB data set concatenation number within the FMTLIB DD statement.  <i>mbrname</i> The 8-character invalid member name in the FMTLIB data set.            For example, if a module is erroneously copied into the inactive FMTLIB before the INITIATE OLC PHASE(COMMIT) command is issued, the INITIATE OLC PHASE(COMMIT) command might fail with this error. If the member is an actual MFS format with an invalid directory size, the first two characters of the format name are unprintable hexadecimal values that might appear as blanks or other non-alphanumeric characters in this output. The first two characters of the format name might appear as different non-alphanumeric characters when browsing the FMTLIB with TSO.</p> <p><b>EMPTY</b> Library is empty.</p> <p><b>ENQUEUE</b> Data set enqueue error.</p> <p><b>EOF</b> Data set end-of-file (EOF) error.</p> <p><b>OPEN</b> Data set open error. Error detail can be 'DLS REG'.</p> <p><b>READ</b> Data set read error.</p> <p><b>WRITE</b> Data set write error.</p> <p>DDname can be ACBLIBA, ACBLIBB, FMTLIBA, FMTLIBB, MODBLKSA, MODBLKSB, or MODSTAT.</p> <p>Return code is the data set service return code.</p> <p>Reason code is the data set service reason code.</p>
90	Internal error	<p>Module name that detected internal error (8 characters), unused (8 characters), return code or function code (8 bytes), and error detail (8 characters).</p>

Table 345. Completion codes for the INITIATE OLC command (continued)

Completion code	Meaning	ERROR TEXT (uppercase)
91	The online change prepare phase, commit phase, or abort phase timed out before this IMS responded to the online change prepare or commit phase. The online change prepare phase or commit phase might have succeeded on this IMS. Issue QUERY MEMBER TYPE(IMS) to determine the online change state of this IMS.	
B0	Resource definition error.	<p>Resource type (8 characters), resource name (8 characters), error detail (16 characters).</p> <p>The resource type can be AREA, DB, DMB, PSB, SECURITY, or SMB.</p> <p>The resource name can be a resource name, ACBLIB or MODBLKS.</p> <p>The error detail can be one of the following:</p> <p><b>CISIZE ERROR</b> An attempt to add the area name to the DEDB failed. The CI size of the area exceeds the Fast Path buffer size (BSIZ=) of the IMS online control region.</p> <p><b>CPCTRAN CONFLICT</b> The online change command attempted to add a transaction name that conflicts with a CPIC transaction name. Tranname specifies the transaction that exists as a CPIC transaction.</p> <p><b>DESC CONFLICT</b> The online change command attempted to add a transaction name that conflicts with a descriptor name. Tranname specifies the transaction that exists as a descriptor name.</p> <p><b>DUPLICATE</b> A duplicate area name was found in the named DEDB DBD defined in the IMS system. The area name, that was defined in the DD1 operand on one of the AREA statements in the DBDGEN, is defined in another DEDB DBDGEN or is defined within the DEDB DBDGEN.</p> <p><b>INCOMPATIBLE</b> The DMB in the inactive ACBLIB is not compatible with IMS. The level of the inactive library is not compatible with the current IMS release.</p> <p><b>PDIR NAME</b> The online change attempted to change the program name of the transaction, but the program does not exist, nor is it being added by online change. The program name for a local transaction must exist or be added with the same online change.</p>

Table 345. Completion codes for the INITIATE OLC command (continued)

Completion code	Meaning	ERROR TEXT (uppercase)
B0	Resource definition error (continued).	<p><b>INCONSISTENT</b> IMS detected an inconsistency between the indicated security module and the definition of the new blocks in the inactive MODBLKS data set. Either the security option was requested on the online change command or the prevailing security option is YES or FORCE. In the module name, <i>mmns</i> is the specific table name, where <i>s</i> is the suffix.</p> <p><b>LTERM CONFLICT</b> The online change command attempted to add a transaction name that conflicts with a dynamic lterm name.</p> <p>Tranname specifies the transaction that exists as a dynamically created LTERM. The error detail can be one of the following (continued):</p> <p><b>MSNAME CONFLICT</b> The online change command attempted to add a transaction name that conflicts with an MSNAME. Tranname specifies the transaction that exists as an MSNAME.</p> <p><b>NOT DEFINED</b> Control block was not defined in the active library during the online prepare phase. This is an internal error that should not occur.</p> <p><b>NO BLOCKS</b> Control blocks were missing from the inactive library during the online prepare phase.</p>
B0	Resource definition error (continued).	<p><b>NO FP INSTALLED</b> An attempt was made to add a data entry database (DEDB) to an IMS system that does not have Fast Path installed.</p> <p><b>NO OTHREADS</b> An attempt was made to add a data entry database (DEDB) to an IMS system that was not initialized with DEDBs. No output threads (OTHREADS) were initialized and no I/O will be possible to the added DEDB.</p> <p><b>PARTITION EXISTS</b> A database partition by that name exists.</p> <p><b>RSRC CONFLICT</b> The online change command is attempting to add a transaction name that conflicts with another message destination resource in the Resource Manager (RM). The resource can be an LTERM, CPIC transaction, MSNAME, or descriptor. Tranname specifies the transaction that exists as another resource type.</p> <p><b>UNSUPPORTED MSDB</b> The online change attempted to add an MSDB or change a DEDB to an MSDB, which is not permitted.</p>

Table 345. Completion codes for the INITIATE OLC command (continued)

Completion code	Meaning	ERROR TEXT (uppercase)
B1		Resource type (8 characters), resource name (8 characters), error detail (16 characters).  The resource name is the name of the resource as it is defined to IMS, such as the database name, the program name, or the transaction name.  The resource type can be DATABASE, DMB, FORMAT, PROGRAM, PSB, RTCODE, or TRAN.
B1	Resource state error.  The online change phase failed on this IMS because of the state of the specified resource.  Many of the resource states indicate work is in progress for resources to be changed or deleted by online change. Online change commit detected a resource in a state that results in commit failure. This is the first resource for which an error is detected. IMS is returned to an online change prepare state. Issue a /DISPLAY MODIFY command to display the work in progress for resources to be changed or deleted by online change. Resolve the work in progress, then attempt the INITIATE OLC PHASE(COMMIT) command again.	The error detail can be one of the following:  <b>ACTIVE</b> The routing code <i>rtcode</i> is active.  <b>AREA OPEN</b> An area associated with the FP DEDB is open.  <b>AREA STARTED</b> An area associated with the FP DEDB is started.  <b>CONVERSATION</b> The transaction to be changed or deleted or the transaction referencing a program or database to be changed or deleted is in conversation.  <b>DATABASE <i>dbname</i> ALTER INCOMPLETE</b> Alter processing is not complete for a HALDB database. After alter processing completes, retry the INITIATE OLC command.  <b>DATABASE <i>dbname</i> STG TOO SMALL</b> The OLC function cannot process the altered DDIR because there is not enough internal block storage.  The lack of storage could be caused by the start of another HALDB alter operation after OLC reaches the commit phase.  Retry the INITIATE OLC PHASE(COMMIT) command.  Error details continued in next row.

Table 345. Completion codes for the INITIATE OLC command (continued)

Completion code	Meaning	ERROR TEXT (uppercase)
B1	Resource state error (continued).	<p><b>DBD ACTIVE</b> A /DBDUMP command is active for the database.</p> <p><b>DBR ACTIVE</b> A /DBRECOVERY command is active for the database.</p> <p><b>DBR NEEDED</b> A /DBRECOVERY command is needed for the database resource. Online change was initiated to either change or delete one or more HALDB databases which have not been taken offline with a /DBR command.</p> <p>If the IMS catalog is enabled with an alias other than DFSC (the standard alias prefix), and the catalog databases have been modified with online change, issue the DBR DB command for the databases IMS1D000 and IMS1X000 instead of DFSCD000 and DFSCX000.</p> <p><b>DEP REGS ACTIVE</b> Dependent regions using the FP DEDB are active.</p>
Error details continued in next row.		

Table 345. Completion codes for the INITIATE OLC command (continued)

Completion code	Meaning	ERROR TEXT (uppercase)
B1	Resource state error (continued).	<p data-bbox="938 247 1170 268">Error detail (continued):</p> <p data-bbox="938 289 1149 310"><b>DYNAMIC OR CPIC</b> The transaction being added exists and is not dynamic or CPIC. This is an internal error that should not occur.</p> <p data-bbox="938 411 1110 432"><b>ELIGIBLE SCHD</b> The named transaction is eligible for scheduling and cannot be deleted by online change. Stop the transaction before attempting another online change commit.</p> <p data-bbox="938 558 1138 579"><b>INTERNAL ERROR</b> The resource control block is not found in the active IMS, or the resource control block exists. These are internal errors that should not occur.</p> <p data-bbox="938 705 1016 726"><b>IN USE</b> The resource is in use.  A transaction has queuing in progress (either terminal input program-to-program switch).  A database is in use.</p> <p data-bbox="938 915 1122 936"><b>PSB SCHEDULED</b> A program referencing a database to be changed or deleted is scheduled.</p> <p data-bbox="938 1010 1057 1031"><b>QUEUEING</b> Messages are queued to the transaction or program.</p> <p data-bbox="938 1104 1073 1125"><b>SCHEDULED</b> The named resource (a transaction or program to be changed or deleted, or a program referencing a database to be changed or deleted) is scheduled.</p> <p data-bbox="938 1251 1073 1272"><b>SUSPENDED</b> The transaction to be changed or deleted is on the suspend queue.</p>

Table 345. Completion codes for the INITIATE OLC command (continued)

Completion code	Meaning	ERROR TEXT (uppercase)
B2	IMS state error.	<p data-bbox="907 249 1425 300">IMS state error (32 characters). The IMS state can be one of the following:</p> <p data-bbox="907 317 1016 338"><b>ABENDED</b></p> <p data-bbox="1002 344 1425 632">This IMS ended abnormally since the last successful online change. Online change is not permitted if any IMS ended abnormally since the last online change, unless the FRCABND option is specified on prepare. If the online change phase was rejected, issue TERMINATE OLC to abort the online change, correct the problem, and try the online change again. If an online change specified with the FRCABND option succeeds, this IMS might have to cold start.</p> <p data-bbox="907 651 1214 672"><b>CHECKPOINT IN PROGRESS</b></p> <p data-bbox="1002 678 1354 699">This IMS has checkpoint in progress.</p> <p data-bbox="907 718 1097 739"><b>NOT-REACHABLE</b></p> <p data-bbox="1002 745 1425 898">The online change phase is rejected because this IMS is NOT-REACHABLE. The SCI on the OS image where this IMS is active is down. Restart the SCI and reissue the INITIATE OLC or TERMINATE OLC command.</p> <p data-bbox="907 917 1214 938"><b>OLC ALREADY COMMITTED</b></p> <p data-bbox="1002 945 1425 1050">The online change phase is rejected because online change is already committed. All IMS systems have completed commit phase 1 and the OLCSTAT data set was updated.</p> <p data-bbox="907 1068 1166 1089"><b>OLC NOT IN PROGRESS</b></p> <p data-bbox="1002 1096 1425 1146">The online change phase is rejected because this IMS is not in an online change state.</p>

Table 345. Completion codes for the INITIATE OLC command (continued)

Completion code	Meaning	ERROR TEXT (uppercase)
B2	IMS state error (continued).	<p>IMS state error (32 characters). The IMS state can be one of the following:</p> <p><b>OLC PHASE IN PROGRESS</b> The online change phase is rejected because this IMS has an online change phase already in progress.</p> <p><b>RECOVERY IN PROGRESS</b> The online change command is rejected because the online recovery service is already in progress.</p> <p><b>RESOLVE DB INDOUBTS</b> This IMS has DB indoubts. You must resolve the DB indoubts either by reconnecting the coordinator controller and IMS or by using an operator command.</p> <p><b>RESTART IN PROGRESS</b> This IMS has restart in progress.</p> <p><b>RESTART NOT COMPLETE</b> This IMS initialized before the online change was initiated, but has not completed restart. The online prepare or abort phase is rejected as long as this IMS is in this state. Cancel this IMS, then abort the online change before attempting the online change prepare phase again.</p> <p><b>SHUTDOWN</b> This IMS was shut down normally since the last successful online change. Online change is not permitted if any IMS was shut down normally since the last online change, unless the FRCNRML option is specified on prepare. If the online change phase was rejected, issue TERMINATE OLC to terminate the online change, correct the problem, and try the online change again. If an online change specified with the FRCNRML option succeeds, this IMS might have to cold start.</p>
120	The INIT OLC TYPE(ACBMBR) online change failed.	Staging ACBLIB is empty.
121	The INIT OLC TYPE(ACBMBR) online change failed.	Resource in command has no change.
122	The INIT OLC TYPE(ACBMBR) online change failed.	Not all PSBs are rebuilt for this DMB.
123	The INIT OLC TYPE(ACBMBR) online change failed.	Staging library level is not compatible with the current IMS.
124	The INIT OLC TYPE(ACBMBR) online change failed.	DOPT PSB will not be copied to the active ACBLIB.
125	The INIT OLC TYPE(ACBMBR) online change failed.	No PSB is rebuilt for this change DMB.
126	The INIT OLC TYPE(ACBMBR) online change failed.	The previous member OLC failed in the commit phase 1 processing. The terminate OLC command must be entered to terminate the pending member OLC.
127	The INIT OLC TYPE(ACBMBR) online change failed.	COMMIT MEMBER failure.
128	The INIT OLC TYPE(ACBMBR) online change failed.	REFRESH MBR failure.
129	The INIT OLC TYPE(ACBMBR) online change failed.	Mismatch in ACBSHR= values among IMS systems in the OLCSTAT.
12A	Existing DBDs in the active ACB library cannot be modified.	NAMEONLY DBD CHANGES NOT ALLOWED

Table 345. Completion codes for the INITIATE OLC command (continued)

Completion code	Meaning	ERROR TEXT (uppercase)
12B	The PSB that is specified in the command includes references to the existing DBDs that have been changed in the staging ACB library. The INIT OLC command failed because existing DBDs in the active ACB library cannot be modified.	NAMEONLY PSB CONTAINS CHANGED DB
12C	The DBD that is specified in the command includes references to the existing DBDs that have been changed in the staging ACB library. The INIT OLC command failed because existing DBDs in the active ACB library cannot be modified.	NAMEONLY DBD HAS CHNG LOGICAL DB
12D	The online change prepare phase failed. The catalog PSB DFSCP000 cannot be deleted from the online system while the catalog is enabled. To allow the online change prepare phase to proceed, perform an ACBGEN of the staging ACBLIB with BUILD PSB=DFSCP000.	DELETING CATALOG PSB NOT ALLOWED
12E	The online change prepare phase failed. The catalog DBD DFSCD000 or DFSCX000 cannot be deleted from the online system while the catalog is enabled. In order for the online change prepare phase to proceed, perform and ACBGEN with BUILD PSB=DFSCP000 to the staging ACBLIB.	DELETING CATALOG DBD NOT ALLOWED
130	The INIT OLC TYPE(ACBMBR) online change failed.	Allocation of staging ACBLIB failed.
131	The INIT OLC TYPE(ACBMBR) online change failed.	Allocation of active ACBLIB failed.
132	The INIT OLC TYPE(ACBMBR) online change failed.	Open of staging ACBLIB failed.
133	The INIT OLC TYPE(ACBMBR) online change failed.	IO error of active ACBLIB.
134	The INIT OLC TYPE(ACBMBR) online change failed.	Open of active ACBLIB failed.
135	The INIT OLC TYPE(ACBMBR) online change failed.	IO error of staging ACBLIB.
136	The INIT OLC TYPE(ACBMBR) online change failed because there were more than the maximum supported concatenations for the active ACB library.	
14B	IMS is being shut down. The ACBMBR online change prepare command is terminated.	SHUTDOWN
180	During an INIT OLC PHASE(PREPARE) command processing, the load of the DFSMDA member for the inactive ACBLIB failed. Message DFS0887A is issued on the system console.	DATASET ERROR MDA LOAD
181	During an INIT OLC PHASE(PREPARE) command processing, the allocation of the inactive ACBLIB failed. Message DFS2500W is issued on the system console.	DATASET ERROR ALLOCATE

## Examples

The following are examples of the INITIATE OLC command.

### *Example 1 for INITIATE OLC command*

TSO SPOC input:

```
INITIATE OLC PHASE(PREPARE) TYPE(MODBLKS,ACBLIB)
```

TSO SPOC output:

MbrName	Member	CC	ACBLIB	FMTLIB	MODBLKS	ModId
IMS3	IMS2	0				
IMS3	IMS3	0	B	A	B	1
IMS3	SYS3	0				

### OM API input:

```
CMD (INITIATE OLC PHASE(PREPARE) TYPE(MODBLKS,ACBLIB))
```

### OM API output:

```
<imsout>
<ctl>
<omname>OM10M </omname>
<omvsn>1.1.0</omvsn>
<xm1vsn>1 </xm1vsn>
<statime>2003.163 15:40:53.336327</statime>
<stotime>2003.163 15:41:08.282146</stotime>
<staseq>B7C4AC11FED070C6</staseq>
<stoseq>B7C4AC203FB220C1</stoseq>
<rqsttkn1>USRT011 10084053</rqsttkn1>
<rc>0200000C</rc>
<rsn>00003000</rsn>
</ctl>
<cmderr>
<mbr name="IMS2 ">
<typ>IMS </typ>
<styp>DBDC </styp>
<rc>02000004</rc>
<rsn>00001008</rsn>
</mbr>
<mbr name="SYS3 ">
<typ>IMS </typ>
<styp>DBDC </styp>
<rc>02000004</rc>
<rsn>00001008</rsn>
</mbr>
</cmderr>
<cmd>
<master>IMS3 </master>
<userid>USRT011 </userid>
<verb>INIT</verb>
<kwd>OLC </kwd>
<input>INITIATE OLC PHASE(PREPARE) TYPE(MODBLKS,ACBLIB)</input>
</cmd>
<cmdrsphdr>
<hdr s1b1="MBR" l1b1="MbrName" scope="LCL" sort="n" key="0" scroll="yes" len="8"
dtype="CHAR" align="left" />
<hdr s1b1="MSMBR" l1b1="Member" scope="LCL" sort="a" key="1" scroll="no" len="8"
dtype="CHAR" align="left" />
<hdr s1b1="CC" l1b1="CC" scope="LCL" sort="n" key="0" scroll="yes" len="4"
dtype="INT" align="right" />
<hdr s1b1="ACBL" l1b1="ACBLIB" scope="GBL" sort="n" key="0" scroll="yes" len="1"
dtype="CHAR" align="right" />
<hdr s1b1="FMTL" l1b1="FMTLIB" scope="GBL" sort="n" key="0" scroll="yes" len="1"
dtype="CHAR" align="right" />
<hdr s1b1="MODB" l1b1="MODBLKS" scope="GBL" sort="n" key="0" scroll="yes" len="1"
dtype="CHAR" align="right"/>
<hdr s1b1="MODI" l1b1="ModId" scope="GBL" sort="n" key="0" scroll="yes" len="8"
dtype="CHAR" align="right"/>
</cmdrsphdr>
<cmdrspdata>
<rsp>MBR(IMS3 ) IMSMBR(SYS3 ) CC( 0) </rsp>
<rsp>MBR(IMS3 ) IMSMBR(IMS2 ) CC( 0) </rsp>
<rsp>MBR(IMS3 ) IMSMBR(IMS3 ) CC( 0) ACBL(B) FMTL(A) MODB(B) MODI( 1)
</rsp>
</cmdrspdata>
</imsout>
```

**Explanation:** The IMSplex consists of three IMS systems: SYS3, IMS2, and IMS3. All three of the IMS systems complete the online change prepare phase successfully for the MODBLKS library. IMS3 is the master of the prepare phase. The current online change libraries are ACBLIBB, FMTLIBA, and MODBLKSB. The current modify ID is 1.

*Example 2 for INITIATE OLC command*

TSO SPOC input:

```
INITIATE OLC PHASE(COMMIT)
```

TSO SPOC output:

MbrName	Member	CC	ACBLIB	FMTLIB	MODBLKS	ModId
IMS3	IMS2	0				
IMS3	IMS3	0	B	A	A	2
IMS3	SYS3	0				

OM API input:

```
CMD (INITIATE OLC PHASE(COMMIT))
```

OM API output:

```
<imsout>
<ctl>
<omname>OM10M </omname>
<omvsn>1.1.0</omvsn>
<xmlvsn>1 </xmlvsn>
<statime>2002.163 16:26:14.660563</statime>
<stotime>2002.16316:26:22.293891</stotime>
<staseq>B7C4B635409D30C6</staseq>
<stoseq>B7C4B63C88383347</stoseq>
<rqsttkn1>USRT011 10092614</rqsttkn1>
<rc>0200000C</rc>
<rsn>00003000</rsn>
</ctl>
<cmderr>
<mbr name="IMS2 ">
<typ>IMS </typ>
<styp>DBDC </styp>
<rc>02000004</rc>
<rsn>00001008</rsn>
</mbr>
<mbr name="SYS3 ">
<typ>IMS </typ>
<styp>DBDC </styp>
<rc>02000004</rc>
<rsn>00001008</rsn>
</mbr>
</cmderr>
<cmd>
<master>IMS3 </master>
<userid>USRT011 </userid>
<verb>INIT</verb>
<kwd>OLC </kwd>
<input>INITIATE OLC PHASE(COMMIT) </input>
</cmd>
<cmdrsphdr>
<hdr s1bl="MBR" l1bl="MbrName" scope="LCL" sort="n" key="0" scroll="yes" len="8"
dtype="CHAR" align="left" />
<hdr s1bl="MSMBR" l1bl="Member" scope="LCL" sort="a" key="1" scroll="no" len="8"
dtype="CHAR" align="left" />
<hdr s1bl="CC" l1bl="CC" scope="LCL" sort="n" key="0" scroll="yes" len="4"
dtype="INT" align="right" />
<hdr s1bl="ACBL" l1bl="ACBLIB" scope="GBL" sort="n" key="0" scroll="yes" len="1"
```

```

dtype="CHAR" align="right" />
<hdr s1b1="FMTL" l1b1="FMTLIB" scope="GBL" sort="n" key="0" scroll="yes" len="1"
dtype="CHAR" align="right" />
<hdr s1b1="MODB" l1b1="MODBLKS" scope="GBL" sort="n" key="0" scroll="yes" len="1"
dtype="CHAR" align="right"/>
<hdr s1b1="MODI" l1b1="ModId" scope="GBL" sort="n" key="0" scroll="yes" len="8"
dtype="CHAR" align="right" />
</cmdrsphdr>
<cmdrspdata>
<rsp>MBR(IMS3 ) IMSMBR(SYS3 ) CC( 0) </rsp>
<rsp>MBR(IMS3 ) IMSMBR(IMS2 ) CC( 0) </rsp>
<rsp>MBR(IMS3 ) IMSMBR(IMS3 ) CC( 0) ACBL(B) FMTL(A) MODB(A) MODI( 2)
</rsp>
</cmdrspdata>
</imsout>

```

**Explanation:** A global online change commit is initiated and completes successfully.

### *Example 3 for INITIATE OLC command*

TSO SPOC input:

```
INITIATE OLC PHASE(PREPARE) TYPE(ACBMBR) NAME(OLCDB105 OLCDX111)
```

TSO SPOC output:

MbrName	Member	CC	ACBSHR	DBDName	PSBName	ADD
IMS2	IMS1	0	Y			
IMS2	IMS2	0	Y			
IMS2	IMS1	0		OLCDB105		Y
IMS2	IMS2	0		OLCDB105		Y
IMS2	IMS1	0		OLCDB111		Y
IMS2	IMS2	0		OLCDB111		Y
IMS2	IMS1	0		OLCDI111		Y
IMS2	IMS2	0		OLCDI111		Y
IMS2	IMS1	0		OLCDX111		Y
IMS2	IMS2	0		OLCDX111		Y
IMS2	IMS1	0			OLCPB105	Y
IMS2	IMS2	0			OLCPB105	Y
IMS2	IMS1	0			OLCPB111	Y
IMS2	IMS2	0			OLCPB111	Y

OM API input:

```
CMD (INITIATE OLC PHASE(PREPARE) TYPE(ACBMBR) NAME(OLCDB105 OLCDX111))
```

OM API output:

```

<imsout>
<ctl>
<omname>OM10M </omname>
<omvsn>1.3.0</omvsn>
<xmlvsn>20 </xmlvsn>
<statime>2006.268 17:31:50.028126</statime>
<stotime>2006.268 17:31:50.672287</stotime>
<staseq>BF75AD9623B5EAE3</staseq>
<stoseq>BF75AD96C0F9F723</stoseq>
<rqsttkn1>USRT001 10103150</rqsttkn1>
<rc>0200000C</rc>
<rsn>00003000</rsn>
<rsnmsg>CSLN023I</rsnmsg>
<rsntxt>At least one request was successful.</rsntxt>
</ctl>
<cmderr>
<mbr name="IMS1 ">
<typ>IMS </typ>

```

```

<styp>DBDC </styp>
<rc>02000004</rc>
<rsn>00001008</rsn>
<rsntxt>Command ROUTE list overridden w/ANY</rsntxt>
</mbr>
</cmderr>
<cmd>
<master>IMS2 </master>
<userid>USRT001 </userid>
<verb>INIT</verb>
<kwd>OLC </kwd>
<input>INIT OLC PHASE(PREPARE) TYPE(ACBMBR) NAME(OLCDB105 OLCDX111)
</input>
</cmd>
<cmdrsphdr>
<hdr s1bl="MBR" l1bl="MbrName" scope="LCL" sort="n" key="0"
  scroll="yes" len="8" dtype="CHAR" align="left" />
<hdr s1bl="IMSMBR" l1bl="Member" scope="LCL" sort="a" key="4"
  scroll="no" len="8" dtype="CHAR" align="left" />
<hdr s1bl="CC" l1bl="CC" scope="LCL" sort="n" key="0" scroll="yes"
  len="4" dtype="INT" align="right" />
<hdr s1bl="CCTXT" l1bl="CCText" scope="LCL" sort="n" key="0"
  scroll="yes" len="*" dtype="CHAR" align="left" skipb="yes" />
<hdr s1bl="ACBS" l1bl="ACBSHR" scope="LCL" sort="n" key="0"
  scroll="yes" len="1" dtype="CHAR" align="left" skipb="yes" />
<hdr s1bl="RSC" l1bl="RSCName" scope="LCL" sort="a" key="3" scroll="no"
  len="8" dtype="CHAR" align="left" skipb="yes" />
<hdr s1bl="DBD" l1bl="DBDName" scope="LCL" sort="a" key="2" scroll="no"
  len="8" dtype="CHAR" align="left" skipb="yes" />
<hdr s1bl="PSB" l1bl="PSBName" scope="LCL" sort="a" key="1" scroll="no"
  len="8" dtype="CHAR" align="left" skipb="yes" />
<hdr s1bl="ADD" l1bl="ADD" scope="LCL" sort="n" key="0" scroll="yes"
  len="1" dtype="CHAR" align="left" skipb="yes" />
<hdr s1bl="CHG" l1bl="CHNG" scope="LCL" sort="n" key="0" scroll="yes"
  len="1" dtype="CHAR" align="left" skipb="yes" />
<hdr s1bl="CPY" l1bl="COPY" scope="LCL" sort="n" key="0" scroll="yes"
  len="1" dtype="CHAR" align="left" skipb="yes" />
<hdr s1bl="RFS" l1bl="RFSH" scope="LCL" sort="n" key="0" scroll="yes"
  len="1" dtype="CHAR" align="left" skipb="yes" />
</cmdrsphdr>
<cmdrspdata>
<rsp>MBR(IMS2 ) IMSMBR(IMS1 ) CC( 0) ACBS(Y) </rsp>
<rsp>MBR(IMS2 ) IMSMBR(IMS1 ) CC( 0) DBD(OLCDB105) ADD(Y) </rsp>
<rsp>MBR(IMS2 ) IMSMBR(IMS1 ) CC( 0) DBD(OLCDX111) ADD(Y) </rsp>
<rsp>MBR(IMS2 ) IMSMBR(IMS1 ) CC( 0) DBD(OLCDB111) ADD(Y) </rsp>
<rsp>MBR(IMS2 ) IMSMBR(IMS1 ) CC( 0) DBD(OLCDI111) ADD(Y) </rsp>
<rsp>MBR(IMS2 ) IMSMBR(IMS1 ) CC( 0) PSB(OLCPB105) ADD(Y) </rsp>
<rsp>MBR(IMS2 ) IMSMBR(IMS1 ) CC( 0) PSB(OLCPB111) ADD(Y) </rsp>
<rsp>MBR(IMS2 ) IMSMBR(IMS2 ) CC( 0) ACBS(Y) </rsp>
<rsp>MBR(IMS2 ) IMSMBR(IMS2 ) CC( 0) DBD(OLCDB105) ADD(Y) </rsp>
<rsp>MBR(IMS2 ) IMSMBR(IMS2 ) CC( 0) DBD(OLCDX111) ADD(Y) </rsp>
<rsp>MBR(IMS2 ) IMSMBR(IMS2 ) CC( 0) DBD(OLCDB111) ADD(Y) </rsp>
<rsp>MBR(IMS2 ) IMSMBR(IMS2 ) CC( 0) DBD(OLCDI111) ADD(Y) </rsp>
<rsp>MBR(IMS2 ) IMSMBR(IMS2 ) CC( 0) PSB(OLCPB105) ADD(Y) </rsp>
<rsp>MBR(IMS2 ) IMSMBR(IMS2 ) CC( 0) PSB(OLCPB111) ADD(Y) </rsp>
</cmdrspdata>

```

**Explanation:** The INIT OLC PHASE(PREPARE) TYPE(ACBMBR) NAME(OLCDB105 OLCDX111) command is issued. IMS2 is the command master. IMS1 and IMS2 are in the OLCSTAT data set. The ACBSHR=Y is defined at all IMS systems so only the command master writes to the ACBLIB. The online change prepare phase is successful at all IMS systems. The command master returns to the OM the members that are to be added, changed, or copied from the staging ACBLIB to the active ACBLIB.

You can issue the INIT OLC PHASE(COMMIT) command to complete the online change.

*Example 4 for INITIATE OLC command*

TSO SPOC input:

```
INITIATE OLC PHASE(PREPARE) TYPE(ACBMBR) NAME(OLCDB105)
```

TSO SPOC output:

MbrName	Member	CC	CCText	ACBSHR	RSCName	DBDName	PSBName	ADD
IMS2	IMS1	0		N				
IMS2	IMS2	0		N				
IMS2	IMS2	10	NO RESOURCES FOUND		OLCDB105			
IMS2	IMS1	0				OLCDB105		Y
IMS2	IMS1	0					OLCPB105	Y

OM API input:

```
CMD (INITIATE OLC PHASE(PREPARE) TYPE(ACBMBR) NAME(OLCDB105))
```

OM API output:

```
<imsout>
<ctl>
<omname>OM10M </omname>
<omvsn>1.3.0</omvsn>
<xmlvsn>20 </xmlvsn>
<statime>2006.268 18:08:10.305797</statime>
<stotime>2006.268 18:08:10.993076</stotime>
<staseq>BF75B5B56A1054E3</staseq>
<stoseq>BF75B5B611DB442C</stoseq>
<rqsttkn1>USRT001 10110810</rqsttkn1>
<rc>0200000C</rc>
<rsn>0000300C</rsn>
<rsnmsg>CSLN055I</rsnmsg>
<rsntxt>The command completed with warning(s).</rsntxt>
</ctl>
<cmderr>
<mbr name="IMS2 ">
<typ>IMS </typ>
<styp>DBDC </styp>
<rc>00000004</rc>
<rsn>0000100C</rsn>
<rsntxt>Some down or not applicable</rsntxt>
</mbr>
<mbr name="IMS1 ">
<typ>IMS </typ>
<styp>DBDC </styp>
<rc>02000004</rc>
<rsn>00001008</rsn>
<rsntxt>Command ROUTE list overridden w/ANY</rsntxt>
</mbr>
</cmderr>
<cmd>
<master>IMS2 </master>
<userid>USRT001 </userid>
<verb>INIT</verb>
<kwd>OLC </kwd>
<input>INIT OLC PHASE(PREPARE) TYPE(ACBMBR) NAME(OLCDB105) </input>
</cmd>
<cmdrsphdr>
<hdr s1bl="MBR" l1bl="MbrName" scope="LCL" sort="n" key="0"
scroll="yes" len="8" dtype="CHAR" align="left" />
<hdr s1bl="IMSMBR" l1bl="Member" scope="LCL" sort="a" key="4"
scroll="no" len="8" dtype="CHAR" align="left" />
```

```

<hdr s1bl="CC" l1bl="CC" scope="LCL" sort="n" key="0" scroll="yes"
  len="4" dtype="INT" align="right" />
<hdr s1bl="CCTXT" l1bl="CCText" scope="LCL" sort="n" key="0"
  scroll="yes" len="*" dtype="CHAR" align="left" skipb="yes" />
<hdr s1bl="ACBS" l1bl="ACBSHR" scope="LCL" sort="n" key="0"
  scroll="yes" len="1" dtype="CHAR" align="left" skipb="yes" />
<hdr s1bl="RSC" l1bl="RSCName" scope="LCL" sort="a" key="3" scroll="no"
  len="8" dtype="CHAR" align="left" skipb="yes" />
<hdr s1bl="DBD" l1bl="DBDName" scope="LCL" sort="a" key="2" scroll="no"
  len="8" dtype="CHAR" align="left" skipb="yes" />
<hdr s1bl="PSB" l1bl="PSBName" scope="LCL" sort="a" key="1" scroll="no"
  len="8" dtype="CHAR" align="left" skipb="yes" />
<hdr s1bl="ADD" l1bl="ADD" scope="LCL" sort="n" key="0" scroll="yes"
  len="1" dtype="CHAR" align="left" skipb="yes" />
<hdr s1bl="CHG" l1bl="CHNG" scope="LCL" sort="n" key="0" scroll="yes"
  len="1" dtype="CHAR" align="left" skipb="yes" />
<hdr s1bl="CPY" l1bl="COPY" scope="LCL" sort="n" key="0" scroll="yes"
  len="1" dtype="CHAR" align="left" skipb="yes" />
<hdr s1bl="RFS" l1bl="RFSH" scope="LCL" sort="n" key="0" scroll="yes"
  len="1" dtype="CHAR" align="left" skipb="yes" />
</cmdrsphdr>
<cmdrspdata>
<rsp>MBR(IMS2 ) IMSMBR(IMS1 ) CC( 0) ACBS(N) </rsp>
<rsp>MBR(IMS2 ) IMSMBR(IMS1 ) CC( 0) DBD(OLCDB105) ADD(Y) </rsp>
<rsp>MBR(IMS2 ) IMSMBR(IMS1 ) CC( 0) PSB(OLCPB105) ADD(Y) </rsp>
<rsp>MBR(IMS2 ) IMSMBR(IMS2 ) CC( 0) ACBS(N) </rsp>
<rsp>MBR(IMS2 ) IMSMBR(IMS2 ) CC( 10) CCTXT(NO RESOURCES FOUND)
  RSC(OLCDB105) </rsp>
</cmdrspdata>
</imsout>

```

**Explanation:** The INIT OLC PHASE(PREPARE) TYPE(ACBMBR) NAME(OLCDB105) command is issued. IMS2 is the command master. IMS1 and IMS2 are in the OLCSTAT data set. The ACBSHR=N is defined at all IMS systems. All IMS systems will write to their own ACBLIB.

The online change prepare phase is successful at IMS1 and IMS3. The online change is not applicable at IMS2 because member DBD1 is not found in IMS2 staging ACBLIB and active ACBLIB.

All sharing IMS systems return the member information to the command master. The command master returns to the OM the members that are to be added, changed, or copied from the staging ACBLIB to the active ACBLIB.

You can issue the INIT OLC PHASE(COMMIT) command to complete the online change.

*Example 5 for INITIATE OLC command*

TSO SPOC input:

```
INITIATE OLC PHASE(PREPARE) TYPE(ACBMBR) NAME(OLCDB105 OLCDX111)
```

TSO SPOC output:

MbrName	Member	CC	CCText	ACBSHR	DBDName	PSBName	ADD
IMS2	IMS1	1	CMD NOT APPLICABLE TO THIS IMS	N			
IMS2	IMS2	0		N			
IMS2	IMS2	0			OLCDB105		Y
IMS2	IMS2	0			OLCDB111		Y
IMS2	IMS2	0			OLCDI111		Y
IMS2	IMS2	0			OLCDX111		Y
IMS2	IMS2	0				OLCPB105	Y
IMS2	IMS2	0				OLCPB111	Y

## OM API input:

```
CMD (INITIATE OLC PHASE(PREPARE) TYPE(ACBMBR) NAME(OLCDB105 OLCDX111))
```

## OM API output:

```
<imsout>
<ctl>
<omname>OM10M </omname>
<omvsn>1.3.0</omvsn>
<xmlvsn>20 </xmlvsn>
<statime>2006.275 22:29:51.356740</statime>
<stotime>2006.275 22:34:25.572982</stotime>
<staseq>BF7EBD4154744026</staseq>
<stoseq>BF7EBE46D7C761E4</stoseq>
<rqsttkn1>USRT001 10152951</rqsttkn1>
<rc>0200000C</rc>
<rsn>0000300C</rsn>
<rsnmsg>CSLN055I</rsnmsg>
<rsntxt>The command completed with warning(s).</rsntxt>
</ctl>
<cmderr>
<mbr name="IMS2 ">
<typ>IMS </typ>
<styp>DBDC </styp>
<rc>00000004</rc>
<rsn>0000100C</rsn>
<rsntxt>Some down or not applicable</rsntxt>
</mbr>
</cmderr>
<cmd>
<master>IMS2 </master>
<userid>USRT001 </userid>
<verb>INIT</verb>
<kwd>OLC </kwd>
<input>INITIATE OLC PHASE(PREPARE) TYPE(ACBMBR) NAME(OLCDB105 OLCDX111)
</input>
</cmd>
<cmdrsphdr>
<hdr s1b1="MBR" l1b1="MbrName" scope="LCL" sort="n" key="0"
scroll="yes" len="8" dtype="CHAR" align="left" />
<hdr s1b1="MSMBR" l1b1="Member" scope="LCL" sort="a" key="4"
scroll="no" len="8" dtype="CHAR" align="left" />
<hdr s1b1="CC" l1b1="CC" scope="LCL" sort="n" key="0" scroll="yes"
len="4" dtype="INT" align="right" />
<hdr s1b1="CCTXT" l1b1="CCText" scope="LCL" sort="n" key="0"
scroll="yes" len="*" dtype="CHAR" align="left" skipb="yes" />
<hdr s1b1="ACBS" l1b1="ACBSHR" scope="LCL" sort="n" key="0"
scroll="yes" len="1" dtype="CHAR" align="left" skipb="yes" />
<hdr s1b1="RSC" l1b1="RSCName" scope="LCL" sort="a" key="3" scroll="no"
len="8" dtype="CHAR" align="left" skipb="yes" />

<hdr s1b1="DBD" l1b1="DBDName" scope="LCL" sort="a" key="2" scroll="no"
len="8" dtype="CHAR" align="left" skipb="yes" />
<hdr s1b1="PSB" l1b1="PSBName" scope="LCL" sort="a" key="1" scroll="no"
len="8" dtype="CHAR" align="left" skipb="yes" />
<hdr s1b1="ADD" l1b1="ADD" scope="LCL" sort="n" key="0" scroll="yes"
len="1" dtype="CHAR" align="left" skipb="yes" />
<hdr s1b1="CHG" l1b1="CHNG" scope="LCL" sort="n" key="0" scroll="yes"
len="1" dtype="CHAR" align="left" skipb="yes" />
<hdr s1b1="CPY" l1b1="COPY" scope="LCL" sort="n" key="0" scroll="yes"
len="1" dtype="CHAR" align="left" skipb="yes" />
<hdr s1b1="RFS" l1b1="RFSH" scope="LCL" sort="n" key="0" scroll="yes"
len="1" dtype="CHAR" align="left" skipb="yes" />
</cmdrsphdr>
<cmdrspdata>
<rsp>MBR(IMS2 ) MSMBR(IMS1 ) CC( 1) CCTXT(CMD NOT APPLICABLE
TO THIS IMS) ACBS(N) </rsp>
```

```

<rsp>MBR(IMS2 ) IMSMBR(IMS2 ) CC( 0) ACBS(N) </rsp>
<rsp>MBR(IMS2 ) IMSMBR(IMS2 ) CC( 0) DBD(OLCDB105) ADD(Y) </rsp>
<rsp>MBR(IMS2 ) IMSMBR(IMS2 ) CC( 0) DBD(OLCDX111) ADD(Y) </rsp>
<rsp>MBR(IMS2 ) IMSMBR(IMS2 ) CC( 0) DBD(OLCDB111) ADD(Y) </rsp>
<rsp>MBR(IMS2 ) IMSMBR(IMS2 ) CC( 0) DBD(OLCDI111) ADD(Y) </rsp>
<rsp>MBR(IMS2 ) IMSMBR(IMS2 ) CC( 0) PSB(OLCPB105) ADD(Y) </rsp>
<rsp>MBR(IMS2 ) IMSMBR(IMS2 ) CC( 0) PSB(OLCPB111) ADD(Y) </rsp>
</cmdrspdata>
</imsout>

```

**Explanation:** The INIT OLC PHASE(PREPARE) TYPE(ACBMBR) NAME(OLCDB105 OLCDX111) command is issued. IMS2 is the command master. IMS1 and IMS2 are in the OLCSTAT data set. The ACBSHR=N is defined at all IMS systems. All IMS systems will write to their own ACBLIB.

The online change prepare phase is successful at IMS1 and IMS3. The online change prepare phase fails at IMS2 because PSBY is not found in IMS2 staging ACBLIB and PSBY is found in IMS2 active ACBLIB. This implies a member deletion for PSBY. The command fails with a return and reason code of LIST | SOME because member online change does not support deleting members.

All sharing IMS systems return the member information to the command master. The command master returns to the OM the members that are to be added, changed, or copied from the staging ACBLIB to the active ACBLIB.

You must issue the TERM OLC command to terminate the online change. You must add PSBY to the staging ACBLIB for member online change or perform a full library switch online change to have PSBY deleted.

**Explanation:** The INIT OLC PHASE(PREPARE) TYPE(ACBMBR) NAME(OLCDB105) command is issued. IMS2 is the command master. IMS1 and IMS2 are in the OLCSTAT data set. The ACBSHR=N is defined at all IMS systems. All IMS systems will write to their own ACBLIB.

The online change prepare phase is successful at IMS1 and IMS3. The online change is not applicable at IMS2 because member DBD1 is not found in IMS2 staging ACBLIB and active ACBLIB.

All sharing IMS systems return the member information to the command master. The command master returns to the OM the members that are to be added, changed, or copied from the staging ACBLIB to the active ACBLIB.

You can issue the INIT OLC PHASE(COMMIT) command to complete the online change.

*Example 6 for INITIATE OLC command*

TSO SPOC input:

```
INIT OLC PHASE(PREPARE) TYPE(ACBMBR) NAME(ECHOPGM,ECHODBD) OPTION(NAMEONLY)
```

TSO SPOC output:

MbrName	Member	CC	ACBSHR	DBDName	PSBName	ADD
IMS1	IMS1	0	N			
IMS1	IMS1	0		ECHODBD		Y
IMS1	IMS1	0			ECHOPGM	Y

OM API input:

INIT OLC PHASE(PREPARE) TYPE(ACBMBR) NAME(ECHOPGM,ECHODBD) OPTION(NAMEONLY)

OM API output:

```
<cmdrsphdr>
<hdr s1b1="MBR" l1b1="MbrName" scope="LCL" sort="n" key="0"
  scroll="yes" len="8" dtype="CHAR" align="left" />
<hdr s1b1="MSMBR" l1b1="Member" scope="LCL" sort="a" key="4"
  scroll="no" len="8" dtype="CHAR" align="left" />
<hdr s1b1="CC" l1b1="CC" scope="LCL" sort="n" key="0" scroll="yes"
  len="4" dtype="INT" align="right" />
<hdr s1b1="CCTXT" l1b1="CCText" scope="LCL" sort="n" key="0"
  scroll="yes" len="*" dtype="CHAR" align="left" skipb="yes" />
<hdr s1b1="ACBS" l1b1="ACBSHR" scope="LCL" sort="n" key="0"
  scroll="yes" len="1" dtype="CHAR" align="left" skipb="yes" />
<hdr s1b1="RSC" l1b1="RSCName" scope="LCL" sort="a" key="3" scroll="no"
  len="8" dtype="CHAR" align="left" skipb="yes" />
<hdr s1b1="DBD" l1b1="DBDName" scope="LCL" sort="a" key="2" scroll="no"
  len="8" dtype="CHAR" align="left" skipb="yes" />
<hdr s1b1="PSB" l1b1="PSBName" scope="LCL" sort="a" key="1" scroll="no"
  len="8" dtype="CHAR" align="left" skipb="yes" />
<hdr s1b1="ADD" l1b1="ADD" scope="LCL" sort="n" key="0" scroll="yes"
  len="1" dtype="CHAR" align="left" skipb="yes" />
<hdr s1b1="CHG" l1b1="CHNG" scope="LCL" sort="n" key="0" scroll="yes"
  len="1" dtype="CHAR" align="left" skipb="yes" />
<hdr s1b1="CPY" l1b1="COPY" scope="LCL" sort="n" key="0" scroll="yes"
  len="1" dtype="CHAR" align="left" skipb="yes" />
<hdr s1b1="RFS" l1b1="RFSH" scope="LCL" sort="n" key="0" scroll="yes"
  len="1" dtype="CHAR" align="left" skipb="yes" />
</cmdrsphdr>
<cmdrspdata>
<rsp>MBR(IMS1 ) IMSMBR(IMS1 ) CC( 0) ACBS(N) </rsp>
<rsp>MBR(IMS1 ) IMSMBR(IMS1 ) CC( 0) PSB(ECHOPGM ) CPY(Y) </rsp>
<rsp>MBR(IMS1 ) IMSMBR(IMS1 ) CC( 0) DBD(ECHODBD ) CPY(Y) </rsp>
</cmdrspdata>
```

**Explanation:** PSB ECHOPGM and DBD ECHOPGM are specified to be added to the active ACB library. These are new to the active ACB library.

*Example 7 for INITIATE OLC command*

TSO SPOC input:

INIT OLC PHASE(PREPARE) TYPE(ACBMBR) NAME(ECHODBD) OPTION(NAMEONLY)

TSO SPOC output:

MbrName	Member	CC	CCText	ACBSHR	DBDName
IMS1	IMS1	0		N	
IMS1	IMS1	12A	NAMEONLY DBD CHANGES NOT ALLOWED		ECHODBD

OM API input:

INIT OLC PHASE(PREPARE) TYPE(ACBMBR) NAME(ECHODBD) OPTION(NAMEONLY)

OM API output:

```
<cmdrsphdr>
<hdr s1b1="MBR" l1b1="MbrName" scope="LCL" sort="n" key="0"
  scroll="yes" len="8" dtype="CHAR" align="left" />
<hdr s1b1="MSMBR" l1b1="Member" scope="LCL" sort="a" key="4"
  scroll="no" len="8" dtype="CHAR" align="left" />
<hdr s1b1="CC" l1b1="CC" scope="LCL" sort="n" key="0" scroll="yes"
  len="4" dtype="INT" align="right" />
<hdr s1b1="CCTXT" l1b1="CCText" scope="LCL" sort="n" key="0"
  scroll="yes" len="*" dtype="CHAR" align="left" skipb="yes" />
<hdr s1b1="ACBS" l1b1="ACBSHR" scope="LCL" sort="n" key="0" />
```

```

scroll="yes" len="1" dtype="CHAR" align="left" skipb="yes" />
<hdr s1b1="RSC" l1b1="RSCName" scope="LCL" sort="a" key="3" scroll="no"
len="8" dtype="CHAR" align="left" skipb="yes" />
<hdr s1b1="DBD" l1b1="DBDName" scope="LCL" sort="a" key="2" scroll="no"
len="8" dtype="CHAR" align="left" skipb="yes" />
<hdr s1b1="PSB" l1b1="PSBName" scope="LCL" sort="a" key="1" scroll="no"
len="8" dtype="CHAR" align="left" skipb="yes" />
<hdr s1b1="ADD" l1b1="ADD" scope="LCL" sort="n" key="0" scroll="yes"
len="1" dtype="CHAR" align="left" skipb="yes" />
<hdr s1b1="CHG" l1b1="CHNG" scope="LCL" sort="n" key="0" scroll="yes"
len="1" dtype="CHAR" align="left" skipb="yes" />
<hdr s1b1="CPY" l1b1="COPY" scope="LCL" sort="n" key="0" scroll="yes"
len="1" dtype="CHAR" align="left" skipb="yes" />
<hdr s1b1="RFS" l1b1="RFSH" scope="LCL" sort="n" key="0" scroll="yes"
len="1" dtype="CHAR" align="left" skipb="yes" />
</cmdrsphdr>
<cmdrspdata>
<rsp>MBR(IMS1 ) IMSMBR(IMS1 ) CC( 0) ACBS(N) </rsp>
<rsp>MBR(IMS1 ) IMSMBR(IMS1 ) CC( 12A) CCTXT(NAMEONLY DBD CHANGES
NOT ALLOWED) DBD(ECHODBD ) (Y) </rsp>
</cmdrspdata>

```

**Explanation:** ECHODBD already exists in the active ACB library. ECHODBD is not in the MODBLKS library.

#### *Example 8 for INITIATE OLC command*

In the following example, the INIT OLC PHASE(PREPARE) TYPE(MODBLKS,ACBLIB) command is issued in an IMSplex that has two IMS systems with DRD and IMS management of ACBs enabled. The PREPARE phase fails as the command is not applicable in the environment. The COMMIT fails as there is no OLC to commit. The OLC must be terminated.

```

INIT OLC PHASE(PREPARE) TYPE(MODBLKS,ACBLIB)
  MbrName  Member      CC CCText
  IMS1     IMS2         1  CMD NOT APPLICABLE TO THIS IMS
  IMS2     IMS2         1  CMD NOT APPLICABLE TO THIS IMS

INIT OLC PHASE(COMMIT)
  MbrName  Member      CC CCText
  IMS1     IMS2         1  CMD NOT APPLICABLE TO THIS IMS
  IMS2     IMS2         1  CMD NOT APPLICABLE TO THIS IMS

TERM OLC
  MbrName  Member      CC
  IMS1     IMS2         0
  IMS2     IMS2         0

```

#### *Example 9 for INITIATE OLC command*

In the following example, the TYPE(ALL) is specified on the command in an environment where ACBLIB is not used because the IMS management of ACBs is enabled, but MODBLKS OLC and FMTLIB OLC are enabled. The OLC for the ACBLIB is ignored.

```

INIT OLC PHASE(PREPARE) TYPE(ALL)
  MbrName  Member      CC ACBLIB FMTLIB MODBLKS  ModId
  SYS3     SYS3         0      A      A      A      5

INIT OLC PHASE(COMMIT)
  MbrName  Member      CC ACBLIB FMTLIB MODBLKS  ModId
  SYS3     SYS3         0      B      B      B      6

```

**Related concepts:**

➡ How to interpret CSL request return and reason codes (System Programming APIs)

**Related reference:**

Chapter 2, “Command keywords and their synonyms,” on page 63

Chapter 40, “/MODIFY command,” on page 959

➡ Global Online Change utility (DFSUOLC0) (System Utilities)

## INITIATE OLC error handling

The INITIATE OLC command can result in errors that leave one or more of the IMS systems in the IMSplex in various online change states. Issue the QUERY MEMBER command and the QUERY OLC command to help you determine whether to terminate the online change or try the INITIATE OLC command again.

Before attempting online change, issue the QUERY OLC LIBRARY(OLCSTAT) SHOW(MODID) command to get the current modify id. If the INITIATE OLC command fails, issue the QUERY OLC LIBRARY(OLCSTAT) SHOW(MODID) command again, to see if the modify id is the same. If the modify id increased by 1, the online change is considered to be successfully completed.

If the INITIATE OLC command fails, issue the QUERY MEMBER TYPE(IMS) SHOW(STATUS) command to display the online change state of all the IMS systems in the IMSplex. Evaluate the QUERY MEMBER TYPE(IMS) SHOW(STATUS) output to help you to determine what to do:

- None of the IMS systems in an online change state

If the INITIATE OLC PHASE(PREPARE) command failed on all of the IMS systems in the IMSplex, none of them are in an online change state. This QUERY OLC LIBRARY(OLCSTAT) SHOW(MODID) command output shows the same modify id as the QUERY OLC LIBRARY(OLCSTAT) SHOW(MODID) command issued before the INITIATE OLC PHASE(PREPARE) command. If you want to attempt online change again, issue another INITIATE OLC PHASE(PREPARE) command.

If an INITIATE OLC PHASE(COMMIT) command successfully completes commit phase 3 on all the IMS systems but fails because of an error such as a command timeout, then no IMS will remain in an online change state. The QUERY MEMBER TYPE(IMS) SHOW(STATUS) command output shows no IMS in an online change state, so no further action is needed. The online change successfully completed.

If an INITIATE OLC PHASE(PREPARE) command failed for all the IMS systems in the IMSplex, but Resource Manager (RM) still has information about the process, then the online change needs to be terminated. This is a very rare case where the command master initiates the online change process with RM, an error occurs, or the online change prepare phase fails on all the IMS systems, and the command master is unable to terminate the online change process with RM. RM issues a CSLR2200 message but no CSLR2201 message. The QUERY MEMBER TYPE(IMS) SHOW(STATUS) command output shows no IMS in an online change state. If you try to initiate an online change again with another INITIATE OLC PHASE(PREPARE) command, it is rejected. You must terminate the online change with a TERMINATE OLC command.

- Some of the IMS systems in a prepare complete state

If the INITIATE OLC PHASE(PREPARE) command fails and some of the IMS systems show a status of online change prepare complete (OLCPREPC), then the prepare phase succeeded on some of the IMS systems. You must terminate the

online change with the TERMINATE OLC command. The IMS systems that have a status of OLCPREPC remain in an online change prepare complete state until you abort the online change.

- All IMS systems in a prepare complete state

If the INITIATE OLC PHASE(PREPARE) or INITIATE OLC PHASE(COMMIT) command fails and all of the IMS systems are in a prepare complete state (OLCPREPC) and the master shows a global status of online change prepare complete (OLCPREPC), then the prepare succeeded. You can issue INITIATE OLC PHASE(COMMIT) if you want to proceed with the online change. You can issue TERMINATE OLC if you want to abort the online change.

If the specified INITIATE OLC PHASE(PREPARE) command did not apply to any IMS in the IMSplex, the command is rejected, but all of the IMS systems show a status of OLCPREPC. For example, if the IMSplex consists of only DBCTL subsystems, then the INITIATE OLC PHASE(PREPARE) TYPE(FMTLIB) command does not apply to any of the IMS systems and is rejected. However, all of the IMS systems will show a status of OLCPREPC, indicating they are participating in the online change. If this situation occurs, you must terminate the online change. INITIATE OLC PHASE(COMMIT) will be rejected, since the online change applies to no IMS in the IMSplex.

- Some IMS systems in prepare complete and commit phase 1 complete state

If an INITIATE OLC PHASE(COMMIT) command fails during commit phase 1, some of the IMS systems may be in a prepare complete state (OLCPREPC) and some of the IMS systems may be in a commit phase 1 complete state (OLCCMT1C). The commit phase failed before the master updated the OLCSTAT data set, so the online change is not committed. You may correct the problem and try the commit again with the INITIATE OLC PHASE(COMMIT) command. Or, you may decide to abort the online change with the TERMINATE OLC command.

The IMS systems that are in an online change state remain in an online change state until you abort the online change or commit the online change.

- All IMS systems in a commit phase 1 complete state

If an INITIATE OLC PHASE(COMMIT) command fails after commit phase 1 is completed, all of the IMS systems are in a commit phase 1 complete state (OLCCMT1C).

If the error occurs before the master updates the OLCSTAT data set, then the online change is not yet successful. You may abort the online change with the TERMINATE OLC command or attempt the commit again with another INITIATE OLC PHASE(COMMIT) command.

If the error occurs after the master updates the OLCSTAT data set, then the online change is successful. You may not abort the online change. You must attempt the commit again with another INITIATE OLC PHASE(COMMIT) command to finish up the online change. The INITIATE OLC PHASE(COMMIT) command finishes the online change by completing commit phase 2.

The IMS systems that are in an online change state remain in an online change state until you abort the online change or commit the online change.

You can determine if the OLCSTAT data set has been updated by checking the modify id. Issue the QUERY OLC LIBRARY(OLCSTAT) SHOW(MODID) command. Check if the modify id returned is different from the modify id returned by the INITIATE OLC PHASE(PREPARE) command, or the modify id returned by a QUERY OLC LIBRARY(OLCSTAT) SHOW(MODID) command issued before the INITIATE OLC PHASE(COMMIT) command.

- Some IMS systems in commit phase 1 complete state and some in commit phase 2 complete state

If an INITIATE OLC PHASE(COMMIT) command fails during commit phase 2, some of the IMS systems may be in a commit phase 1 complete state (OLCCMT1C) and some may have completed commit phase 2 and be in a commit phase2 complete state (OLCCMT2C). The INITIATE OLC PHASE(COMMIT) command failed after commit phase 1 completed on all of the IMS systems and the master updated the OLCSTAT data set. Once the OLCSTAT data set is updated, the online change is considered to be successful and cannot be aborted. You must finish the online change commit phase 2 by issuing the INITIATE OLC PHASE(COMMIT) command again. The INITIATE OLC PHASE(COMMIT) command finishes the online change by completing commit phase 2.

The IMS systems that are in an online change state remain in an online change state until you finish the online change with an INITIATE OLC PHASE(COMMIT) command.

- All IMS systems in commit phase 2 complete state

If an INITIATE OLC PHASE(COMMIT) command fails during commit phase 2 or commit phase 3, all of the IMS systems may end up in a commit phase 2 complete state (OLCCMT2C). The INITIATE OLC PHASE(COMMIT) command failed after commit phase 1 completed on all of the IMS systems and the master updated the OLCSTAT data set. Once the OLCSTAT data set is updated, the online change is considered to be successful and can not be aborted. you must finish the online change commit phase 2 and commit phase 3 by issuing the INITIATE OLC PHASE(COMMIT) command again.

The IMS systems that are in an online change state remain in an online change state until you finish the online change with an INITIATE OLC PHASE(COMMIT) command.

- Some IMS systems in commit phase 2 complete state and some not in online change state

If an INITIATE OLC PHASE(COMMIT) command fails during commit phase 3, some of the IMS systems may be in a commit phase 2 complete state (OLCCMT2C) and some may have completed commit phase 3 and no longer be in an online change state. The INITIATE OLC PHASE(COMMIT) command failed after commit phase 1 completed on all of the IMS systems and master updated the OLCSTAT data set. Once the OLCSTAT data set is updated, the online change is considered to be successful and can not be aborted. You must finish the online change commit phase 3 by issuing the INITIATE OLC PHASE(COMMIT) command again, routing it to an IMS that is still in a commit phase 2 complete state. The INITIATE OLC PHASE(COMMIT) command finishes the online change by completing commit phase 3.

The IMS systems that are in an online change state remain in an online change state until you finish the online change with an INITIATE OLC PHASE(COMMIT) command.

- One IMS in commit phase 3 failed state

If an INITIATE OLC PHASE(COMMIT) command fails during commit phase 3 and the command master associated with this command determines the command was not successful at other IMS systems, the commit phase 3 fails and the command master exits with an error. At this point, the command master is left in an online change state. Issue another INITIATE OLC PHASE(COMMIT) command routed to the master (perhaps with a longer command timeout value) to complete the commit phase 3 cleanup of online change information.

Errors unique to the processing of this command are returned as a completion code. A completion code is returned for an IMS participating in the online change phase.

## INITIATE OLREORG command

Use the INITIATE OLREORG command to reorganize online HALDB PHDAM or PHIDAM partitions or to alter the definition of an online HALDB database.

This command can be issued in either a type-1 format (/INITIATE OLREORG) or a type-2 format (INITIATE OLREORG).

The ALTER option is supported only by the type-2 format of the command.

Subsections:

- "Environment"
- "Syntax"
- "Keywords"
- "Usage notes" on page 939
- "Command responses for /INITIATE OLREORG" on page 940
- "Output fields" on page 940
- "Return, reason, and completion codes" on page 941
- "Examples" on page 944

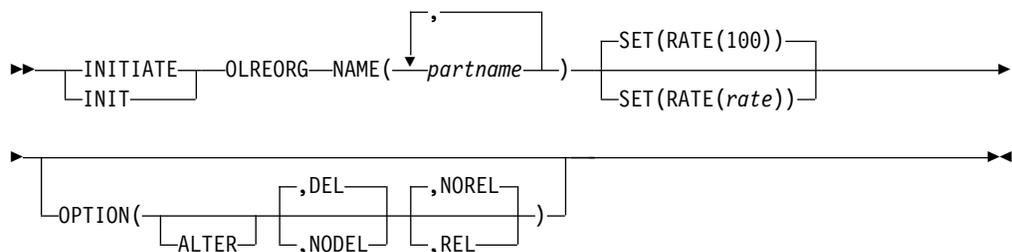
### Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) from which the INITIATE command and keywords can be issued.

Table 346. Valid environments for the INITIATE OLREORG command and keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
INITIATE OLREORG	X	X	
NAME	X	X	
SET	X	X	
OPTION	X	X	

### Syntax



### Keywords

The following keywords are valid for the /INITIATE OLREORG command:

**NAME ()**

The name of a HALDB partition or, when OPTION(ALTER) is specified, the name of a HALDB master database.

You can specify only PHDAM or PHIDAM partition names. You cannot use the wildcard character (\*). For example, you cannot specify DBPART\*.

When you use the type-2 version of this command, you can specify multiple partition names.

When you use the type-1 version of this command, you can specify only one partition name.

### **SET(RATE)**

Specifies the RATE at which the HALDB OLR runs.

#### ***rate***

You can specify a value of 1 to 100 for the rate values. A HALDB OLRs impact on the system is affected by the available system resources, by total system utilization (including other HALDB OLRs), by total logging volume, by log contention, and by the intensity at which this reorganization was requested to run. These same factors also affect the speed at which the reorganization runs.

You can use the RATE parameter to control the intensity at which the reorganization runs. This can affect both the reorganization's speed and its impact on the rest of the system. The value you specify for rate is the percentage of elapsed time to be devoted to copying records. The remaining time is to be an intentionally introduced delay in the copying process that minimizes the reorganization's impact on other IMS work and on the whole system.

A rate value of 50 specifies that 50% of the elapsed time be spent copying records and the remaining 50% be spent in a delay. This causes the reorganization to run approximately twice as long as it would have run with a rate value of 100.

RATE(100) is the default.

### **OPTION()**

You can specify the following values on the OPTION keyword:

#### **ALTER**

Applies changes that have been made to a database definition, such as an increase in the size of a segment, to an online database via the online reorganization process.

Prior to specifying the ALTER option when the IMS management of ACBs is enabled, the changes must be defined by using one of the following methods:

- Submitting SQL DDL statements.
- Running one of the IMS catalog population utilities with the MANAGEDACBS=STAGED control statement referenced by the SYSINP DD statement.

The new ACB member for the database must be in the staging data set of the IMS catalog.

Prior to specifying the ALTER option when ACB libraries are used, the changes must be defined in DBD generation macros, and both DBD and ACB generation must be complete. The new ACB member for the database must be in the ACB staging library.

When the ALTER keyword is specified, the name of the HALDB master database must be specified on the NAME parameter of the INITIATE OLREORG command. All of the partitions in the database are reorganized.

**Restrictions:**

The ALTER keyword is supported only by the type-2 version of the INITIATE OLREORG command.

The ALTER keyword cannot be specified when one or more partition names are specified on the NAME parameter.

**DEL | NODEL**

Specifies whether IMS attempts to delete the inactive data sets after the reorganization process is complete. DEL and NODEL are mutually exclusive keywords.

When DEL is specified, IMS attempts to delete the inactive data sets when the reorganization process is complete. IMS attempts to delete the data sets regardless of who created the data sets or when the data sets were created.

When NODEL is specified, IMS does not attempt to delete the inactive data sets when the reorganization process is complete.

**Attention:** If the HALDB OLR is stopped before completion, the OPTION(NODEL) specification is not retained by IMS. You must specify OPTION(NODEL) on the INITIATE OLREORG command to resume the stopped HALDB OLR. You can also specify OPTION(NODEL) on the UPDATE OLREORG command.

If the partition is tracked at an RSR tracker site, the DEL or NODEL value in effect at the completion of the HALDB OLR also determines whether the inactive data sets for the shadow partition are deleted when the tracking of the online reorganization is completed.

**REL | NOREL**

Specifies whether the IMS system releases or retains the ownership of the reorganization process if IMS shuts down or terminates before the reorganization process is complete. REL and NOREL are mutually exclusive keywords.

Specifying OPTION(REL) or OPTION(NOREL) with this command overrides any setting in the database section of the DFSDFxxx member of the IMS PROCLIB data set.

**NOREL**

Specifies that the IMS system retains ownership of the reorganization when it shuts down or terminates. NOREL is the default.

**REL**

Specifies that the IMS system releases ownership of the reorganization process if it terminates before the reorganization is completed. Specifying this keyword allows any IMS system to resume the suspended reorganization.

The LIST.DB DBD(*partition\_database*) command shows the owning IMS (OLRIMSID=*ssid*). RELEASE OLR OWNERSHIP is also displayed when an OLR is owned and the REL option is active for the OLR.

When OLR is active and running on an IMS and the IMS terminates abnormally, and if release OLR ownership is requested, the LIST.DB command will show OLRIMSID=*ssid* (instead of OLRIMSID=NULL) and RELEASE OLR OWNERSHIP because IMS was unable to request OLR ownership release.

**Requesting release OLR ownership in an XRF environment:** In an XRF environment, if release OLR ownership is requested while OLR is running on an active IMS, OLR will not automatically resume on the new active IMS after an XRF takeover occurs. You must issue the INIT OLREORG command again on the new active IMS.

## Usage notes

HALDB OLR is not supported for PHDAM and PHIDAM databases that use 8-GB OSAM data sets.

When OPTION(ALTER) is specified, before the command is issued, the ACB members that contain the altered database definition must be in the staging ACB library. After alter processing is complete, Online Change must be performed to activate the ACB members and complete the alter process.

When the /INITIATE OLREORG command is complete and the integrated HALDB Online Reorganization function (OLR) is initiated successfully, a DFS0725I message with a completion code of 0 is sent back to the system console and the MTO as an asynchronous message. If the /INITIATE OLREORG command is issued from a terminal that is not the MTO or system console, the DFS0725I message is also sent to the terminal asynchronously. The DFS0725I message is not sent back to an AOI program that issues the /INITIATE OLREORG command. If the /INITIATE OLREORG command is issued from an LU 6.2 application, the LU 6.2 program receives the DFS0725I message followed by the DFS058I INITIATE COMMAND COMPLETED message.

When OPTION(ALTER) is specified, DFS3197I is also issued to indicate the start of alter processing. When alter processing completes or otherwise stops, DFS3198I is issued.

A type X'22' log record is created when INITIATE OLREORG is issued. The command is non-recoverable and is not reprocessed when IMS is restarted.

The type-1 /INIT OLREORG command can be entered at an IMS terminal, MTO, system console, APPC, OTMA application, or a CMD or ICMD call. You can issue /INIT OLREORG as a type-1 command from a non-OM API using the first three command characters; for example, /INI OLREORG. The type-1 command is processed at the IMS where the command was entered. However, when you issue INIT OLREORG as a type-2 command from an OM API, the only valid command verb form is INIT or INITIATE.

This command can be issued to an IMSplex by using the Batch SPOC utility.

While the database is being quiesced, this command cannot be processed successfully.

Part of the HALDB OLR processing involves:

- The creation of the output data sets at the beginning of the online reorganization

- The optional deletion of the inactive data sets when the HALDB OLR is completed

If the partition is tracked at an RSR tracker site, the OPTION value (DEL or NODEL) that is in effect at the completion of the HALDB OLR also determines whether the inactive data sets for the shadow partition are deleted at the completion of the tracking of the reorganization.

The output for this command is defined in XML and is available to automation programs that communicate with OM if the command is entered from OM API. When entered as a type-1 command, the command output is in message format.

When a HALDB OLR is completed, a DFS2974I message is sent to the system console indicating that the OLR is complete. The command response indicates if an OLR is initiated.

## Command responses for /INITIATE OLREORG

When you issue the type-1 /INITIATE OLREORG command, the command response is returned in a message format.

When the command is completed successfully, message DFS0725I with a completion code of 0 is returned to the system console and to the master terminal. If the command results in an error, a nonzero completion code or an error message is returned to the master terminal and system console.

If the command results in an error, an error message is returned to the entering terminal. One or more of the following messages might also be returned:

```
DFS0725I INITIATE|UPDATE|TERMINATE OLREORG COMMAND FOR DB dbnamexx COMPLETE.
CC= nn
```

where: dbnamexx is the HALDB partition name entered on the command  
nn is the completion code

```
DFS058 - INITIATE COMMAND IN PROGRESS
DFS107 - REQUIRED KEYWORD NOT PRESENT
DFS110 - COMMAND KEYWORD OLREORG INVALID FOR yyyyyyyy (tracker, XRF alt)
DFS128 - TOO FEW KEYWORDS
DFS136 - COMMAND xxxxxxxxxxxx INVALID FOR yyyyyyyy (tracker, XRF alt)
DFS165 - COMMAND VERB IS INVALID
DFS2026 - NO WKAP STORAGE
DFS2038 - INSUFFICIENT STORAGE AVAILABLE TO PROCESS COMMAND
DFS2262 - TOO MANY DATABASE PARAMETERS
DFS3630 - GENERIC PARAMETER NOT ALLOWED
```

## Output fields

This section describes the responses from the OM API for the INITIATE OLREORG command. The following table shows the INITIATE OLREORG output fields and a description of each field:

### Short label

The short label that is generated in the XML output. This field does not apply to the /INITIATE command.

### Show Keyword

The command keyword that caused the field to be generated. N/A appears for output fields that are always returned.

### Meaning

A brief description of the output field.

Table 347. Output fields for INITIATE OLREORG command

Short label	Show Keyword	Meaning
PART	N/A	Partition name.
MBR	N/A	The IMS that processed the command and built the command response line.
CC	N/A	Completion code.

## Return, reason, and completion codes

The OM return and reason codes that might be returned as a result of the INITIATE OLREORG command are standard for all commands that are entered through the OM API.

The following table includes the return and reason codes and a brief explanation of the codes. The return or reason code returned for the command might also indicate an error from a CSL request.

Table 348. Return and reason codes for the INITIATE OLREORG command

Return code	Reason code	Meaning
X'00000000'	X'00000000'	The INITIATE OLREORG command completed successfully.
X'00000004'	X'00001010'	No matches found for filter.
X'00000008'	X'00002008'	Insufficient number of keywords.
X'00000008'	X'00002020'	Invalid rate value.
X'00000008'	X'00002040'	No filter, an invalid filter, or an insufficient number of filters was specified.
X'0000000C'	X'00003000'	At least one request was successful.
X'0000000C'	X'00003004'	None of the requests was successful.
X'00000010'	X'0000400C'	Command issued on an XRF alternate.
X'00000010'	X'00004014'	Command issued on an RSR tracker.
X'00000010'	X'00004030'	Command is invalid for LSO=Y.
X'00000010'	X'00004520'	Another HALDB alter operation is in progress.
X'00000014'	X'00005000'	A GETMAIN error occurred.

The following table includes an explanation of the completion codes. Errors that are unique to the processing of INITIATE OLREORG command are returned as completion codes. A completion code is returned for each action against a HALDB partition. Also, if the type-1 /INITIATE OLREORG command is entered, the completion codes are displayed in the DFS0725I message.

Table 349. Completion codes for the INITIATE OLREORG command

Completion code	Completion code text	Meaning
0		The INITIATE OLREORG command completed successfully for the partition.
10		Resource name is invalid.
14		Resource is not a partition name.

Table 349. Completion codes for the INITIATE OLREORG command (continued)

Completion code	Completion code text	Meaning
17	ANOTHER CMD IN PROGRESS	The INITIATE OLREORG command failed because IMS was already processing another command that was issued against the same database. If OPTION(ALTER) was specified, the IMS system might be processing another INITIATE OLREORG command that has OPTION(ALTER) specified for the same or a different database.
1C		Resource is a partitioned secondary index.
20		HALDB OLR is already in progress.
24		No OLR is in progress.
28		No DMB is loaded.
90		Internal error. The command entered is not processed because of an internal error.
120		Staging ACBLIB empty.
122		Not all PSBS rebuilt for this DMB.
AE		Dynamic allocation error.
C1		OLR database directory (DDIR) is missing.
C3 <sup>®</sup>		OLR partition is either stopped or locked, or is in an input-only access state.
C4		OLR partition is being recovered.
C5		OLR partition access is not UPDATE or EXCLUSIVE.
C6		OLR scheduling failed.
C7		OLR partition is not authorized.
C8		OLR master database is either stopped or locked, or is in an input-only access state.
C9		OLR master database access is not UPDATE or EXCLUSIVE.
CA		OLR ownership setting failed.
CC	OLR IS ACTIVE FOR DATABASE	The INITIATE OLREORG command failed because the OLR function, with or without OPTION(ALTER) specified, is already reorganizing one or more partitions in the database.
CD		HALDB OLR was attempted for an RSR-covered HALDB.
CE		HALDB OLR cannot access all the data in the partition because an EEQE exists.
CF		Parameter value conflict, or invalid parameter value.
DA		The command processing failed because the database is being recalled from HSM.
DF	DB DIRECTORY INIT FAILURE	The DDIR for the database cannot be initialized.
E7		The database command could not be processed because a command that specifies START(QUIESCE) is in progress.

Table 349. Completion codes for the INITIATE OLREORG command (continued)

Completion code	Completion code text	Meaning
1E1	OLR ITASK creation failed	An internal OLR ITASK cannot be created.
1E2	RECON MINVERS or CHANGE# error	In the RECON data set, either the minimum IMS version that is specified in the RECON data set does not support OPTION(ALTER), or the change version number in a CHANGE# field does not match the change version number in the database directory (DDIR) control block for the HALDB database. The minimum IMS version is specified in the RECON by the MINVERS keyword on either the INIT.RECON or the CHANGE.RECON DBRC command.
1E3	Partition queued for OLR	The partition is queued for online reorganization processing by either the OLR function or the OLR function with the ALTER option specified.
1E4	OLC is pending for HALDB alter	An alter request against the same HALDB was done, but an Online Change has not yet done for the HALDB.
1E5	No DB structure change detected	<p>OPTION(ALTER) was specified, but the ACB member for the specified HALDB database does not contain any structural changes.</p> <p>If you intended to change the structure of the database, ensure that the ACB member that contains the database changes is in the ACB staging library. If you are not changing the structure of the database, remove the OPTION(ALTER) keyword from the command.</p>
1E6	Insufficient CI/Block size detected	<p>The size of the block or CI of an output data set is too small.</p> <p>Before initiating the alter operation, increase the size of the block or CI by issuing the DBRC CHANGE.PART command.</p>
1E7	Unsupported DBD changes detected	<p>The ACB member for the specified database contains a structural change that is not supported by the ALTER option.</p> <p>See message DFS1849E for the reason code that identifies the unsupported change.</p>
1E8	Altered DMB not loaded	The DMB for the altered database was not loaded, so the delta block required to validate the structural changes to the database could not be created.
1E9	DB level error detected	During alter processing, a database level error on the altered DBD is detected.
1EA	Segment level error detected	During alter processing, a segment level error on the altered DBD is detected.
1EB	Field level error detected	During alter processing, a field level error on the altered DBD is detected.
1EC	Logical database error	During alter processing, an error related to a logical relationship on the altered DBD is detected.

Table 349. Completion codes for the INITIATE OLREORG command (continued)

Completion code	Completion code text	Meaning
1ED	HALDB structure not init	Required internal HALDB control blocks were not initialized or accessible.
1EE	Alter owned by other IMS	Alter processing could not be resumed on this IMS system because all partitions that remain to be altered are owned by one or more other IMS systems.
1EF	Not OLR capable	The database is not capable of being reorganized online, which can be caused by a HALDB database that uses 8-GB OSAM data sets.

## Examples

The following are examples of the INITIATE OLREORG command.

### *Example 1 for /INITIATE OLREORG command*

Entry ET:

```
/INITIATE OLREORG NAME(HALDBPRT) SET(RATE(50))
```

Response ET:

```
DFS058I INITIATE COMMAND IN PROGRESS
DFS0725I INITIATE OLREORG COMMAND FOR DB HALDBPRT COMPLETE. CC=0
```

System Console or MTO:

```
DFS2970I - OLR STARTED FOR NAME=HALDBPRT
DFS0725I INITIATE OLREORG COMMAND FOR DB HALDBPRT COMPLETE. CC= 0
```

Explanation: The /INITIATE OLREORG command for the partition named HALDBPRT is issued from a terminal, resulting in the message, DFS058I INITIATE COMMAND IN PROGRESS being sent to that entering terminal. When OLR is started successfully, the DFS0725I and DFS2970I messages are sent to the system console and to the MTO. The DFS0725I message is also sent to the entering terminal if the entering terminal is not the system console, MTO, AOI program.

### *Example 2 for /INITIATE OLREORG command*

Entry ET:

```
/INIT OLREORG NAME(PDHDOKA) SET(RATE(5))
```

Response ET:

```
DFS058I INITIATE COMMAND IN PROGRESS
DFS0725I INITIATE OLREORG COMMAND FOR DB PDHDOKA COMPLETE. CC=0
```

Response MTO or System console:

```
DFS2970I - OLR STARTED FOR NAME=PDHDOKA
DFS0725I INITIATE OLREORG COMMAND FOR DB PDHDOKA COMPLETE. CC=0
```

Explanation: The INIT OLREORG command for the partition named PDHDOKA is issued from a terminal, resulting in the DFS058I INITIATE COMMAND IN PROGRESS

message being sent to that entering terminal. The DFS0725I message is also sent to the entering terminal if the entering terminal is not the system console, the MTO, or an AOI program.

When OLR is started successfully, the DFS0725I and the DFS2970 messages are sent to the system console and to the MTO.

### *Example 3 for INITIATE OLREORG command*

TSO SPOC Input:

```
INIT OLREORG NAME(PDHOKA,PDHOKB) SET(RATE(5))
```

TSO SPOC output:

Partition	MbrName	CC
PDHOKA	IMSA	0
PDHOKA	IMS1	C3
PDHOKB	IMSA	0
PDHOKB	IMS1	C3

OM API Input:

```
CMD (INIT OLREORG NAME(PDHOKA,PDHOKB) SET(RATE(5)))
```

OM API output:

```
<imsout>
<ctl>
<omname>OM10M </omname>
<omvsn>1.2.0</omvsn>
<xmlvsn>1 </xmlvsn>
<statime>2003.168 21:19:06.827362</statime>
<stotime>2003.168 21:19:07.036292</stotime>
<staseq>B99629BFEC66256E</staseq>
<stoseq>B99629C01F684F44</stoseq>
<rqsttkn1>USRT005 10141906</rqsttkn1>
<rc>00000000</rc>
<rsn>00000000</rsn>
</ctl>
<cmd>
<master>IMSA </master>
<userid>USRT005 </userid>
<verb>INIT</verb>
<kwd>OLREORG </kwd>
<input>INIT OLREORG NAME(PDHOKA,PDHOKB) SET(RATE(5)) </input>
</cmd>
<cmdrsphdr>
<hdr s1b1="PART" l1b1="Partition" scope="LCL" sort="A" key="1"
  scroll="NO" len="9" dtype="CHAR" align="left" />
<hdr s1b1="MBR" l1b1="MbrName" scope="LCL" sort="N" key="0" scroll="NO"
  len="8" dtype="CHAR" align="left" />
<hdr s1b1="CC" l1b1="CC" scope="LCL" sort="N" key="0" scroll="YES"
  len="4" dtype="INT" align="right" />
</cmdrsphdr>
<cmdrspdata><rsp> PART(PDHOKA ) MBR(IMSA ) CC( 0) </rsp>
<rsp> PART(PDHOKB ) MBR(IMSA ) CC( 0) </rsp>
<rsp> PART(PDHOKA ) MBR(IMS1 ) CC( C3) </rsp>
<rsp> PART(PDHOKB ) MBR(IMS1 ) CC( C3) </rsp>
</cmdrspdata>
</imsout>
```

Explanation: The INIT OLREORG command is issued from TSO SPOC for partitions PDHDOKA and PDHDOKB. The command is routed to IMSA and IMS1. The command is successful at IMSA for both partitions, but is not successful at IMS1 for both partitions.

**Example 4: INITIATE OLREORG command with OPTION(ALTER)**

In the following example, the INITIATE OLREORG command with the ALTER option is issued for the HALDB database DBOVLFPC, which contains four partitions.

Two IMS systems are participating in the alter operation, IMS1 and IMS2. IMS1 gets ownership of partitions PDBOVLA and PDBOVLC. IMS2 gets ownership of PDBOVLB and PDBOVL D.

When alter processing stops in an IMS system, the IMS system issues message DFS3198I.

**TSO SPOC Input:**

```
INIT OLREORG OPTION(NODEL,ALTER,NOREL) SET(RATE(100)) NAME(DBOVLFPC)
```

**TSO SPOC output:**

Partition	MbrName	CC
PDBOVLA	IMS2	CA
PDBOVLA	IMS1	0
PDBOVLB	IMS2	0
PDBOVLB	IMS1	CA
PDBOVLC	IMS2	CA
PDBOVLC	IMS1	0
PDBOVL D	IMS2	0
PDBOVL D	IMS1	CA

**OM API Input:**

```
CMD (INIT OLREORG OPTION(NODEL,ALTER,NOREL) SET(RATE(100)) NAME(DBOVLFPC))
```

**OM API output:**

```
<statime>2013.217 22:36:48.063386</statime>
<stotime>2013.217 22:36:50.823178</stotime>
<staseq>CBC4B3A59D79A7C2</staseq>
<stoseq>CBC4B3A83F40A950</stoseq>
<rqsttkn1>USRT011 10153648</rqsttkn1>
[...]
<cmd>
<master>IMS2 </master>
<userid>USRT011 </userid>
<verb>INIT</verb>
<kwd>OLREORG </kwd>
<input>INIT OLREORG OPTION(NODEL,ALTER,NOREL) SET(RATE(100))
NAME(DBOVLFPC) </input>
</cmd>
<cmdsphdr>
<hdr s1bl="PART" l1bl="Partition" scope="LCL" sort="A" key="1"
scroll="NO" len="7" dtype="CHAR" align="left" />
<hdr s1bl="MBR" l1bl="MbrName" scope="LCL" sort="N" key="0" scroll="NO"
len="8" dtype="CHAR" align="left" />
<hdr s1bl="CC" l1bl="CC" scope="LCL" sort="N" key="0" scroll="YES"
len="4" dtype="INT" align="right" />
</cmdsphdr>
<cmdspdata>
<rsp> PART(PDBOVLA ) MBR(IMS2 ) CC( CA) </rsp>
<rsp> PART(PDBOVLB ) MBR(IMS2 ) CC( 0) </rsp>
```

```
<rsp> PART(PDBOVL C ) MBR(IMS2 ) CC( CA) </rsp>  
<rsp> PART(PDBOVL D ) MBR(IMS2 ) CC( 0) </rsp>  
<rsp> PART(PDBOVL A ) MBR(IMS1 ) CC( 0) </rsp>  
<rsp> PART(PDBOVL B ) MBR(IMS1 ) CC( CA) </rsp>  
<rsp> PART(PDBOVL C ) MBR(IMS1 ) CC( 0) </rsp>  
<rsp> PART(PDBOVL D ) MBR(IMS1 ) CC( CA) </rsp>  
</cmdrspdata>
```

Each IMS system issues DFS3198I when alter processing stops. The message shows the number of partitions in the database, the number of partitions that the IMS system started altering, and the number of partitions that the IMS system successfully altered.

```
DFS3198I HALDB ALTER ENDED FOR NAME=DBOVL FPC  
PARTITION NUMBER=4 STARTED=2 COMPLETED=2 IMS1
```

```
DFS3198I HALDB ALTER ENDED FOR NAME=DBOVL FPC  
PARTITION NUMBER=4 STARTED=2 COMPLETED=2 IMS2
```

**Related concepts:**

 [How to interpret CSL request return and reason codes \(System Programming APIs\)](#)

**Related reference:**

Chapter 2, “Command keywords and their synonyms,” on page 63



---

## Chapter 37. /LOCK command

The /LOCK command stops the sending and receiving of messages to and from a terminal, stops the scheduling of messages containing a specific transaction code, stops the scheduling of a specific program, and stops the use of a database.

Queuing of output messages for a particular communication line, terminal, or transaction code is not affected.

Subsections:

- “Environment”
- “Syntax”
- “Keywords” on page 950
- “Usage notes” on page 951
- “Equivalent IMS type-2 commands” on page 952
- “Examples” on page 952

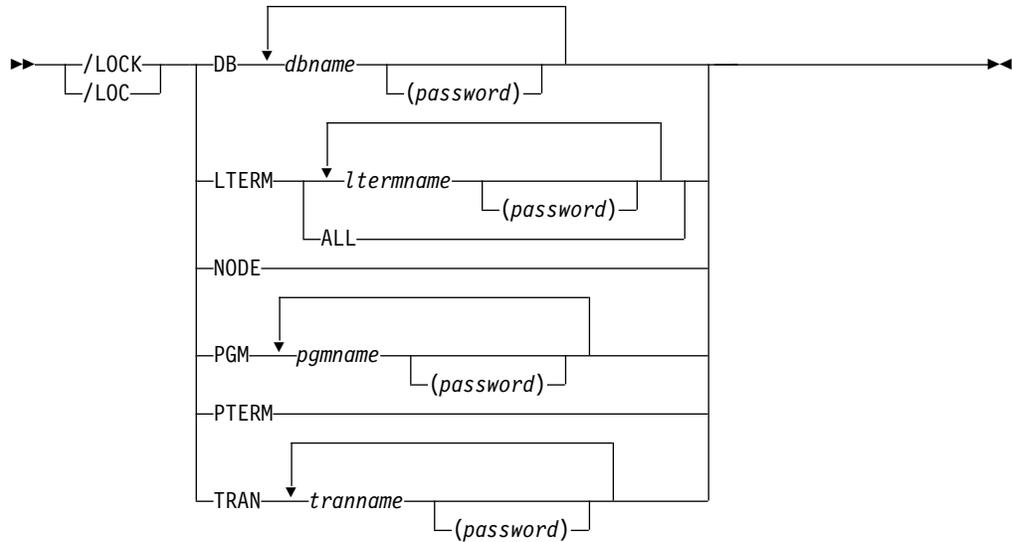
### Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

*Table 350. Valid environments for the /LOCK command and keywords*

Command / Keywords	DB/DC	DBCTL	DCCTL
/LOCK	X	X	X
DB	X	X	
LTERM	X		X
NODE	X		X
PGM	X	X	X
PTERM	X		X
TRAN	X		X

### Syntax



## Keywords

The following keywords are valid for the /LOCK command:

**DB** LOCK prevents subsequently scheduled programs from accessing the database. The /LOCK DB command does not close the database or affect currently scheduled programs.

If the database is a DEDB or MSDB, programs using the database will not be scheduled. For other databases, the programs will still be scheduled. If the INIT call was issued, however, a call against the database will result in either a 3303 pseudoabend or a BA status code.

For DBCTL, CCTL can specify LONG or SHORT when it schedules a PSB. If the database is currently scheduled to a LONG thread, the command is rejected. If not, the thread completes before the database is acted upon. If the thread completes, a commit point or transaction termination is the result.

While the database is being quiesced, this command cannot be processed successfully.

### LTERM

Specifies the logical terminal for which sending and receiving of messages is to be stopped. /LOCK LTERM applies only to logical terminals associated with the entering physical terminal.

The /LOCK LTERM ALL command can only be used when none of the logical terminals associated with the entering physical terminal have passwords.

### NODE

Specifies that the sending and receiving of messages for the VTAM node associated with the entering physical terminal is to be stopped.

### PGM

Specifies the program for which scheduling is to be stopped.

### PTERM

Specifies that the sending and receiving of messages for the entering physical terminal is to be stopped.

## TRAN

Specifies that scheduling of messages containing this transaction code is to be stopped. The /LOCK TRAN command cannot be used with Fast Path exclusive transactions but can be used with Fast Path potential transactions.

The /LOCK TRAN command cannot be used for CPI Communications driven programs.

## Usage notes

The /LOCK command can be used on HALDB databases.

The output of the /LOCK DB command is changed when the command is entered through the OM API. In this case, the DFS058I message is not returned to OM. The command response returned to OM contains one or more of the following messages: DFS0488I, DFS3466I, or DFS132. The OM API does not support the /LOCK command without any keywords.

The /LOCK command can be issued by APPC and OTMA clients.

This command can be issued to an IMSplex by using the Batch SPOC utility.

A resource name can be defined with password protection in SAF for the DB, LTERM, PGM, and TRAN keywords. If the parameter, LOCKSEC=Y (N is the default) is specified on the DFSPBxxx IMS.PROCLIB member, IMS calls the SAF and user exit calls. If the resource is not defined to SAF, or is defined and is authorized to the user, the command is processed. If the resource is defined to SAF but not authorized for use, the command is rejected with a DFS3689W message.

The password associated with a signed on user, and specified after a command resource parameter, will be used to perform a reverification check, if the resource is defined to RACF with 'REVERIFY' specified in the APPLDATA field. Passwords can be mixed case or uppercase depending on what is specified on the PSWDC keyword in the DFSPBxxx IMS.PROCLIB member. If the resource passes the RACF authorization check, and RVFY=Y is specified as an IMS startup parameter, IMS will verify that the password following the parameter is the same as the password entered during signon for the user that entered the command. If 'REVERIFY' is specified for a resource, but a password is not provided, or the wrong password is provided, the command processing for that resource will be rejected.

A period that is specified in the password brackets is not treated as a delimiter.

The /LOCK LTERM, NODE, and PTERM commands apply only to the entering physical terminal. The /LOCK DB | PGM | TRAN command is only valid if entered from the master terminal, the system console, a TCO script, or an AOI application program.

Issuing the /LOCK command for a physical terminal, logical terminals, or nodes prevents application program output from being sent (assuming nonresponse mode transactions) and message switches from any other terminals. Output message queuing continues for the locked physical terminal. The /STOP LTERM command can be used instead of the /LOCK LTERM command, because both stop the sending and receiving of messages to the physical terminal.

If the terminals are on a switched communication network and a physical or logical terminal disconnection occurs, an implied /UNLOCK command is processed against the physical terminal and inquiry logical terminal.

The /LOCK command cannot be used with logical terminals assigned to deallocated users.

When you enter this command, the database name can be an existing non-HALDB, a HALDB master, or a HALDB partition. A command against a HALDB partition operates exactly like a command against a non-HALDB with the exception of the /START DB and the UPDATE DB START(ACCESS) commands. A HALDB partition is not allocated during the command unless it was previously authorized but not allocated, the OPEN keyword was specified, or the partition has EEQEs. The partition is allocated at first reference.

The HALDB partition reflects conditions such as STOPPED, LOCKED, or NOTOPEN. When a HALDB partition is stopped, it must be explicitly started again. Commands with the keyword ALL and commands against a HALDB master do not change the STOPPED and LOCKED indicators in each HALDB partition.

When the command target is a HALDB master, processing acts on all HALDB partitions. For example, if the IMS command is /DBR on the HALDB master, all of the HALDB partitions are closed, deallocated, and deauthorized. Only the HALDB master displays STOPPED (each HALDB partition does not display STOPPED unless it was itself stopped). If a /DBR command was issued against a HALDB master, the display output of a /DISPLAY DB command shows the HALDB master (as STOPPED), but does not display the status of the partitions.

Each partition inherits the access limitations of its HALDB master. If the /DBD command is issued against a HALDB master, all of its partitions close. A subsequent reference to any of the partitions results in the partition opening for input, although the partition's access might be UPDATE or EXCLUSIVE. The DBRC authorization state reflects the limited access.

## Equivalent IMS type-2 commands

The following table shows variations of the /LOCK command and the IMS type-2 commands that perform similar functions.

*Table 351. Type-2 equivalents for the /LOCK command*

Task	/LOCK command	Similar IMS type-2 command
Stops the use of a database.	/LOCK DB <i>dbname</i>	UPDATE DB NAME( <i>dbname</i> ) SET(LOCK(ON))
Locks a program.	/LOCK PGM <i>pgmname</i>	UPDATE PGM NAME( <i>pgmname</i> ) SET(LOCK(ON))
Locks a transaction.	/LOCK TRAN <i>tranname</i>	UPDATE TRAN NAME( <i>tranname</i> ) SET(LOCK(ON))

## Examples

The following are examples of the /LOCK command:

### *Example 1 for /LOCK command*

Entry ET:

```
/LOCK DB TREEFARM
```

Response ET:

```
DFS058I LOCK COMMAND COMPLETED
```

Explanation: Application programs that use the MSDB database named TREEFARM are no longer scheduled. Transaction input can continue.

*Example 2 for /LOCK command*

Entry ET:

```
/LOCK LTERM ALL
```

Response ET:

```
DFS058I LOCK COMMAND COMPLETED
```

Explanation: Output is queued but not sent to the logical terminals associated with the physical terminal from which the /LOCK command was entered.

*Example 3 for /LOCK command*

Entry ET:

```
/LOCK PGM APPLETRE
```

Response ET:

```
DFS058I LOCK COMMAND COMPLETED
```

Explanation: Program APPLETRE can no longer be scheduled.

*Example 4 for /LOCK command*

Entry ET:

```
/LOCK PTERM
```

Response ET:

```
DFS058I LOCK COMMAND COMPLETED
```

Explanation: Application program output (if nonresponse mode transaction) and message switches from other terminals are not sent to the entering physical terminal. Output is queued for later transmission when the terminal is unlocked.

*Example 5 for /LOCK command*

Entry ET:

```
/LOCK TRAN SEED
```

Response ET:

```
DFS058I LOCK COMMAND COMPLETED
```

Explanation: The application program for transaction code SEED can no longer be scheduled. Transaction input can continue.

Entry ET:

/LOCK TRAN TXCDLM06

Response ET:

DFS058I LOCK COMMAND COMPLETED EXCEPT TRANSACTION  
TXCDLM06

Explanation: TXCDLM06 is a Fast Path exclusive transaction. The /LOCK command cannot stop the scheduling of Fast Path exclusive transactions.

**Related concepts:**

[↗](#) Sharing secondary index databases (Database Administration)

**Related reference:**

[↗](#) UPDATE DB command (Commands)

[↗](#) UPDATE PGM command (Commands)

[↗](#) UPDATE TRAN command (Commands)

---

## Chapter 38. /LOG command

The /LOG command writes a single-segment message to the IMS system log. Any text can be logged, including transaction codes or command streams.

Subsections:

- “Environment”
- “Syntax”
- “Keywords”
- “Usage notes”
- “Example”

### Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) from which the command can be issued.

Table 352. Valid environments for the /LOG command

Command	DB/DC	DBCTL	DCCTL
/LOG	X	X	X

### Syntax

►► /LOG *text* ◀◀

### Keywords

The following keywords are valid for the /LOG command:

***text***

An alphanumeric character message to be logged.

### Usage notes

An X'02' log record is produced and contains the text. This command applies only to the currently entered message segment and does not establish a continuing operational mode.

This command can be issued to an IMSplex using the Batch SPOC utility.

### Example

Entry ET:

```
/LOG TODAY IS MONDAY
```

Response ET:

```
DFS058I LOG COMMAND COMPLETED
```

Explanation: An X'02' log record containing the text

TODAY IS MONDAY

is written to the IMS system log.

---

## Chapter 39. /LOOPTEST command

The /LOOPTEST command is a multisegment command that creates an output write loop that tests for output errors.

It causes a single-segment message to be repeatedly transmitted to the terminal being tested.

Subsections:

- “Environment”
- “Syntax”
- “Usage notes”
- “Example” on page 958

### Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

Table 353. Valid environments for the /LOOPTEST command and keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
/LOOPTEST	X		X
LINE	X		X
PTERM	X		X

### Syntax

► `/LOOPTEST` `LINE` *line#* `PTERM` *pterm#* ◄

`/LOO`

### Usage notes

Multisegment input is combined into a single segment for output transmission. For the terminal in looptest mode to receive the complete message, the cumulative output segment must not exceed the terminal's output size restrictions.

/LOOPTEST requires an EOM indication to denote end-of-message; an EOS indication must be included for all segments that precede the last segment.

/LOOPTEST causes all I/O error detection and notification procedures within IMS to be bypassed. Depending on the types of errors involved, some write loops might result in high processor utilization.

The /END command terminates looptest mode.

#### LINE, PTERM

Specifies the communication line or physical terminal to be placed into looptest mode. If the terminal in looptest mode is attached to a multipoint line, the

entire line must be stopped and idle before the /LOOPTEST command is allowed. /LOOPTEST is not allowed on VTAM-attached terminals.

If the device specified is not powered on and ready, intervention-required messages will be sent to the master terminal.

The /DISPLAY LINE PTERM command identifies a terminal in looptest mode.

### **Example**

Entry ET:

```
/LOOPTEST LINE 5 PTERM 1 (EOS)
DATA TEST (EOS)
12345 (EOM)
```

Response ET:

```
DFS058I LOOPTEST COMMAND COMPLETED
```

Response RT:

```
DATA TEST 12345
DATA TEST 12345
and so forth.
```

Explanation: DATA TEST 12345 is repetitively sent to PTERM 1 on LINE 5.

#### **Related reference:**

“How to enter IMS commands” on page 2



## Keywords

The following keywords are valid for the /MODIFY command:

### ABORT

Resets the status that was set by the /MODIFY PREPARE command after a /MODIFY PREPARE, or can be used if the /MODIFY COMMIT was not successful and the operator chooses not to attempt the online change at that time.

### COMMIT

Brings all the newly defined resources online, updates the changed resources, and invalidates the deleted resources.

If ALL was specified on the /MODIFY PREPARE command, the ACBLIB, FMTLIB, and MODBLKS changes must be successful or the /MODIFY COMMIT will be rejected. When COMMIT is successful, the modifications will persist across all IMS restarts. If FMTLIB is entered, most new and in-progress requests will be queued for processing after the /MODIFY COMMIT command is completed.

If MODBLKS or ACBLIB is specified, work in progress for changed or deleted resources in the IMS.ACBLIB data set or in the IMS.MODBLKS data set that are directly affected must be completed, or the /MODIFY command is rejected.

If DRD is enabled and /MODIFY PREPARE ALL was specified, the /MODIFY COMMIT command does not swap the MODBLKS library suffix. The DFS3499 message shows the same MODBLKS suffix before and after the online change.

IMS does not reject the /MODIFY COMMIT command if the online change indirectly affects transactions (having work in progress) that reference resources in the ACB library or in the MODBLKS data set that are directly affected by the online change. These transactions themselves are not being changed (no attribute on the TRANSACT macro is being changed), but they reference a PDIR, PSB, or DMB that is affected by the online change. A transaction can be indirectly affected by an online change in one of the following situations:

- The transaction references a PSB directory (PDIR) control block in the IMS.MODBLKS data set that is being changed
- The transaction references a PSB in the ACB library (IMS.ACBLIB) that is being changed online
- The transaction references a PSB directory (PDIR) control block in the IMS.MODBLKS data set which in turn references a database that is being changed online
- The transaction references a PSB in the ACB library (IMS.ACBLIB) that references a database that is being changed online

/DISPLAY MODIFY is used to display the work that was in progress prior to the entering of /MODIFY COMMIT.

IMS will prevent messages entered from terminals from being queued to transactions, but program-to-program switches from applications can still occur. Therefore, a /DISPLAY MODIFY command can indicate that no work is pending for the resources to be modified or deleted, but the /MODIFY COMMIT might be rejected. If this occurs, the new work must be allowed to complete and then the /MODIFY COMMIT command must be reentered.

/MODIFY COMMIT is rejected if:

- A /MODIFY PREPARE command was not previously entered.

- Transactions to be deleted have messages queued.
- Transactions to be changed by the following system definition keywords have messages queued:
  - MSGTYPE
  - INQUIRY
  - FPATH
  - EDIT
  - SPA
- Transactions to be changed by keywords not shown in the previous bullet have messages queued, and the transactions have not been the object of a /STOP or /PSTOP command.
- Transactions with access to programs or databases to be changed or deleted are prevented from updating, and the transactions have not been the object of a /STOP or /PSTOP command. When a transaction shows a status of USTOP during a /DISPLAY TRAN, the /STOP command might need to be entered, because the /PSTOP command might not enable the /MODIFY command to complete the transaction.
- Any program (PSB) that is currently scheduled has a program, database, or transaction change or deletion. This includes Wait for Input (WFI) and Fast Path transactions. All WFI and Fast Path regions that reference changed or deleted routing codes, programs, or databases must be stopped before entering the /MODIFY COMMIT.
- An I/O error occurs while reading the inactive MFS FORMAT library directory or the resident index (\$\$IMSDIR) member.
- A /START DATABASE command is in progress for any database that is changed or deleted.
- A /DBDUMP or /DBRECOVERY command is in progress for any database that is changed or deleted.

Operator action might be required to prevent the preceding conditions. For example, if a program or database is scheduled, the operator should either wait until the program or database is finished before the /MODIFY COMMIT command is entered, or the operator should issue /STOP or /PSTOP for the associated transaction.

If IMS is in an IMSplex and Resource Manager (RM) is using a resource structure, the transactions being added by the online change are registered to RM if they are not already registered to RM. If any error occurs during the CSLRMUPD request to register the transactions, the /MODIFY or INITIATE 0LC command will fail and IMS will be in a COMMIT2F state.

If a /MODIFY COMMIT command results in the COMMIT2F state, the command is rejected with a DFS178 message. A DFS3308 message is sent to the system console with the CSLRMUPD return/reason code. The IMS remains in a COMMIT2F state and the /MODIFY cannot be committed or aborted. The IMS must be canceled and warm started as the online change is not complete.

#### **PREPARE**

/MODIFY PREPARE is not recovered across an emergency restart and must be reentered after restart if the /MODIFY COMMIT did not complete prior to an IMS failure.

The parameters in the following table are valid with /MODIFY PREPARE. The environments in which the parameters are valid are also shown in this table.

Table 355. /MODIFY PREPARE parameters environments

Parameter	DB/DC	DBCTL	DCCTL
ACBLIB	X	X	X
ALL	X	X <sup>1</sup>	X
BLDL	X	X	X
FMTLIB	X		X
LTERM	X		X
MODBLKS	X	X	X
RACF	X		X

**Note:**

1. Only applies to ACBLIB, MODBLKS, and BLDL.

**ACBLIB**

Quiesces the transactions to be changed or deleted and transactions that access databases or programs to be changed or deleted. ACBLIB members that are not found are displayed as CHANGED because a DDIR or PDIR exists in MODBLKS. Additions or deletions to ACBLIB require the MODBLKS parameter.

If a DMB is being added to the ACBLIB with no MODBLKS change (no DMB exists in the active ACBLIB and a DMB exists in the inactive ACBLIB), then the display will show ADDED instead of CHANGED. If a DMB is being deleted from the ACBLIB with no MODBLKS change (DMB exists in the active ACBLIB and no DMB exists in the inactive ACBLIB), then the display will show DELETED instead of CHANGED. No MODBLKS change means a DDIR exists in the MODBLKS and the same MODBLKS is used before and after the online change.

The /MODIFY command is rejected if it is issued with the ACBLIB keyword on an IMS system in which the IMS management of ACBs is enabled. IMS systems that manage ACBs do not use an ACB library.

**ALL**

Indicates that /MODIFY PREPARE is to be performed for ACBLIB, FMTLIB, RACF, MODBLKS and BLDL.

If the ALL keyword is specified when both dynamic resource definition (DRD) and the IMS management of ACBs are enabled, the IMS system rejects the /MODIFY command.

ACBLIB is ignored if IMS management of ACBs is enabled.

MODBLKS is ignored if DRD is enabled.

FMTLIB is ignored in a DBCTL system.

In a DBCTL IMS with IMS management of ACBs and DRD enabled the command with ALL is rejected.

**BLDL**

Causes information message DFS3498I to be issued for each PSB or DMB not found in ACBLIB. This message does not affect the success of the command.

**FMTLIB**

Indicates that there will be no restrictions on the use of MFS format blocks until the /MODIFY COMMIT command is entered.

**LTERM *ltermname***

Specifies a logical terminal to receive lists of added, changed, or deleted resources.

**MODBLKS**

Causes IMS to prepare to bring the changes or deletions defined by the new system definition to IMS. Using MODBLKS also quiesces the transactions to be changed or deleted and transactions that access databases or programs to be changed or deleted. This command is rejected if it is issued on an IMS that has DRD enabled.

**RACF**

Use of the RACF data space invalidates the IMS online change support for RACF with the /MODIFY command. The IMS online change support is still valid, though, when the RACF data space is not being used.

The message DFS3432 RACF PARAMETER INVALID IF RACF DATA SPACE IS USED is issued if the RACF parameter is used on the /MODIFY PREPARE command when the RACF data space is being used. You can use the RACF command SETROPTS RACLIST (classname) REFRESH to refresh the RACF resource profiles in the RACF data space without requiring the IMS applications to suspend work.

**Usage notes**

The following table lists the resources and parameters available to the /MODIFY command:

*Table 356. Resources and parameters available to the /MODIFY command*

Parameter	Library	Description
ACBLIB	IMS.ACBLIB <sub>x</sub> <sup>1</sup>	Contains database and program descriptors such as DMBs and PSBs.
FMTLIB	IMS.FORMAT <sub>x</sub> <sup>1</sup>	Contains MFS maps produced by the MFS language utility and service utility. The FMTLIB parameter is not valid with DBCTL.
MODBLKS	IMS.MODBLKS <sub>x</sub> <sup>1</sup>	Contains a subset of control blocks produced by IMS system definition.
RACF		RACF in-storage profiles. The RACF parameter is not valid with DBCTL.

**Note:**

1. <sub>x</sub> = A or B
2. Use the SETROPTS RACLIST(CIMS) REFRESH command to refresh RACF updates.

The correct sequence for /MODIFY commands is PREPARE followed by COMMIT. If the online change is not to be completed, the /MODIFY ABORT command is used. The /MODIFY PREPARE command specifies which resources are to be changed or deleted. Depending on the parameters entered, the system will initiate quiescing of appropriate resources. COMMIT initiates the changes entered in a preceding PREPARE. If a previously entered /MODIFY PREPARE command is to be canceled, ABORT must be entered.

An IMS automated operator program can issue the /MODIFY PREPARE and /MODIFY ABORT commands, but it cannot issue the /MODIFY COMMIT command. Thus, you can automatically prepare or abort online change.

If this command is used to delete a database, IMS closes the database and makes it unavailable to programs. Also, if the database is authorized for DBRC, /MODIFY deauthorizes it.

Fast Path DEDBs are supported by the /MODIFY command, but Fast Path MSDBs are not supported.

In an IMSplex, /MODIFY is not supported if coordinated online change is enabled. This command can be issued to an IMSplex using the Batch SPOC utility.

If DRD is enabled, the /MODIFY command specified with the MODBLKS keyword alone or with other keywords is rejected. The command /MODIFY PREPARE ALL ignores resources in the MODBLKS data set. If IMS is started without the MODBLKS DD cards, the MODBLKS-name field in the DFS3499 message is blank, which means that the MODBLKS data set is not applicable. This is consistent with what DBCTL does with FMTLIB. If IMS is started with the MODBLKS DD statements, the MODBLKS-name field is displayed, with a name suffix of A during prepare and during commit, which means that the MODBLKS libraries were not changed. The ACBLIB and FMTLIB data sets are still supported if DRD is enabled.

An online change of a PSB or DBD will remove the member from the 31-bit pool as well as the 64-bit pool.

If a resident program is changed, it becomes nonresident locally because its definition in the resident PSB pool is no longer valid. If its resident definition attribute was not changed, it becomes resident again at the next IMS restart. If a nonresident program is changed to be resident, the residency takes effect at the next IMS restart.

## Examples

The following are examples of the /MODIFY command:

### *Example 1 for /MODIFY command*

This is an example of a successful PREPARE, COMMIT sequence. In this example, DRD is not enabled. First, the command /MODIFY PREPARE ALL is issued.

Entry ET:

```
/MODIFY PREPARE ALL
```

Response ET:

```
DFS3499I ACTIVE DDNAMES MODBLKSA IMSACBA FORMATA MODSTAT ID: 103
```

Explanation: The modifications are brought online to IMS during a subsequent successful /MODIFY COMMIT command. The ddnames of the active data sets and the current MODSTAT identifier are shown. This message is the normal response.

### *Example 2 for /MODIFY command*

The following example show the modifications from a /MODIFY PREPARE ACBLIB LTERM WTOR command.

Entry ET:

```
/MODIFY PREPARE ACBLIB LTERM WTOR
```

Response ET:

```
DFS3499I ACTIVE DDNAMES: MODBLKSA IMSACBA FORMATA MODSTAT ID:  
DFS3430I THE FOLLOWING MODIFICATIONS WILL OCCUR DURING /MODIFY COMMIT:  
DFS3430I DMB      OLCDB115      CHANGED  
DFS3430I DMB      OLCDB116      CHANGED  
DFS3430I DMB      OLCDB117      CHANGED  
DFS3430I DMB      OLCDB118      CHANGED  
DFS3430I DMB      OLCDB119      CHANGED  
DFS3430I PSB      BMP255        NOT CHANGED(DMB CHANGED ONLY)  
DFS3430I END OF MODIFICATIONS
```

Explanation: The PSB references a DBD that is changed, but it is not required to be rebuilt. The DBDs' PSB copies are purged, and the PSB is read from ACBLIB the next time it is scheduled.



---

## Chapter 41. /MONITOR command

The /MONITOR command stops output from being sent to programmable remote stations (System/3). Input and output message queuing are allowed to continue.

The /MONITOR command can be used to reset conditions previously established by the /START, /RSTART, /STOP, /PSTOP, or /PURGE command.

This command can be issued to an IMSplex using the Batch SPOC utility.

Subsections:

- "Environment"
- "Syntax"
- "Keywords"
- "Example"

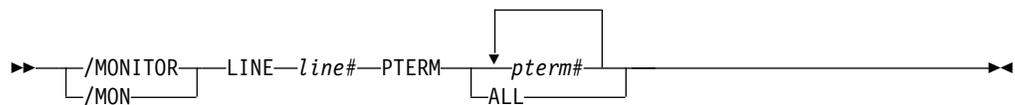
### Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the command and keywords.

Table 357. Valid environments for the /MONITOR command and keywords

Command / Keywords	DB/DC	DBCTL	DCCTL
/MONITOR	X		X
LINE	X		X
PTERM	X		X

### Syntax



### Keywords

The following keywords are valid for the /MONITOR command:

#### LINE, PTERM

Specifies the communication line or physical terminal for which output is being stopped.

### Example

Entry ET:

```
/MONITOR LINE 4 PTERM 3
```

Response ET:

```
DFS058I MONITOR COMMAND COMPLETED
```

Response RT:

DFS059I TERMINAL MONITORED

Explanation: No further output is allowed to PTERM 3 on LINE 4.



## Keywords

The following keywords are valid for the /MSASSIGN command:

### LINK

Specifies the logical link in a multiple system configuration.

Logical links can be assigned to physical links for input/output purposes. Only one logical link at a time can be assigned to a physical link (except for VTAM MSPLINKs, which can have multiple logical links assigned to them). The logical link must be stopped and idle for this assignment. Systems can only communicate with each other across a logical link established through specification of the same partner ID in both systems.

This type of assignment must be synchronized between the master terminal operators of the two systems. If a connection becomes inoperable while a transmission is in progress, this type of assignment allows switching to an alternate connection and proceeding with the unfinished transmission.

### LOCAL

Specifies the local system. that is the subsystem in which the command is issued. The command /MSASSIGN TRAN p1 TO LOCAL assigns transaction p1 to the local system.

### MSNAME

Specifies the logical link path in a multiple systems configuration.

### MSPLINK

Specifies the physical link in a multiple subsystem configuration.

The command /MSASSIGN LINK p1 MSPLINK p2 assigns the logical link p1 to the physical link p2 for input and output purposes.

### SYSID

Specifies the system identification of a remote system in a multiple system configuration. Remote SYSIDs can be assigned to logical links for output purposes. More than one SYSID (=MSNAME) can be assigned to one logical link. A remote SYSID (=MSNAME) cannot be assigned to another logical link if a message is currently transmitting on the connection corresponding to this SYSID. Therefore, the logical link must be stopped and idle for this assignment.

### TRAN

Specifies the transaction to be assigned to either the local or the remote system.

For successful completion of the command:

- The transactions must be stopped.
- When assigning a remote transaction to local, a local APPLCTN macro (no SYSID parameter) for the corresponding program (PSB) must have been included in the system definition. There must be a local as well as a remote PSB of the same name defined during system definition.

The /MSASSIGN TRAN command cannot be used for the CPI Communications driven transaction program.

## Usage notes

All changes made by an /MSASSIGN command remain until changed with another /MSASSIGN command or an IMS cold start is performed. After the /MSASSIGN command is used, /MSVERIFY should be used to ensure that the assignment produced a valid configuration.

The `/MSASSIGN TRAN tranname TO LOCAL` command sets the remote system ID to be the same as the local system ID. It is equivalent to the command `UPDATE TRAN NAME(tranname) SET(REMOTE(N),SIDR(localsysid),SIDL(localsysid))`.

This command can be issued to an IMSplex by using the Batch SPOC utility.

## Equivalent IMS type-2 commands

The following table shows variations of the `/MSASSIGN` command and the IMS type-2 commands that perform similar functions.

*Table 359. Type-2 equivalents for the /MSASSIGN command*

Task	/MSASSIGN command	Similar IMS type-2 command
Changes the transaction so that it is local and runs on the local system.	<code>/MSASSIGN TRAN tranname TO LOCAL</code>	<code>UPDATE TRAN NAME(tranname) SET(REMOTE(N))</code>
Changes the transaction so that it is remote, and assign it to a specific logical link path.	<code>/MSASSIGN TRAN tranname TO MSNAME msname</code>	<code>UPDATE TRAN NAME(tranname) SET(MSNAME(name))</code>

## Examples

The following are examples of the `/MSASSIGN` command:

### *Example 1 for /MSASSIGN command*

Entry ET:

```
/MSASSIGN LINK 2 TO MSPLINK BBB
```

Response ET:

```
DFS058I MSASSIGN COMMAND COMPLETED
```

Explanation: The input system now has a logical connection to physical link BBB for all the SYSIDs assigned to link 2.

### *Example 2 for /MSASSIGN command*

Entry ET:

```
/MSASSIGN MSNAME BOSTON TO LINK 3
```

Response ET:

```
DFS058I MSASSIGN COMMAND COMPLETED
```

Explanation: The MSNAME BOSTON and its associated SYSID (19) have been reassigned to link 3. Link 3 can be used to associate all its SYSIDs with a main storage-to-main storage connection as opposed to a channel-to-channel or a bisynchronous connection.

### *Example 3 for /MSASSIGN command*

Entry ET:

```
/MSASSIGN SYSID 19 TO LINK 2
```

Response ET:

```
DFS058I MSASSIGN COMMAND COMPLETED
```

Explanation: SYSID 19 might have been originally assigned to link 1 in the remote system and known by the MSNAME of CHICAGO. In this example, SYSID 19 (CHICAGO) is reassigned to link 2. This assignment can be made when both link 1 and link 2 are connected to the same two IMS systems and link 1 is inoperable.

*Example 4 for /MSASSIGN command*

Entry ET:

```
/MSASSIGN TRAN XYZ TO MSNAME LNKY
```

Response ET:

```
DFS058I MSASSIGN COMMAND COMPLETED
```

Explanation: Local and remote SYSIDs of transaction XYZ are assigned to the link associated with MSNAME LNKY.

**Related reference:**

-  [UPDATE TRAN command \(Commands\)](#)
-  [UPDATE MSNAME command \(Commands\)](#)

---

## Chapter 43. /MSVERIFY command

The /MSVERIFY command verifies local transactions and local LTERMs for corresponding remote transactions and remote LTERMs.

The /MSVERIFY command also verifies the consistency of the attributes of corresponding transactions. For instance, Fast Path exclusive transactions cause an error message to be returned because the transactions can only be processed as local transactions. Defined paths that are usable between the two systems can also be verified with this command.

Subsections:

- “Environment”
- “Syntax”
- “Keywords”
- “Usage notes” on page 974
- “Examples” on page 974

### Environment

The following table lists the environments (DB/DC, DBCTL, and DCCTL) in which you can use the commands and keywords.

*Table 360. Valid environments for the /MSVERIFY command and keywords*

Command / Keywords	DB/DC	DBCTL	DCCTL
/MSVERIFY	X		X
MSNAME	X		X
SYSID	X		X

### Syntax

►► `/MSVERIFY` `MSNAME` *msname* `SYSID` *sysid#* ◀◀  
└───┬───┘ └───┬───┘  
`/MSV` `SYSID` *sysid#*

### Keywords

The following keywords are valid for the /MSVERIFY command:

#### **MSNAME**

Specifies the logical link path in a multiple systems configuration.

#### **SYSID**

Specifies the remote system identification of a system in a multiple systems configuration.

## Usage notes

The IMS Multiple Systems Verification utility is run offline following system definition. The /MSVERIFY command is used during online execution when verification of two systems is desired, such as after an /MSASSIGN command has been entered.

**Restrictions:** The /MSVERIFY command has the following restrictions:

- Only one remote system can be specified for each /MSVERIFY command.
- The /MSVERIFY command does not apply to Intersystem Communication (ISC) or directed routing.

## Examples

The following are examples of the /MSVERIFY command:

### *Example 1 for /MSVERIFY command*

Entry ET:

```
/MSVERIFY MSNAME ABC
```

Response ET:

```
DFS2234I MSVERIFY COMMAND IN PROGRESS FOR LOCAL  
SYSIDS 001,002,003,004 dddddd
```

Explanation: Assume SYSID numbers 001 through 004 are defined as local in the input system. dddddd is the time stamp.

Response ET:

```
DFS2243 TA11A IS NOT DEFINED AS TRANSACTION  
IN BOTH SYSTEMS.
```

Explanation: TA11A is defined as a remote transaction in one system but is not defined as a transaction in the partner system.

Response ET:

```
DFS2237I MSVERIFY COMPLETED FOR LOCAL SYSID 001  
ddddd
```

Explanation: Verification of local SYSID 001 is completed. The response message is repeated for local SYSIDs 002, 003, and 004 if all verify. dddddd is the time stamp.

Response ET:

```
DFS2235I SYSID 004 IS DEFINED AS LOCAL IN BOTH  
SYSTEMS. dddddd
```

Explanation: SYSID 004 has no corresponding remote SYSID. dddddd is the time stamp.

Remote SYSIDs are processed next.

Response ET:

```
DFS2236I MSVERIFY COMMAND IN PROGRESS FOR REMOTE  
SYSIDS 005,006,007,008 dddddd
```

Response ET:

```
DFS2237I MSVERIFY COMPLETED FOR SYSID 005  
ddddd
```

Explanation: The verification of one remote SYSID is completed. The response is repeated for remote SYSIDs 006, 007, and 008 if all verify. ddddd is the time stamp.

*Example 2 for /MSVERIFY command*

Entry ET:

```
/MSVERIFY SYSID 8
```

Explanation: The input system sends a list of its local SYSIDs to the remote system and then displays the following message on the input terminal.

Response ET:

```
DFS2234I MSVERIFY COMMAND IN PROGRESS FOR LOCAL SYSIDS  
001, 002, 003, 004 ddddd
```

Explanation: Assume SYSID numbers 001 through 004 are defined as local in the input system.

As the input system completes processing of the information returned for each of the input system's local SYSIDs, the following message is displayed on the input terminal:

Response RT:

```
DFS2237I MSVERIFY COMPLETED FOR SYSID 001  
ddddd
```

Explanation: The verification of local SYSID 001 is completed. The response message is repeated for local SYSIDs 002, 003, 004 if all verify. Verification is complete only if a DFS2237I message is received for every SYSID identified in the DFS2234I message. Definition and assignment errors can cause conditions that prevent these messages from being returned. Stopped or unassigned links can also prevent the return of the message. ddddd is the time stamp.

Remote SYSIDs are processed next. The input system sends all its pertinent information related to the remote system's local SYSIDs and displays the following message:

Response ET:

```
DFS2236I MSVERIFY COMMAND IN PROGRESS FOR REMOTE  
SYSIDS 005, 006, 007, and 008 ddddd
```

As the remote system completes processing of the information it received for each of its local SYSIDs, the following message is displayed on the input terminal. ddddd is the time stamp.

Response ET:

```
DFS2237I MSVERIFY COMPLETED FOR SYSID 005  
ddddd
```

Explanation: The verification of one remote SYSID is complete. The last response message is repeated for remote SYSIDs 006, 007, 008 if all verify. ddddd is the time stamp.



---

## 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

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\_.

---

## Programming interface information

This information documents Product-sensitive Programming Interface and Associated Guidance Information provided by IMS.

Product-sensitive Programming Interfaces allow the customer installation to perform tasks such as diagnosing, modifying, monitoring, repairing, tailoring, or tuning of this software product. Use of such interfaces creates dependencies on the detailed design or implementation of the IBM software product. Product-sensitive Programming Interfaces should be used only for these specialized purposes. Because of their dependencies on detailed design and implementation, it is to be expected that programs written to such interfaces may need to be changed in order to run with new product releases or versions, or as a result of service. Product-sensitive Programming Interface and Associated Guidance Information is identified where it occurs, either by an introductory statement to a section or topic, or by a Product-sensitive programming interface label. IBM requires that the preceding statement, and any statement in this information that refers to the preceding statement, be included in any whole or partial copy made of the information described by such a statement.

---

## Trademarks

IBM, the IBM logo, and [ibm.com](http://www.ibm.com)<sup>®</sup> 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 14 library.

<b>Title</b>	<b>Acronym</b>	<b>Order number</b>
<i>IMS Version 14 Application Programming</i>	APG	SC19-4208
<i>IMS Version 14 Application Programming APIs</i>	APR	SC19-4209
<i>IMS Version 14 Commands, Volume 1: IMS Commands A-M</i>	CR1	SC19-4210
<i>IMS Version 14 Commands, Volume 2: IMS Commands N-V</i>	CR2	SC19-4211
<i>IMS Version 14 Commands, Volume 3: IMS Component and z/OS Commands</i>	CR3	SC19-4212
<i>IMS Version 14 Communications and Connections</i>	CCG	SC19-4213
<i>IMS Version 14 Database Administration</i>	DAG	SC19-4214
<i>IMS Version 14 Database Utilities</i>	DUR	SC19-4215
<i>IMS Version 14 Diagnosis</i>	DGR	GC19-4216
<i>IMS Version 14 Exit Routines</i>	ERR	SC19-4217
<i>IMS Version 14 Installation</i>	INS	GC19-4218
<i>IMS Version 14 Licensed Program Specifications</i>	LPS	GC19-4231
<i>IMS Version 14 Messages and Codes, Volume 1: DFS Messages</i>	MC1	GC18-4219
<i>IMS Version 14 Messages and Codes, Volume 2: Non-DFS Messages</i>	MC2	GC18-4220
<i>IMS Version 14 Messages and Codes, Volume 3: IMS Abend Codes</i>	MC3	GC18-4221
<i>IMS Version 14 Messages and Codes, Volume 4: IMS Component Codes</i>	MC4	GC18-4222
<i>IMS Version 14 Operations and Automation</i>	OAG	SC19-4223
<i>IMS Version 14 Release Planning</i>	RPG	GC19-4224
<i>IMS Version 14 System Administration</i>	SAG	SC19-4225
<i>IMS Version 14 System Definition</i>	SDG	GC19-4226
<i>IMS Version 14 System Programming APIs</i>	SPR	SC19-4227
<i>IMS Version 14 System Utilities</i>	SUR	SC19-4228
<i>Program Directory for Information Management System Transaction and Database Servers V14.01.00</i>		GI10-8988
<i>Program Directory for Information Management System Database Value Unit Edition V14.01.00</i>		GI13-4602
<i>Program Directory for Information Management System Transaction Manager Value Unit Edition V14.01.00</i>		GI13-4601



---

# Index

## Special characters

- /ACTIVATE command
  - description 97
  - environments 97
  - examples 98
  - syntax 97
- /ALLOCATE command
  - description 99
  - environments 99
  - syntax 99
- /ASSIGN command
  - description 111
  - dynamic terminals, using in 113
  - environments 101
  - examples 114
  - restriction 112
  - static terminals, using in 112
  - syntax 102
  - VTAM terminals, using in 113
- /BROADCAST command
  - description 121
  - environment 121
  - examples 124
  - keywords 122
  - requirement 123
  - syntax 121
  - usage notes 123
- /CANCEL command
  - description 127
  - environments 127
  - example 127
  - syntax 127
  - usage notes 127
- /CHANGE command
  - APPC keyword
    - description 129
    - environment 129
    - keywords 130
    - syntax 130
    - usage notes 130
  - CCTL keyword
    - description 130
    - environment 130
    - examples 131
    - keywords 131
    - syntax 131
  - CPLOG keyword
    - description 132
    - environment 132
    - keywords 132
    - syntax 132
  - DESC keyword
    - description 132
    - environment 133
    - keywords 133
    - syntax 133
    - usage notes 134
  - description 129
  - DIR MFS keyword
    - description 134
    - environment 135
  - /CHANGE command (*continued*)
    - DIR MFS keyword (*continued*)
      - syntax 135
      - usage notes 135
    - FDR keyword
      - description 135
      - environment 135
      - syntax 136
      - usage notes 136
    - LINK keyword
      - description 136
      - environment 136
      - examples, ASR 139
      - examples, TCP/IP links 138
      - keywords 137
      - syntax 137
    - NODE keyword
      - description 140
      - environment 141
      - examples 143
      - keywords 141
      - syntax 141
    - SUBSYS keyword
      - description 144
      - environment 144
      - examples 145
      - keywords 144
      - syntax 144
    - SURV keyword
      - description 145
      - environment 146
      - examples 147
      - keywords 146
      - syntax 146
      - usage notes 146
    - TRAN keyword
      - description 148
      - environment 148
      - examples 149
      - keywords 149
      - syntax 148
    - UOR keyword
      - description 150
      - environment 150
      - examples 150
      - keywords 150
      - syntax 150
    - USER keyword
      - description 151
      - environment 151
      - keywords 152
      - syntax 151
  - /CHANGE LINK command
    - examples, ASR 139
    - examples, TCP/IP links 138
  - /CHECKPOINT command
    - description 153
    - environment 153
    - examples 157
    - shutdown
      - description 153
  - /CHECKPOINT command (*continued*)
    - simple checkpoint
      - description 154
    - statistics checkpoint
      - description 154
    - usage notes 157
  - /CLSDST command
    - description 161
    - environment 161
    - example 163
    - syntax 161
    - usage notes 163
  - /COMPT command
    - description 165
    - environment 165
    - examples 167
    - parameters 166
    - syntax 166
    - usage notes 167
  - /CQCHKPT command
    - description 169
    - environment 169
    - examples 170
    - syntax 169
    - usage notes 170
  - /CQQUERY command
    - description 171
    - environment 171
    - examples 172
    - keywords 171
    - syntax 171
    - usage notes 172
  - /CQSET command
    - description 175
    - environment 175
    - examples 176
    - syntax 175
    - usage notes 176
  - /DBDUMP command
    - description 339
    - environment 339
    - examples 343
    - keywords 339
    - syntax 339
    - usage notes 341
  - /DBRECOVERY command
    - description 345
    - environment 345
    - examples 351
    - keywords 346
    - syntax 345
    - usage notes 349
  - /DELETE command
    - description 355
    - environment 355
    - keywords 355
    - syntax 355
  - /DEQUEUE command 471
    - description 471
    - environment 471
    - examples 477

/DEQUEUE command *(continued)*  
 keywords 472  
 usage notes 476

/DIAGNOSE command  
 description 479, 480  
 environment 479, 481  
 example 480, 515  
 keywords 483  
 SET keyword 479  
 SNAP keyword 483  
 syntax 479, 481

/DISPLAY command  
 ACT keyword  
 CLASS 546  
 DC 538  
 description 537  
 environments 537  
 examples 546  
 JOBNAME 540  
 OTMA GROUP 539  
 PROGRAM 541  
 REGID 540  
 REGION 538  
 STATUS 539, 541  
 syntax 537  
 TRANSACTION/STEP 541  
 TYPE 540  
 usage notes 538  
 VTAM ACB 539

AFFIN keyword  
 description 554  
 environments 554  
 examples 555  
 output fields 557  
 syntax 554  
 usage notes 555

AOITKN keyword  
 description 556  
 environment 556  
 examples 557  
 syntax 556  
 usage notes 557

APPC keyword  
 description 557  
 environment 557  
 examples 559  
 output fields 558  
 syntax 558

AREA keyword  
 description 560  
 environments 560  
 examples 563  
 output fields 562  
 syntax 560  
 usage notes 561

ASMT keyword  
 description 565  
 environments 566  
 examples 569  
 syntax 566

CCTL keyword  
 description 575  
 environments 576  
 examples 577  
 keywords 576  
 output fields 577  
 syntax 576

/DISPLAY command *(continued)*  
 CONV keyword  
 description 578  
 environments 578  
 examples 581  
 output fields 580  
 syntax 579  
 usage notes 580

CPLOG keyword  
 description 582  
 environments 582  
 syntax 583

CQS keyword  
 description 583  
 environments 583  
 examples 583  
 syntax 583

DB keyword  
 description 583  
 environments 584  
 examples 589  
 output fields 587  
 syntax 584  
 usage notes 586

DB keyword with BKERR  
 description 588

DBD keyword  
 description 594  
 environments 594  
 examples 594  
 syntax 594

DESC keyword  
 description 595  
 environments 595  
 examples 596  
 output fields 595  
 syntax 595

description 521

FDR keyword  
 description 596  
 environments 596  
 examples 597  
 output fields 596  
 syntax 596

FPV keyword  
 description 597  
 environments 597  
 examples 598  
 output fields 597  
 syntax 597

HSB keyword  
 description 599  
 environments 599  
 examples 601  
 output fields 600  
 syntax 600

HSSP keyword  
 description 602  
 environments 602  
 examples 603  
 output fields 602  
 syntax 602

LINE keyword  
 description 603  
 environments 603  
 examples 606  
 keywords 604

/DISPLAY command *(continued)*  
 LINE keyword *(continued)*  
 reserved parameters 604  
 syntax 604

LINK keyword  
 description 607  
 environments 607  
 examples 609  
 keywords 608  
 output fields 608  
 syntax 608  
 usage notes 608

LTERM keyword  
 description 611  
 EMHQ 611  
 environments 611  
 examples 613  
 keywords 611  
 QCNT 611  
 syntax 611  
 usage notes 612

LU keyword  
 environments 615  
 examples 617  
 syntax 615

LUNAME keyword  
 description 614  
 keywords 615  
 output fields 616

MASTER keyword  
 description 620  
 environments 620  
 examples 620  
 syntax 620

MODIFY keyword  
 description 621  
 environments 621  
 examples 629  
 keywords 622  
 output fields 624  
 syntax 622  
 usage notes 624

MSNAME keyword  
 description 635  
 environments 635  
 examples 636  
 keywords 635  
 syntax 635

NODE keyword  
 description 637  
 EMHQ 639  
 environments 637  
 examples 643  
 keywords 639  
 MODE 639  
 QCNT 639  
 syntax 637  
 usage notes 641

OASN SUBSYS keyword  
 description 648  
 environments 648  
 examples 649  
 syntax 648

OLDS keyword  
 description 649  
 environments 650  
 examples 652

/DISPLAY command *(continued)*  
   OLDS keyword *(continued)*  
     keywords 650  
     syntax 650  
     usage notes 650  
   OTMA keyword  
     description 652  
     environments 652  
     examples 655  
     output fields 653  
     syntax 652  
   OVERFLOWQ keyword  
     description 658  
     environments 658  
     examples 660  
     keywords 658  
     output fields 659  
     syntax 658  
     usage notes 658  
   PGM keyword  
     description 661  
     environments 661  
     syntax 662  
     usage notes 662  
   POOL keyword  
     description 663  
     environments 664  
     examples 673  
     syntax 664  
     usage notes 664  
   PSB keyword  
     description 682  
     environments 683  
     examples 683  
     syntax 683  
     usage notes 683  
   PTERM keyword  
     description 684  
     environments 684  
     examples 686  
     syntax 685  
     usage notes 685  
   Q keyword  
     description 686  
     environments 686  
     examples 687  
     keywords 686  
     syntax 686  
   QCNT keyword  
     description 689  
     environments 690  
     examples 692  
     output fields 692  
     syntax 690  
     usage notes 691  
   RECOVERY keyword  
     description 694  
     environments 694  
     examples 697  
     keywords 694  
     output fields 697  
     syntax 694  
     usage notes 695  
   RTCODE keyword  
     description 700  
     environments 700  
     examples 701

/DISPLAY command *(continued)*  
   RTCODE keyword *(continued)*  
     syntax 700  
   SHUTDOWN STATUS keyword  
     description 701  
     environments 701  
     examples 704  
     syntax 702  
     usage notes 702  
   STATUS keyword  
     description 707  
     environments 707  
     examples 710  
     syntax 707  
     usage notes 708  
   STRUCTURE keyword  
     description 714  
     environments 714  
     examples 715  
     syntax 714  
     usage notes 714  
   SUBSYS keyword  
     description 716  
     environments 716  
     examples 718  
     syntax 716  
     usage notes 716  
   SYSID TRANSACTION keyword  
     environments 719  
     examples 719  
     syntax 719  
   TIMEOVER keyword  
     description 720  
     environments 720  
     examples 720  
     keywords 720  
     syntax 720  
     usage notes 720  
   TMEMBER keyword  
     description 721  
     environments 721  
     examples 721  
     output fields 724  
     syntax 721  
     usage notes 721  
   TMEMBER QCNT keyword  
     description 722  
   TMEMBER TPIPE keyword  
     description 722  
   TMEMBER TPIPE OUTPUT keyword  
     description 723  
   TMEMBER TPIPE QCNT keyword  
     description 722  
   TMEMBER TPIPE SYNC keyword  
     description 723  
   TRACE keyword  
     description 734  
     environments 734  
     examples 737  
     keywords 735  
     output fields 736  
     syntax 735  
     usage notes 736  
   TRACKING STATUS keyword  
     description 744  
     environments 744  
     examples 751

/DISPLAY command *(continued)*  
   TRACKING STATUS keyword  
     *(continued)*  
     output fields 744  
     syntax 744  
   TRAN keyword  
     description 752  
     environments 753  
     examples 755  
     keywords 753  
     output fields 754  
     QCNT 753  
     syntax 753  
     usage notes 754  
   TRANSACTION keyword  
     description 719  
   UOR keyword  
     description 757, 758  
     environments 757  
     examples 759  
     output fields 758  
     syntax 757  
   USER keyword  
     description 760  
     environments 760  
     examples 764  
     keywords 761  
     syntax 761  
     usage notes 763  
 /DISPLAY MODIFY command  
   library status information 624  
 /END command  
   description 767  
   environment 767  
   example 768  
   keywords 767  
   syntax 767  
   usage notes 768  
 /ERESTART command  
   description 769  
   environment 769  
   example 769  
   keywords 769  
   syntax 769  
   usage notes 769  
 /EXCLUSIVE command  
   description 781  
   environment 781  
   example 782  
   keywords 781  
   syntax 781  
   usage notes 782  
 /EXIT command  
   description 783  
   environment 783  
   examples 785  
   keywords 783  
   syntax 783  
   usage notes 784  
 /FORMAT command  
   description 819  
   environment 819  
   example 820  
   keywords 819  
   syntax 819  
   usage notes 820

- /HOLD command
  - description 821
  - environment 821
  - example 821
  - syntax 821
  - usage notes 821
- /IDLE command
  - description 823
  - environment 823
  - examples 826
  - keywords 824
  - syntax 823
- /LOCK command
  - description 949
  - environment 949
  - examples 952
  - keywords 950
  - syntax 949
  - usage notes 951
- /LOG command
  - description 955
  - environment 955
  - example 955
  - keywords 955
  - syntax 955
- /LOOPTEST command
  - description 957
  - environment 957
  - example 958
  - syntax 957
- /MODIFY command
  - description 959
  - environment 959
  - examples 964
  - keywords 960
  - syntax 959
  - usage notes 963
- /MONITOR command 967
  - description 967
  - environment 967
  - example 967
  - keywords 967
- /MSASSIGN command
  - description 969
  - environment 969
  - examples 971
  - keywords 970
  - syntax 969
  - usage notes 970
- /MSVERIFY command
  - description 973
  - environment 973
  - examples 974
  - keywords 973
  - syntax 973
  - usage notes 974

## Numerics

- 3270 Information Display System
  - display screen 2
  - MFS bypass 34
  - multisegment command input
    - description 6
    - example 6
    - printer 2

- 3270 terminal
  - FORMAT command 820
- 3275 terminal
  - COMPONENT keyword 104
- 3770 terminal
  - COMPONENT keyword 104

## A

- ABDUMP keyword
  - CHECKPOINT command 154
- ABORT keyword
  - MODIFY Command 960
- accessibility
  - features ix
  - keyboard shortcuts ix
- ACT keyword
  - /DISPLAY command
    - CLASS 546
    - DC 538
    - JOBNAME 540
    - OTMA GROUP 539
    - PROGRAM 541
    - REGID 540
    - REGION 538
    - STATUS 539, 541
    - TRANSACTION/STEP 541
    - TYPE 540
    - VTAM ACB 539
  - DISPLAY command
    - description 537
- ACTIVATE command
  - description 97
  - environments 97
  - examples 98
  - LINK keyword 97
  - NODE keyword 97
  - syntax 97
- ACTIVE keyword
  - BROADCAST command 122
- active system
  - status 537
- AFFIN keyword
  - DISPLAY command
    - description 554
- ALL parameter
  - KEYWD macro statement 18
  - specifying 18
- ALLOCATE command
  - description 99
  - environments 99
  - LUNAME keyword 99
  - MODE keyword 99
  - syntax 99
  - TPNAME keyword 100
- ALTER option
  - INITIATE OLREORG command 936
- AO (automated operator) application
  - CMD call 32
  - commands supported 32
  - ICMD call 34
  - using CMD 32
  - using ICMD 34
- AOI (automated operator interface) 32
- AOITKN keyword
  - DISPLAY command
    - description 556
- AOITOKEN keyword
  - DEQUEUE command 472
- APPC keyword
  - DISPLAY command
    - description 557
- AREA keyword
  - DBRECOVERY command 346
  - DISPLAY command
    - description 560
    - example 563
- ASMT keyword
  - DISPLAY command
    - description 565
    - example 569
- ASSIGN command
  - CLASS keyword 103
  - COMPONENT keyword 103
  - CPRI keyword 104
  - description 111
  - dynamic terminals, using in 113
  - environments 101
  - equivalent IMS type-2
    - commands 114
  - examples 114
  - ICOMPONENT keyword 104
  - IMSpIex, using in 112
  - INPUT keyword 104
  - LINE keyword 104
  - LMCT keyword 104
  - LPRI keyword 104
  - LTERM keyword
    - data considerations 107
    - description 105
    - master terminal 105
    - physical terminal 105
    - PRIMARY parameter 106
  - master terminal 112
  - NODE keyword 108
  - NOSAVE keyword 108
  - NPRI keyword
    - BMP transactions 108
  - OUTPUT keyword
    - description 108
  - PARLIM keyword
    - description 108
  - PLMCT keyword
    - description 109
  - PTERM keyword 110
  - REGION keyword 110
  - SAVE keyword 110
  - SEGNO keyword
    - description 110
  - SEGSIZE keyword
    - description 110
  - static terminals, using in 112
  - syntax 102
  - TRAN keyword 110
  - USER keyword 110
  - VTAM terminals, using in 113
  - VTAMPPOOL keyword 111
- assignment changes
  - losing when control blocks are
    - deleted 108
- attributes
  - DISPLAY command 523

AUTOLOGON keyword  
  DISPLAY USER command  
  examples 765  
automated operator interface (AOI) 32

## B

BACKUP keyword  
  ERESTART command 769  
BALGRP keyword  
  DISPLAY command 686  
Base Primitive Environment (BPE)  
  commands in CSL 25  
Batch SPOC  
  format 10  
BKERR keyword  
  DISPLAY command  
  description 588  
BMP (batch message processing) program  
  no backout of updates 769  
BMP transactions  
  ASSIGN command  
  NPRI keyword 108  
  LPRI keyword 104  
BPE (Base Primitive Environment)  
  commands in CSL 25  
BROADCAST command  
  ACTIVE keyword 122  
  CHECKPOINT command 157  
  description 121  
  environment 121  
  examples 124  
  keywords 122  
  LINE keyword 122  
  LTERM keyword  
  ALL parameter 123  
  description 122  
  MASTER parameter 123  
  MSNAME keyword 123  
  NODE keyword 123  
  PTERM keyword 123  
  syntax 121  
  SYSID keyword 123  
  usage notes 123  
  USER keyword 123  
BUILDQ keyword  
  ERESTART command 769

## C

CANCEL command  
  description 127  
  environments 127  
  example 127  
  syntax 127  
  usage notes 127  
catalog, IMS  
  IMPORT DEFN  
  SOURCE(CATALOG) 827  
CBT (control block table) pool  
  display abbreviations 667  
  DISPLAY POOL command 665  
  storage pool 665  
CCTL (coordinator controller)  
  status 575

CCTL keyword  
  CHANGE command  
  ABORT 131  
  cctlname 131  
  COMMIT 131  
  prtkn 131  
  DISPLAY command  
  description 575  
  examples 577  
CHANGE command  
  APPC keyword  
  description 129  
  environment 129  
  keywords 130  
  syntax 130  
  usage notes 130  
CCTL keyword  
  description 130  
  environment 130  
  examples 131  
  keywords 131  
  syntax 131  
CPLOG keyword  
  description 132  
  environment 132  
  keywords 132  
  syntax 132  
DESC keyword  
  description 132  
  environment 133  
  keywords 133  
  syntax 133  
  usage notes 134  
  description 129  
DIR MFS keyword  
  description 134  
  environment 135  
  syntax 135  
  usage notes 135  
FDR keyword  
  description 135  
  environment 135  
  syntax 136  
  usage notes 136  
INTERVAL value 147  
LINK keyword  
  description 136  
  environment 136  
  examples, ASR 139  
  examples, TCP/IP links 138  
  keywords 137  
  syntax 137  
NODE keyword  
  description 140  
  environment 141  
  examples 143  
  keywords 141  
  syntax 141  
OUTBND keyword 130  
SUBSYS keyword  
  description 144  
  environment 144  
  examples 145  
  keywords 144  
  syntax 144  
SURV keyword  
  description 145

CHANGE command (*continued*)  
  SURV keyword (*continued*)  
  environment 146  
  examples 147  
  keywords 146  
  syntax 146  
  usage notes 146  
TIMEOUT keyword 130  
TIMEOUT value 147  
TRAN keyword  
  description 148  
  environment 148  
  examples 149  
  keywords 149  
  syntax 148  
UOR keyword  
  description 150  
  environment 150  
  examples 150  
  keywords 150  
  syntax 150  
USER keyword  
  AUTOLOGON 152  
  description 151  
  environment 151  
  ID 152  
  keywords 152  
  LOGOND 152  
  MODE 152  
  NODENAME 152  
  NOSAVE 152  
  SAVE 152  
  syntax 151  
CHANGE LINK command  
  examples, ASR 139  
  examples, TCP/IP links 138  
CHECKPOINT command  
  ABDUMP keyword 154  
  DBCTL (database control) 155  
  description 153  
  DUMPQ keyword 154  
  environment 153  
  examples 157  
  FREEZE keyword 154  
  LEAVEGR keyword 155  
  LEAVEPLEX keyword 156  
  NOCQSSHUT keyword 156  
  notifying with the BROADCAST  
  command 157  
  PURGE keyword 154  
  QUIESCE keyword 156  
  shutdown  
  description 153  
  simple checkpoint  
  description 154  
  SNAPQ keyword 157  
  statistics checkpoint  
  description 154  
  STATISTICS keyword 157  
  system messages 157  
  usage notes 157  
CHECKPOINT keyword  
  ERESTART command 769  
CLASS keyword  
  /DISPLAY command  
  ACT 546  
  ASSIGN command 103

CLASS keyword (*continued*)  
 DISPLAY command 686

CLSDST command  
 description 161  
 environment 161  
 example 163  
 FORCE keyword 161  
 ISC node 161  
 NODE keyword 162  
 syntax 161  
 usage notes 163  
 USER keyword 162  
 VTAM terminal 163

CMD call 32

CMDAUTH keyword  
 ERESTART command 769

CMDAUTHE keyword  
 ERESTART command 769

COLDBASE keyword  
 ERESTART command 769

COLDCOMM keyword  
 ERESTART command 769

COLDSYS keyword  
 ERESTART command 769

COMM macro  
 time stamp 12

command  
 ACTIVATE 97  
 ALLOCATE 99  
 alter IMS resources 29  
 ASSIGN 101  
 automated operator transaction 32  
 behavior in an IMSplex 26  
 BROADCAST 121  
 CANCEL 127  
 CHANGE 129  
 CHECKPOINT 153  
 choosing a master client 26  
 CLSDST 161  
 CMD call 32  
 COMPT 165  
 CQCHKPT 169  
 CQQUERY 171  
 CQSET 175  
 DBCTL 7, 85  
 DBDUMP command 339  
 DBRECOVERY command 345  
 DCCTL  
 list 87  
 DELETE  
 type-2 357  
 DELETE command 355  
 DEQUEUE 471  
 DFSnnn message 12  
 DIAGNOSE 479, 480  
 DISPLAY 521  
 END 767  
 ERESTART 769  
 ETO 95  
 EXCLUSIVE 781  
 EXIT 783  
 format 22  
 DBCTL 17  
 FORMAT command 819  
 HOLD 821  
 ICMD 34  
 IDLE 823

command (*continued*)  
 INITIATE 891  
 INITIATE OLREORG 937  
 input maximum length from z/OS  
 consoles 5  
 keywords 17  
 LOCK 949  
 LOG 955  
 logged to secondary master  
 terminal 31  
 LOOPTEST 957  
 LU 6.2 device  
 allocate conversation 8  
 command response 14  
 security defaults 44  
 maximum length input from z/OS  
 consoles 5  
 mirrored on XRF alternate 37  
 MODIFY 959  
 MONITOR 967  
 MSASSIGN 969  
 MSVERIFY 973  
 multisegment input 6  
 OM security 37  
 OTMA security defaults 44  
 parameters 17  
 processing in a DRD environment 30  
 processing in an IMSplex 26  
 recovered at emergency restart 29  
 recovered at emergency restart 29  
 reserved words 18  
 response 12  
 response with EXCEPT phrase 12  
 status  
 emergency restart 30  
 supported by AOI 32  
 supported by OM API 45  
 supported from LU 6.2 device 44  
 supported from OTMA 44  
 supported on the RSR tracker 40  
 supported on the XRF alternate 38  
 terminal security defaults 27  
 transaction-entered 32  
 type-1  
 entering commands 1  
 type-2 22  
 entering commands 1  
 verb 17  
 z/OS  
 entering commands 1

command characteristics 27  
 command comments 21  
 command format  
 type-2 22  
 command recognition character 17  
 command responses 13  
 command routing  
 IMSplex 26

commands  
 BPE 25  
 equivalent IMS Connect WTOR, z/OS,  
 and type-2 commands 57  
 equivalent IMS type-1 and type-2  
 commands 48  
 issuing  
 TSO SPOC 9  
 issuing to the IMSplex 25

commands (*continued*)  
 not issued directly to OM 25  
 prerequisite knowledge vii  
 comment  
 command 21

COMMIT keyword  
 MODIFY Command 960

COMPONENT keyword  
 3275 terminal 104  
 3770 terminal 104  
 ASSIGN command 103  
 SLU 1 terminal 104  
 SLU 4 terminal 104  
 SLU P terminal 104

COMPT command  
 description 165  
 environment 165  
 examples 167  
 NOTRDY keyword 166  
 parameters 166  
 READY keyword 166  
 ready state 167  
 syntax 166  
 usage notes 167

control block  
 application, cannot load 543  
 IMS, deletion of affinities 155  
 losing assignment changes when  
 deleted 108  
 table (CBT) pools, defining 665  
 terminal  
 specifying the logon  
 descriptor 152

CONV keyword  
 DISPLAY command  
 description 578  
 example 581

CONVERSATION keyword  
 EXIT command 784

conversational processing  
 status 578

CPI communications driven transactions  
 ASSIGN PARLIM command 108  
 ASSIGN PLMCT command 109  
 ASSIGN SEGNO command 110

CPLOG  
 DISPLAY command 582

CPLOG keyword  
 DISPLAY command  
 description 582

CPRI keyword  
 ASSIGN command 104

CQCHKPT command  
 description 169  
 environment 169  
 examples 170  
 SHAREDQ keyword 169  
 STRUCTURE keyword 169  
 syntax 169  
 SYSTEM keyword 170  
 usage notes 170

CQQUERY command  
 description 171  
 environment 171  
 examples 172  
 keywords 171  
 syntax 171

CQQUERY command (*continued*)  
 usage notes 172

CQS (Common Queue Server)  
 DISPLAY command 583

CQS keyword  
 DISPLAY command  
 description 583  
 examples 583

CQSET command  
 description 175  
 environment 175  
 examples 176  
 SHAREDQ keyword 175  
 SHUTDOWN keyword 175  
 STRUCTURE keyword 176  
 syntax 175  
 usage notes 176

CREATE command 281  
 DB keyword  
 completion codes 181  
 description 177  
 environment 177  
 examples 183  
 keywords 178  
 output fields 181  
 reason codes 181  
 return codes 181  
 syntax 178  
 usage notes 180

DBDESC keyword  
 completion codes 189  
 description 185  
 environment 185  
 examples 191  
 keywords 186  
 output fields 188  
 reason codes 189  
 return codes 189  
 syntax 185  
 usage notes 188

IMS Connect  
 DATASTORE 191

IMSCON keyword  
 description 191

IMSCON TYPE(DATASTORE)  
 keyword  
 completion codes 198, 199  
 description 191  
 environment 192  
 equivalent WTOR and z/OS  
 commands 197  
 examples 199  
 keywords 193  
 output fields 197  
 reason codes 198, 199  
 return codes 198, 199  
 syntax 192  
 usage notes 197

IMSCON TYPE(IMPSPLEX) keyword  
 completion codes 204, 206  
 description 202  
 environment 202  
 equivalent WTOR and z/OS  
 commands 203  
 examples 206  
 keywords 202  
 output fields 203

CREATE command (*continued*)  
 IMSCON TYPE(IMPSPLEX) keyword  
 (*continued*)  
 reason codes 204, 206  
 return codes 204, 206  
 syntax 202  
 usage notes 203

IMSCON TYPE(PORT) keyword  
 completion codes 212, 214  
 description 208  
 environment 208  
 equivalent WTOR and z/OS  
 commands 212  
 examples 214  
 keywords 209  
 output fields 212  
 reason codes 212, 214  
 return codes 212, 214  
 syntax 209  
 usage notes 211

LTERM keyword  
 completion codes 220  
 description 218  
 environment 218  
 examples 221  
 keywords 219  
 output fields 219  
 reason codes 220  
 return codes 220  
 usage notes 219

MSLINK  
 description 222

MSNAME keyword  
 completion codes 235  
 description 231  
 environment 231  
 examples 235  
 keywords 232  
 output fields 233  
 reason codes 234  
 return codes 234  
 usage notes 232

MSPLINK keyword  
 completion codes 243  
 description 236  
 environment 237  
 example for CTC link type 246  
 example for MTM link type 247  
 example for TCPIP link type 248  
 example for VTAM link type 249  
 keywords 238  
 output fields 243  
 reason codes 243  
 return codes 243  
 syntax 237, 238  
 usage notes 242

OTMADESC keyword  
 completion codes 257  
 description 250  
 environment 251  
 examples 259  
 keywords 252  
 output fields 256  
 reason codes 257  
 return codes 257  
 syntax 251  
 usage notes 256

CREATE command (*continued*)  
 PGM keyword  
 completion codes 266  
 description 260  
 environment 260  
 examples 269  
 keywords 261  
 output fields 266  
 reason codes 266  
 return codes 266  
 syntax 261  
 usage notes 265

PGMDESC keyword  
 completion codes 277  
 description 271  
 environment 272  
 examples 280  
 keywords 273  
 output fields 277  
 reason codes 277  
 return codes 277  
 syntax 272  
 usage notes 277

RTC keyword  
 completion codes 284  
 description 281  
 environment 281  
 examples 285  
 keywords 282  
 output fields 283  
 reason codes 284  
 return codes 284  
 usage notes 283

RTCDESC keyword  
 completion codes 290  
 description 287  
 environment 287  
 examples 292  
 keywords 288  
 output fields 289  
 reason codes 290  
 return codes 290  
 syntax 287  
 usage notes 289

TRAN keyword  
 completion codes 310  
 description 292  
 environment 292  
 examples 315  
 keywords 295  
 output fields 309  
 reason codes 310  
 return codes 310  
 syntax 293  
 usage notes 309

TRANDESC keyword  
 completion codes 333  
 description 316  
 environment 316  
 examples 337  
 keywords 318  
 output fields 332  
 reason codes 333  
 return codes 333  
 syntax 317  
 usage notes 332

CREATE MSLINK command  
 completion codes 229  
 description 222  
 environment 223  
 examples 230  
 keywords 224  
 output fields 227  
 reason codes 228  
 return codes 228  
 syntax 223  
 usage notes 227

## D

data parameter  
 FORMAT command 820  
 database  
 DBRC authorization  
 MODIFY command 964  
 deleting  
 MODIFY command 964  
 Fast Path  
 MODIFY command 964  
 status 583  
 database control 85  
 DATABASE keyword  
 DBDUMP command 339  
 databases  
 HALDB  
 INITIATE OLREORG  
 command 936  
 reorganizing  
 INITIATE OLREORG  
 command 936  
 DATAGRP keyword  
 DATAGRP command 346  
 DB keyword 358  
 DBRECOVERY command 346  
 DISPLAY command  
 description 583  
 examples 589  
 DISPLAY command with BKERR  
 keyword 588  
 LOCK command 950  
 DBCTL (database control)  
 CHECKPOINT command 155  
 command  
 examples 7  
 format 7  
 list 85  
 DBDUMP command 340  
 DBRECOVERY command 347  
 LOCK DB command 950  
 DBD keyword  
 DISPLAY command  
 description 594  
 Fast Path 594  
 DBDESC keyword 366  
 DBDUMP command  
 DATABASE keyword 339  
 DBCTL 340  
 description 339  
 environment 339  
 equivalent IMS type-2  
 commands 343  
 examples 343  
 GLOBAL keyword 340

DBDUMP command (*continued*)  
 keywords 339  
 LOCAL keyword 341  
 MSDB 340  
 NOFEOV keyword 341  
 NOPFA keyword 341  
 syntax 339  
 usage notes 341  
 DBRC (database recovery control)  
 DBRECOVERY command 347  
 DBRECOVERY command  
 AREA keyword 346  
 DATAGRP keyword 346  
 DB keyword 346  
 DBCTL 347  
 DBRC 347  
 DEDB 349  
 description 345  
 environment 345  
 equivalent IMS type-2  
 commands 351  
 examples 351  
 GLOBAL keyword 347  
 keywords 346  
 LOCAL keyword 348  
 MSDB 346  
 NOFEOV keyword 348  
 NOPFA keyword 348  
 syntax 345  
 usage notes 349  
 DC keyword  
 /DISPLAY command  
 ACT 538  
 IDLE command 825  
 DCCTL (data communication control)  
 commands  
 list 87  
 DEADQ status  
 DEQUEUE USER command 476  
 DEDB (data entry database)  
 DBRECOVERY command 349  
 DISPLAY command 583  
 ERERESTART command 769  
 LOCK DB command 950  
 DEFN keyword 371  
 DELETE command 358, 366, 371, 457  
 DB keyword  
 completion codes 359  
 description 357, 358  
 environment 357  
 examples 363  
 keywords 358  
 output fields 359  
 reason codes 359  
 return codes 359  
 DBDESC keyword  
 completion codes 367  
 description 365  
 environment 365  
 examples 369  
 keywords 366  
 output fields 367  
 reason codes 367  
 return codes 367  
 usage notes 366  
 DEFN keyword  
 completion codes 379

DELETE command (*continued*)  
 DEFN keyword (*continued*)  
 description 371, 376  
 environment 371  
 examples 382  
 keywords 372  
 output fields 378  
 reason codes 379  
 return codes 379  
 description 355  
 environment 355  
 IMSCON keyword  
 description 390  
 IMSCON TYPE(DATASTORE)  
 keyword  
 completion codes 392  
 description 390  
 environment 390  
 equivalent WTOR and z/OS  
 commands 393  
 example 393  
 keywords 391  
 output fields 391  
 reason codes 392  
 return codes 392  
 syntax 390  
 usage notes 391  
 IMSCON TYPE(IMSPLEX) keyword  
 completion codes 397  
 description 395  
 environment 395  
 equivalent WTOR and z/OS  
 commands 398  
 example 398  
 keywords 396  
 output fields 396  
 reason codes 397  
 return codes 397  
 syntax 395  
 usage notes 396  
 IMSCON TYPE(PORT) keyword  
 completion codes 402  
 description 400  
 environment 400  
 equivalent WTOR and z/OS  
 commands 403  
 example 403  
 keywords 401  
 output fields 401  
 reason codes 402  
 return codes 402  
 syntax 400  
 usage notes 401  
 keywords 355  
 LE keyword  
 completion codes 407  
 description 405  
 environment 405  
 examples 408  
 keywords 406  
 output fields 407  
 reason codes 407  
 return codes 407  
 syntax 405  
 usage notes 406  
 LTERM keyword  
 completion codes 415

- DELETE command *(continued)*
  - LTERM keyword *(continued)*
    - description 412
    - environment 412
    - examples 415
    - keywords 412
    - output fields 413
    - reason codes 414
    - return codes 414
    - syntax 412
    - usage notes 413
  - MSLINK keyword
    - completion codes 419
    - description 416
    - environment 417
    - examples 420
    - keywords 417
    - output fields 418
    - reason codes 419
    - return codes 419
    - syntax 417
    - usage notes 417
  - MSNAME keyword
    - completion codes 424
    - description 421
    - environment 421
    - examples 426
    - keywords 422
    - output fields 423
    - reason codes 423
    - return codes 423
    - syntax 421
    - usage notes 422
  - MSPLINK keyword
    - completion codes 430
    - description 427
    - environment 427
    - examples 431
    - keywords 428
    - output fields 429
    - reason codes 429
    - return codes 429
    - syntax 428
    - usage notes 428
  - OTMADESC keyword
    - completion codes 434
    - description 432
    - environment 433
    - examples 435
    - keywords 433
    - output fields 434
    - reason codes 434
    - return codes 434
    - syntax 433
    - usage notes 433
  - PGM keyword
    - completion codes 438
    - description 436
    - environment 436
    - examples 440
    - keywords 436
    - output fields 437
    - reason codes 438
    - return codes 438
    - syntax 436
    - usage notes 437
- DELETE command *(continued)*
  - PGMDESC keyword
    - completion codes 443
    - description 441
    - environment 442
    - examples 445
    - keywords 442
    - output fields 443
    - reason codes 443
    - return codes 443
    - syntax 442
    - usage notes 442
  - RTC keyword
    - completion codes 449
    - description 447
    - environments 447
    - examples 450
    - keywords 447
    - output fields 448
    - reason codes 449
    - return codes 449
    - syntax 447
    - usage notes 448
  - RTCDESC keyword
    - completion codes 454
    - description 452
    - environment 452
    - examples 455
    - keywords 452
    - output fields 453
    - reason codes 454
    - return codes 454
    - syntax 452
    - usage notes 453
  - syntax 355
  - TRAN keyword
    - completion codes 459
    - description 457
    - environment 457
    - examples 461
    - keywords 457
    - output fields 458
    - reason codes 459
    - return codes 459
    - usage notes 458
  - TRANDESC keyword
    - completion codes 465
    - description 463
    - environment 463
    - examples 467
    - keywords 464
    - output fields 465
    - reason codes 465
    - return codes 465
    - syntax 464
    - usage notes 464
  - type-2
    - environment 405
    - LE keyword 406
    - requirement 407
    - syntax 405
    - usage 357
  - DELETE LE command
    - examples 408
  - DEQUEUE command 471
    - AOITOKEN keyword 472
    - description 471
- DEQUEUE command *(continued)*
  - environment 471
  - examples 477
  - keywords 472
  - LINE keyword 472
  - LTERM keyword 472
  - LUNAME keyword 473
  - MSNAME keyword 473
  - network-qualified LU name 473
  - NODE keyword 473
  - PTERM keyword 472
  - PURGE keyword 474
  - PURGE1 keyword 474
  - SUSPEND keyword 475
  - TMEM keyword
    - OTMA 475
  - TPIPE keyword
    - OTMA 475
  - TPNAME keyword 473
  - TRAN keyword 476
  - usage nose 476
  - USER keyword 476
  - DEQUEUE USER command
    - DEADQ status 476
  - DESC keyword
    - CHANGE command
      - LUNAME 133
      - MODE 133
      - OUTBND 133
      - SIDE 134
      - SYNCLEVEL 134
      - TPNAME 134
      - TYPE 134
    - DELETE command 355
    - DISPLAY command
      - description 595
      - example 596
  - DFSCPIC
    - DISPLAY TRANSACTION
      - command 752
  - DIAGNOSE command
    - description 479, 480
    - environment 479, 481
    - example 480, 515
    - keywords 483
    - syntax 479, 481
  - DISPLAY command
    - ACT keyword
      - CLASS 546
      - DC 538
      - description 537
      - environments 537
      - examples 546
      - JOBNAME 540
      - OTMA GROUP 539
      - PROGRAM 541
      - REGID 540
      - REGION 538
      - STATUS 539, 541
      - syntax 537
      - TRANSACTION/STEP 541
      - TYPE 540
      - usage notes 538
      - VTAM ACB 539
    - AFFIN keyword
      - description 554
      - environments 554

DISPLAY command *(continued)*  
   AFFIN keyword *(continued)*  
     equivalent IMS type-2  
       commands 555  
     examples 555  
     output fields 557  
     syntax 554  
     usage notes 555  
   AOITKN keyword  
     description 556  
     environment 556  
     examples 557  
     syntax 556  
     usage notes 557  
   APPC keyword  
     description 557  
     environment 557  
     examples 559  
     output fields 558  
     syntax 558  
   AREA keyword  
     description 560  
     environments 560  
     equivalent IMS type-2  
       commands 562  
     examples 563  
     output fields 562  
     syntax 560  
     usage notes 561  
   ASMT keyword  
     description 565  
     environments 566  
     equivalent IMS type-2  
       commands 568  
     examples 569  
     syntax 566  
   attributes  
     list 523  
   BALGRP keyword 686  
   BKERR keyword  
     description 588  
   CCTL keyword  
     description 575  
     environments 576  
     examples 577  
     keywords 576  
     output fields 577  
     syntax 576  
   CLASS keyword 686  
   CONV keyword  
     description 578  
     environments 578  
     equivalent IMS type-2  
       commands 580  
     example 581  
     examples 581  
     output fields 580  
     syntax 579  
     usage notes 580  
   conversation processing 579  
   CPLOG keyword  
     description 582  
     environments 582  
     syntax 583  
   CQS keyword  
     description 583  
     environments 583

DISPLAY command *(continued)*  
   CQS keyword *(continued)*  
     examples 583  
     syntax 583  
   DB keyword  
     description 583  
     environments 584  
     equivalent IMS type-2  
       commands 587  
     examples 589  
     output fields 587  
     syntax 584  
     usage notes 586  
   DBD keyword  
     description 594  
     environments 594  
     equivalent IMS type-2  
       commands 594  
     examples 594  
     syntax 594  
   DESC keyword  
     description 595  
     environments 595  
     examples 596  
     output fields 595  
     syntax 595  
   description 521  
   FDR keyword 596  
     environments 596  
     examples 597  
     output fields 596  
     syntax 596  
   FPV keyword 597  
     environments 597  
     examples 598  
     output fields 597  
     syntax 597  
   HSB keyword  
     description 599  
     environments 599  
     examples 601  
     output fields 600  
     syntax 600  
   HSSP keyword  
     description 602  
     environments 602  
     examples 603  
     output fields 602  
     syntax 602  
   INTERVAL keyword 600  
   LINE keyword  
     description 603  
     environments 603  
     examples 606  
     keywords 604  
     reserved parameters 604  
     syntax 604  
   LINK keyword  
     description 607  
     environments 607  
     examples 609  
     keywords 608  
     output fields 608  
     syntax 608  
     usage notes 608  
   LTERM keyword  
     description 611

DISPLAY command *(continued)*  
   LTERM keyword *(continued)*  
     EMHQ 611  
     environments 611  
     equivalent IMS type-2  
       commands 612  
     examples 613  
     keywords 611  
     QCNT 611  
     syntax 611  
     usage notes 612  
   LU keyword  
     environments 615  
     examples 617  
     syntax 615  
   LUNAME keyword  
     description 614  
   MASTER keyword  
     description 620  
     environments 620  
     equivalent IMS type-2  
       commands 620  
     examples 620  
     syntax 620  
   MODIFY keyword  
     description 621  
     environments 621  
     equivalent IMS type-2  
       commands 624  
     examples 629  
     keywords 622  
     output fields 624  
     syntax 622  
     usage notes 624  
   MSNAME keyword  
     description 635  
     environments 635  
     examples 636  
     keywords 635  
     syntax 635  
   network-qualified LU name 557  
   NODE keyword  
     description 637  
     EMHQ 639  
     environments 637  
     equivalent IMS type-2  
       commands 642  
     examples 643  
     keywords 639  
     MODE 639  
     QCNT 639  
     syntax 637  
     usage notes 641  
   OASN keyword  
     description 648  
     examples 649  
   OASN SUBSYS keyword  
     environments 648  
     examples 649  
     syntax 648  
   OLDS keyword  
     description 649  
     environments 650  
     examples 652  
     keywords 650  
     syntax 650  
     usage notes 650

DISPLAY command *(continued)*

- OTMA keyword
  - description 652
  - environments 652
  - examples 655
  - output fields 653
  - syntax 652
- OVERFLOWQ keyword
  - description 658
  - environments 658
  - examples 660
  - keywords 658
  - output fields 659
  - syntax 658
  - usage notes 658
- parameters 691
- PGM keyword
  - description 661
  - environments 661
  - equivalent IMS type-2
    - commands 662
  - examples 662
  - syntax 662
  - usage notes 662
- POOL keyword
  - abbreviations used 667
  - CBT pool 665
  - description 663
  - environments 664
  - equivalent IMS type-2
    - commands 667
  - examples 673
  - syntax 664
  - usage notes 664
- PRIORITY keyword 686
- PSB keyword
  - description 682
  - environments 683
  - equivalent IMS type-2
    - commands 683
  - examples 683
  - syntax 683
  - usage notes 683
- PTERM keyword
  - description 684
  - environments 684
  - examples 686
  - syntax 685
  - usage notes 685
- Q keyword
  - description 686
  - environments 686
  - examples 687
  - keywords 686
  - syntax 686
- QCNT keyword
  - description 689
  - environments 690
  - equivalent IMS type-2
    - commands 691
  - examples 692
  - output fields 692
  - syntax 690
  - usage notes 691
- RECOVERY keyword
  - description 694
  - environments 694

DISPLAY command *(continued)*

- RECOVERY keyword *(continued)*
  - examples 697
  - keywords 694
  - output fields 697
  - syntax 694
  - usage notes 695
- RTC keyword
  - equivalent IMS type-2
    - commands 701
- RTCODE keyword
  - description 700
  - environments 700
  - examples 701
  - syntax 700
- SHUTDOWN keyword
  - description 701
  - examples 704
- SHUTDOWN STATUS keyword
  - environments 701
  - examples 704
  - syntax 702
  - usage notes 702
- status
  - list 523
- STATUS keyword
  - description 701, 707
  - environments 707
  - equivalent IMS type-2
    - commands 709
  - examples 710
  - syntax 707
  - usage notes 708
- STRUCTURE keyword
  - description 714
  - environments 714
  - examples 715
  - syntax 714
  - usage notes 714
- SUBSYS keyword
  - description 648, 716
  - environments 716
  - examples 649, 718
  - syntax 716
  - usage notes 716
- SYSID keyword
  - description 719
  - examples 719
- SYSID TRANSACTION keyword
  - environments 719
  - examples 719
  - syntax 719
- TIMEOVER keyword
  - description 720
  - environments 720
  - examples 720
  - keywords 720
  - syntax 720
  - usage notes 720
- TMEMBER keyword
  - description 721
  - environments 721
  - examples 721
  - output fields 721, 724
  - syntax 721
  - usage notes 721

DISPLAY command *(continued)*

- TMEMBER QCNT keyword
  - description 722
  - example 729
- TMEMBER TPIPE keyword
  - description 722
  - output fields 721
- TMEMBER TPIPE OUTPUT keyword
  - description 723
  - example 733
- TMEMBER TPIPE QCNT keyword
  - description 722
  - example 729
- TMEMBER TPIPE SYNC keyword
  - description 723
- TRACE keyword
  - description 734
  - environments 734
  - equivalent IMS type-2
    - commands 736
  - examples 737
  - keywords 735
  - output fields 736
  - syntax 735
  - usage notes 736
- TRACKING STATUS keyword 744
  - environments 744
  - examples 751
  - output fields 744
  - syntax 744
- TRAN keyword
  - description 752
  - environments 753
  - equivalent IMS type-2
    - commands 754
  - examples 755
  - keywords 753
  - output fields 754
  - QCNT 753
  - syntax 753
  - usage notes 754
- TRANSACTION keyword
  - with Q keyword 686
  - with SYSID keyword 719
- UOR keyword
  - description 757, 758
  - environments 757
  - examples 759
  - output fields 758
  - syntax 757
- USER keyword
  - description 760
  - environments 760
  - equivalent IMS type-2
    - commands 764
  - examples 764
  - keywords 761
  - syntax 761
  - usage notes 763
- DISPLAY NODE command
  - EMHQ keyword
    - examples 647
  - QCNT keyword
    - examples 647
  - display screen format
    - display area 4
    - master terminal 2

display screen format (*continued*)  
 master terminal in XRF 2  
 message area 4  
 system status 2  
 user input area 4  
 warning message area 4  
 XRF system status 2  
 DISPLAY USER command  
 AUTOLOGON keyword  
 examples 765  
 EMHQ keyword  
 examples 765  
 QCNT keyword  
 examples 765  
 DL/I  
 database error queues  
 status 588  
 DL/I databases  
 inflight changes  
 backing out 769  
 DUMPQ keyword  
 CHECKPOINT command 154

## E

E-MCS 5  
 EMHQ keyword  
 /DISPLAY command  
 LTERM 611  
 NODE 639  
 DISPLAY NODE command  
 examples 647  
 DISPLAY USER command  
 examples 765  
 END command  
 description 767  
 environment 767  
 example 768  
 keywords 767  
 LINE keyword 767  
 NODE keyword 767  
 PTERM keyword 767  
 syntax 767  
 usage notes 768  
 USER keyword 768  
 Entry Terminal (ET) 15  
 environments, valid for keywords  
 /ASSIGN command 101  
 EOM  
 multisegment command input 6  
 EOS  
 multisegment command input 6  
 ERESTART command  
 BACKUP keyword 769  
 BUILDQ keyword 769  
 CHECKPOINT keyword 769  
 CMDAUTH keyword 769  
 CMDAUTHE keyword 769  
 COLDBASE keyword 769  
 COLDCOMM keyword 769  
 COLDSYS keyword 769  
 DEDB 769  
 description 769  
 environment 769  
 example 769  
 FORMAT keyword 769  
 keywords 769

ERESTART command (*continued*)  
 MSDB 769  
 MULTISIGN keyword 769  
 NOBMP keyword 769  
 NOCMDAUTH keyword 769  
 NOCMDAUTHE keyword 769  
 NOTRANAUTH keyword 769  
 NOUSER keyword 769  
 OVERRIDE keyword 769  
 restart  
 security definition 769  
 SNGLSIGN keyword 769  
 syntax 769  
 TRANAUTH keyword 769  
 usage notes 769  
 USER keyword 769  
 ET (Entry Terminal)  
 response 15  
 ETO  
 commands 95  
 terminal  
 ERESTART CMDAUTH  
 command 769  
 ERESTART CMDAUTHE  
 command 769  
 examples  
 /DISPLAY ACT command 546  
 /DISPLAY AFFIN command 555  
 /DISPLAY AOITKN command 557  
 /DISPLAY APPC command 559  
 /DISPLAY FDR command 597  
 /DISPLAY FPV command 598  
 /DISPLAY TRACKING STATUS  
 command 751  
 DELETE LE command 408  
 TSO SPOC 15  
 EXCEPT phrase  
 response to command 12  
 EXCLUSIVE command  
 description 781  
 environment 781  
 example 782  
 keywords 781  
 LINE keyword 781  
 NODE keyword 781  
 PTERM keyword 781  
 syntax 781  
 usage notes 782  
 USER keyword 781  
 EXIT command  
 CONVERSATION keyword 784  
 description 783  
 environment 783  
 examples 785  
 keywords 783  
 LINE keyword 784  
 LUNAME keyword 784  
 NODE keyword 784  
 PTERM keyword 784  
 syntax 783  
 TMEMBER TPIPE keyword 784  
 usage notes 784  
 USER keyword 784  
 EXPORT command  
 completion codes 800  
 description 787  
 environment 787

EXPORT command (*continued*)  
 examples 804  
 keywords 789  
 output fields 799  
 reason codes 800  
 return codes 800  
 syntax 787  
 usage notes 796  
 Extended Recovery Facility 2

## F

Fast Path  
 area  
 status 560  
 database  
 MODIFY command 964  
 database buffer pool  
 display 672  
 DEDBs  
 recovering 769  
 DISPLAY DBD command 594  
 ERESTART COLDBASE  
 command 769  
 exclusive transactions  
 DISPLAY command 752  
 LOCK TRAN command 951  
 FDR keyword  
 DISPLAY command 596  
 FORCE keyword  
 CLSDST command 161  
 format  
 lost during conversation  
 restoring 820  
 FORMAT command  
 data parameter 820  
 description 819  
 environment 819  
 example 820  
 keywords 819  
 LTERM keyword 819  
 modname parameter 819  
 syntax 819  
 usage notes 820  
 FORMAT keyword  
 ERESTART command 769  
 FPV keyword  
 DISPLAY command 597  
 FREEZE keyword  
 CHECKPOINT command 154

## G

generic resources  
 affinity  
 displaying 554  
 GLOBAL keyword  
 DBDUMP command 340  
 DBRECOVERY command 347

## H

HALDB  
 example  
 /DISPLAY DB command 592  
 /DISPLAY DB Command 591

HALDB (*continued*)  
     reorganizing  
         INITIATE OLREORG  
             command 936  
 HOLD command  
     description 821  
     environment 821  
     example 821  
     syntax 821  
     usage notes 821  
 HSB keyword  
     DISPLAY command  
         description 599  
         example 601  
 HSSP (high-speed sequential processing)  
     status 602  
 HSSP keyword  
     DISPLAY command  
         description 602  
         example 603

**I**

IAM command  
     password 21  
 ICMD call 34  
 ICOMPONENT keyword  
     ASSIGN command 104  
 IDLE command  
     DC keyword 825  
     description 823  
     environment 823  
     examples 826  
     keywords 824  
     LINE keyword 824  
     LINK keyword 824  
     NODE keyword 825  
     NOSHUT keyword  
         description 824  
         restarting 824  
     syntax 823  
 IMPORT command  
     ACBs 827  
     catalog 827  
         completion codes 836  
         environment 828  
         examples 844  
         keywords 828  
         output fields 832  
         reason codes 834  
         return codes 834  
         syntax 828  
         usage notes 830  
     completion codes 834  
     DBDs 827  
     DRD 848  
     examples 874  
     IMS catalog 827  
     IMSRSC repository 848  
     MODBLKS resources 848  
     PSBs 827  
     RDDS  
         completion codes 861  
         environment 848  
         keywords 850  
         output fields 857  
         reason codes 859  
     IMPORT command (*continued*)  
         RDDS (*continued*)  
             return codes 859  
             syntax 849  
             usage notes 855  
         REPO  
             completion codes 861  
             environment 848  
             keywords 850  
             output fields 857  
             reason codes 859  
             return codes 859  
             syntax 849  
             usage notes 855  
         repository 848  
         resource definition data set 848  
         SOURCE(CATALOG) 827  
             completion codes 836  
             environment 828  
             examples 844  
             keywords 828  
             output fields 832  
             reason codes 834  
             return codes 834  
             syntax 828  
             usage notes 830  
         SOURCE(RDDS) 848  
             completion codes 861  
             environment 848  
             keywords 850  
             output fields 857  
             reason codes 859  
             return codes 859  
             syntax 849  
             usage notes 855  
         SOURCE(REPO) 848  
             completion codes 861  
             environment 848  
             keywords 850  
             output fields 857  
             reason codes 859  
             return codes 859  
             syntax 849  
             usage notes 855  
     IMS catalog  
         IMPORT DEFN  
             SOURCE(CATALOG) 827  
     IMS Command  
         entering  
             MCS 5  
     IMS commands  
         entering 2, 9  
         Batch SPOC 10  
         LU 6.2 device 8  
         Master Terminal Format 2  
         TSO SPOC 9  
     OM API 9  
     supported by OM API 45  
         long form 45  
         short form 45  
     type-2 22  
     IMS Connect  
         CREATE IMSCON  
             TYPE(DATASTORE) 191  
         DATASTORE definition  
             CREATE IMSCON command 191

IMS-supported terminals  
     RSR environment 43  
 IMSplex  
     command processing 26  
     command routing 26  
     issuing commands to 25  
 IMSRSC repository  
     IMPORT DEFN SOURCE(REPO |  
         RDDS) command 848  
 INITIATE command 891  
     OLC keyword  
         completion codes 905  
         description 891  
         environment 891  
         examples 922  
         keywords 892  
         output fields 903  
         reason codes 905  
         return codes 905  
         syntax 892  
         usage notes 900  
     OLREORG keyword  
         completion codes 941  
         description 936  
         environment 936  
         examples 944  
         keywords 936  
         output fields 940  
         reason codes 941  
         return codes 941  
         syntax 936  
         usage notes 939  
     INITIATE OLC command  
         completion codes 911  
         error handling 933  
         keywords  
             OPTION 892  
             PHASE 894  
             TYPE 898  
         return and reason codes 905  
     INITIATE OLC PHASE(COMMIT)  
         command 900  
     INITIATE OLC PHASE(PREPARE)  
         command 900  
     INITIATE OLREORG command  
         completion codes 941  
         keywords  
             NAME 937  
             OPTION 937  
             SET 937  
         output fields 940  
         responses 940  
         return and reason codes 941  
     INPUT keyword  
         ASSIGN command 104  
     input, maximum length from z/OS  
         consoles 5  
     INTERVAL keyword  
         DISPLAY command 600  
     INTERVAL value  
         CHANGE command 147  
     ISC (Intersystem Communication)  
         END NODE command 768  
         node  
             ASSIGN OUTPUT command 108  
             CLSDST command 161

ISC (Intersystem Communication)  
(continued)  
user  
    CLSDST NODE USER  
        command 162

## J

JOBNAME keyword  
    /DISPLAY command  
        ACT 540

## K

keyboard shortcuts ix  
KEYWD macro statement  
    ALL parameter 18  
    generic parameters 19  
keyword  
    guidelines for entering 17  
    reserved words 91  
    supported on the RSR tracker 40  
    synonyms 63  
keywords  
    NAME 937  
    OPTION 937  
    SET 937  
    supported by OM API 45

## L

LEAVEGR keyword  
    CHECKPOINT command 155  
LEAVEPLEX keyword  
    CHECKPOINT command 156  
legal notices  
    notices 977  
    trademarks 977, 979  
library status information 624  
line  
    inclusive parameter 18  
LINE keyword  
    ASSIGN command 104  
    BROADCAST command 122  
    DEQUEUE command 472  
    DISPLAY command  
        description 603  
        example 606  
    END command 767  
    EXCLUSIVE command 781  
    EXIT command 784  
    IDLE command 824  
    LOOPTEST command 957  
    MONITOR command 967  
link  
    inclusive parameter 18  
    status 607  
LINK keyword  
    ACTIVATE command 97  
    CHANGE command  
        ASR 137  
        COLDSESS 137  
        FORCESS 137  
        MODE 138  
        SYNCS 137

LINK keyword (continued)  
    DISPLAY command  
        description 607  
        examples 609  
    IDLE command 824  
    MSASSIGN command 970  
LMCT keyword  
    ASSIGN command 104  
LOCAL keyword  
    DBDUMP command 341  
    DBRECOVERY command 348  
    MSASSIGN command 970  
LOCK command  
    DB keyword 950  
    description 949  
    environment 949  
    equivalent IMS type-2  
        commands 952  
    examples 952  
    keywords 950, 955  
    LTERM keyword 950  
    NODE keyword 950  
    password 21  
    PGM keyword 950  
    PTERM keyword 950  
    syntax 949  
    TRAN keyword 950  
    usage notes 951  
LOG command  
    description 955  
    environment 955  
    example 955  
    syntax 955  
logical link path  
    queue counts 635  
LOOPTEST command  
    description 957  
    environment 957  
    example 958  
    LINE keyword 957  
    PTERM keyword 957  
    syntax 957  
looptest mode  
    DISPLAY LINE PTERM  
        command 957  
LPRI keyword  
    ASSIGN command 104  
    BMP transactions 104  
lterm  
    status 611  
LTERM control block  
    preventing deletion of 110  
LTERM keyword  
    /DISPLAY command  
        EMHQ 611  
        QCNT 611  
    ASSIGN command  
        data considerations 107  
        description 105  
        master terminal 105  
        physical terminal 105  
        PRIMARY parameter 106  
    BROADCAST command  
        ALL parameter 123  
        description 122  
    DELETE type-2 command 406  
    DEQUEUE command 472

LTERM keyword (continued)  
    DISPLAY command  
        description 611  
        examples 613  
    FORMAT command 819  
    LOCK command 950  
LU 6.2 application program  
    DEALLOCATE 785  
    descriptor 595  
    ending conversations 785  
    EXIT command 785  
    network-qualified LU name  
        status 595  
    status  
        related activity 557  
        specific device 614  
LU 6.2 device  
    allocating a synchronous  
        conversation 8  
    command  
        examples 8  
        format 8  
        response 14  
        security defaults 44  
        supported 44  
    multisegment command input 6  
LU keyword  
    DISPLAY command  
        examples 617  
LUNAME keyword  
    ALLOCATE command 99  
    DEQUEUE command 473  
    DISPLAY command  
        description 614  
    EXIT command 784  
    network-qualified LU name 473

## M

MASTER parameter  
    BROADCAST command 123  
    DISPLAY command  
        description 620  
        examples 620  
master terminal  
    ASSIGN command 112  
    display screen 2  
    display screen format 2  
    entering commands 2  
    messages to 14  
    status 620  
    XRF display screen format 2  
Master Terminal (MT) 15  
maximum length of command input from  
z/OS consoles 5  
MCS (multiple console support)  
    console 5  
message  
    destination  
        CHANGE DESCRIPTOR  
            command 134  
Message Requirer program 769  
messages  
    sending  
        Master Terminal (MT) 14  
    restrictions 14

MFS (Message Format Service)  
 bypass 34  
 FORMAT command 820  
 password 21  
 MODE keyword  
 /DISPLAY command  
 NODE 639  
 ALLOCATE command 99  
 MODIFY command  
 ABORT keyword 960  
 COMMIT keyword 960  
 description 959  
 environment 959  
 examples 964  
 keywords 960  
 PREPARE keyword 961  
 RACF keyword 963  
 syntax 959  
 usage notes 963  
 MODIFY keyword  
 DISPLAY command  
 description 621  
 examples 629  
 modname parameter  
 FORMAT command 819  
 MONITOR command 967  
 description 967  
 environment 967  
 example 967  
 keywords 967  
 LINE keyword 967  
 PTERM keyword 967  
 MRQ (Message Requeuer program)  
 ERESTART BUILDQ command 769  
 MSASSIGN command  
 description 969  
 environment 969  
 equivalent IMS type-2  
 commands 971  
 examples 971  
 keywords 970  
 LINK keyword 970  
 LOCAL keyword 970  
 MSNAME keyword 970  
 MSPLINK keyword 970  
 syntax 969  
 SYSID keyword 970  
 TRAN keyword 970  
 usage notes 970  
 MSC (Multiple Systems Coupling)  
 MSASSIGN command 970  
 MSVERIFY command 974  
 MSDB (main storage database)  
 DBDUMP command 340  
 DBRECOVERY command 346  
 ERESTART command 769  
 LOCK DB command 950  
 reloading 769  
 MSNAME keyword  
 BROADCAST command 123  
 DEQUEUE command 473  
 DISPLAY command  
 description 635  
 examples 636  
 MSASSIGN command 970  
 MSVERIFY command 973

MSPLINK keyword  
 DISPLAY command 568  
 MSASSIGN command 970  
 MSVERIFY command  
 description 973  
 environment 973  
 examples 974  
 keywords 973  
 MSNAME keyword 973  
 syntax 973  
 SYSID keyword 973  
 usage notes 974  
 MT (Master Terminal)  
 response 15  
 multiple console support 5  
 multisegment command input  
 3270 Information Display System 6  
 display screen 4  
 EOM 6  
 EOS 6  
 examples 6  
 LU 6.2 device 6  
 MULTISIGN keyword  
 ERESTART command 769

## N

network-qualified LU name 9  
 NOBMP keyword  
 ERESTART command 769  
 NOCMDAUTH keyword  
 ERESTART command 769  
 NOCMDAUTH keyword  
 ERESTART command 769  
 NOCQSSHUT keyword  
 CHECKPOINT command 156  
 node  
 status 637  
 NODE keyword  
 /DISPLAY command  
 EMHQ keyword 639  
 MODE 639  
 QCNT 639  
 ACTIVATE command 97  
 ASSIGN command 108  
 BROADCAST command 123  
 CHANGE command  
 ASR 141  
 COLDSESS 142  
 FORCSESS 142  
 MODE 142  
 SYNCSESS 142  
 USER 142  
 CLSDST command 162  
 DEQUEUE command 473  
 DISPLAY command  
 description 637  
 examples 643  
 END command 767  
 EXCLUSIVE command 781  
 EXIT command 784  
 IDLE command 825  
 LOCK command 950  
 NOFEOV keyword  
 DBDUMP command 341  
 DBRECOVERY command 348

non-VTAM lines  
 terminate input  
 IDLE LINE NOSHUT  
 command 824  
 NOPFA keyword  
 DBDUMP command 341  
 DBRECOVERY command 348  
 NOSAVE keyword  
 ASSIGN command 108  
 NOSHUT keyword  
 IDLE command  
 description 824  
 restarting 824  
 NOTRANAUTH keyword  
 ERESTART command 769  
 NOTRDY keyword  
 COMPT command 166  
 NOUSER keyword  
 ERESTART command 769  
 NPRI keyword  
 ASSIGN command  
 BMP transactions 108  
 null keywords 20  
 null word  
 reserved words 91

## O

OASN (origin application schedule numbers)  
 status 648  
 OASN keyword  
 DISPLAY command  
 description 648  
 examples 649  
 OLDS keyword  
 DISPLAY command  
 description 649  
 examples 652  
 OM (Operations Manager) 26  
 OM API  
 command example 16  
 command responses 13  
 input 16  
 issuing  
 commands 9  
 output 16  
 supported command long form 45  
 supported command short form 45  
 supported commands 45  
 supported keywords 45  
 OM security 37  
 Open Transaction Manager Access (OTMA)  
 status 652  
 Operations Manager (OM)  
 choosing a master client 26  
 supported command long form 45  
 supported command short form 45  
 supported commands 45  
 OSAM (overflow sequential access method)  
 buffer pool  
 /DISPLAY POOL command 668  
 enhanced buffer pool  
 /DISPLAY POOL command 670

OTMA  
 security default commands 44  
 supported commands 44  
 OTMA GROUP keyword  
 /DISPLAY command  
 ACT 539  
 OTMA keyword  
 DISPLAY command  
 description 652  
 examples 655  
 OTMA keyword 655  
 OUTBND keyword  
 CHANGE command 130  
 OUTPUT keyword  
 ASSIGN command  
 description 108  
 outstanding reply numbers  
 z/OS system console 5  
 OVERFLOWQ keyword  
 DISPLAY command  
 description 658  
 examples 660  
 OVERRIDE keyword  
 ERESTART command 769

## P

parameters  
 ALL 18  
 description 17  
 generic 18, 24  
 group 20  
 inclusive 18  
 PARLIM keyword  
 ASSIGN command  
 description 108  
 password  
 command 21  
 definition 21  
 RACF 21  
 terminals 21  
 performance  
 checkpointing the queue  
 structure 169  
 displaying global queue  
 information 691  
 records 157  
 serial search 19, 24  
 when using /CQCHKPT 169  
 when using /DISPLAY ALL 560  
 when using /DISPLAY QCNT 691  
 PGM keyword  
 DELETE type-2 command 406  
 DISPLAY command  
 description 661  
 examples 662  
 LOCK command 950  
 PHIDAM  
 example  
 /DISPLAY DB command 591, 592  
 PLMCT keyword  
 ASSIGN command  
 description 109  
 POOL keyword  
 DISPLAY command  
 description 663  
 examples 673

PREPARE keyword  
 MODIFY command 961  
 preset mode  
 3270 Information Display System  
 MFS bypass 34  
 ASSIGN command  
 LINE keyword 104  
 LTERM keyword 105  
 NODE keyword 108  
 PRIORITY keyword  
 DISPLAY command 686  
 PROGRAM keyword  
 /DISPLAY command  
 ACT 541  
 PSB keyword  
 DISPLAY command  
 description 682  
 examples 683  
 pterm  
 inclusive parameter 18  
 PTERM keyword  
 ASSIGN command 110  
 BROADCAST command 123  
 DEQUEUE command 472  
 DISPLAY command  
 description 684  
 examples 686  
 EXCLUSIVE command 781  
 EXIT command 784  
 LOCK command 950  
 LOOPTEST command 957  
 MONITOR command 967  
 PURGE keyword  
 CHECKPOINT command 154  
 DEQUEUE command 474  
 PURGE1 keyword  
 DEQUEUE command 474

## Q

Q keyword  
 DISPLAY command  
 description 686  
 examples 687  
 QCNT keyword  
 /DISPLAY command  
 LTERM 611  
 NODE 639  
 TRANSACTION 753  
 DISPLAY command  
 description 689  
 examples 692  
 DISPLAY NODE command  
 examples 647  
 DISPLAY USER command  
 examples 765  
 QUIESCE keyword  
 CHECKPOINT command 156

## R

RACF (Resource Access Control facility)  
 MODIFY RACF command 963  
 password 21  
 RACF keyword  
 MODIFY command 963

RCF= parameter  
 overriding with ERESTART COLDSYS  
 command 769  
 RDDS (resource definition data set)  
 IMPORT DEFN SOURCE(REPO |  
 RDDS) command 848  
 READY keyword  
 COMPT command 166  
 ready state  
 COMPT command 167  
 RECOVERY keyword  
 DISPLAY command  
 description 694  
 REGID keyword  
 /DISPLAY command  
 ACT 540  
 REGION keyword  
 /DISPLAY command  
 ACT 538  
 ASSIGN command 110  
 reorganization  
 INITIATE OLREORG command 936  
 reserved words  
 description 18  
 keywords 17, 91  
 null keywords 20  
 null words 91  
 parameters 91  
 RESET command  
 MFS bypass 34  
 resource definition data set (RDDS)  
 IMPORT DEFN SOURCE(REPO |  
 RDDS) command 848  
 Resource Manager (RM)  
 command behavior 26  
 resources  
 assignment  
 status 565  
 changed status 621  
 Response Terminal (RT) 15  
 restart  
 security definition 769  
 restriction  
 /ASSIGN command 112  
 command status in emergency  
 restart 30  
 shutdown checkpoint in  
 shared-queues environment 154  
 RM (Resource Manager) 26  
 RSR (Remote Site Recovery)  
 commands supported on the  
 tracker 40  
 IMS-supported terminals 43  
 RT (Response Terminal)  
 response 15  
 RTC keyword 281  
 RTCODE keyword  
 DISPLAY command  
 description 700  
 examples 701

## S

SAVE keyword  
 ASSIGN command 110  
 secondary master terminal  
 commands logged 31

security  
  definition  
    at restart 769  
    OM 37  
  terminal default 27

SEGNO keyword  
  ASSIGN command  
  description 110

SEGSIZE keyword  
  ASSIGN command  
  description 110

serial search  
  performance 19, 24

SET keyword  
  /DIAGNOSE command 479

SGN= parameter  
  overriding with ERESTART COLDSYS  
  command 769

SHAREDQ keyword  
  CQCHKPT command 169  
  CQSET command 175

shutdown  
  CHECKPOINT command  
  FREEZE keyword 155  
  restriction in shared-queues  
  environment 154  
  status 701

SHUTDOWN keyword  
  CQSET command 175  
  DISPLAY command  
  description 701  
  examples 704

single point of control (SPOC)  
  sending commands with 25

SLU 1 terminal  
  COMPONENT keyword 104

SLU 4 terminal  
  COMPONENT keyword 104

SLU P terminal  
  COMPONENT keyword 104

SMCOPY command  
  commands logged to secondary  
  master terminal 31

SNAP keyword  
  /DIAGNOSE command 483  
  ADDRESS 484  
  AREA 484  
  BLOCK 485  
  DB 489  
  LINE 491  
  LINK 493  
  LTERM 496  
  MODULE 499  
  MSNAME 499  
  NODE 502  
  OPTION 504  
  PGM 506  
  REGION 508  
  STRUCTURE 512  
  TRAN 512  
  USER 512

SNAPQ keyword  
  CHECKPOINT command 157

SNGLSIGN keyword  
  ERESTART command 769

SPOC (single point of control)  
  sending commands with 25

static  
  node  
    inclusive parameter 18  
  terminal  
    ERESTART CMDAUTH  
    command 769

STATISTICS keyword  
  CHECKPOINT command 157  
  CQQUERY command 171

status  
  active system 537  
  CCTL 575  
  communication line 603  
  conversational processing 578  
  database 583  
  database access 594  
  DEDB 583  
  DISPLAY command  
  list 523  
  DL/I database error queues 588  
  external subsystem 716  
  Fast Path  
  area 560  
  database 708  
  routing code 700  
  HSSP activities 602  
  IMS resources 707  
  logical link 607  
  logical link path 635  
  logical terminal 611  
  LU 6.2 application program  
  descriptor 595  
  for specific devices 614  
  related activities 557  
  master terminal 620  
  message queues 686  
  node 637  
  OASN 648  
  OTMA 652  
  OTMA target member 721  
  physical terminal 603, 684  
  program 661  
  PSB 682  
  resource assignment 565  
  resources  
  added, changed, or deleted 621  
  shutdown 701  
  storage pools 663  
  system logging 649  
  trace 734  
  transaction 719, 752  
  unit of recovery 757  
  user structures 760  
  VTAM response 720  
  XRF system 599

STATUS keyword  
  /DISPLAY command  
  ACT 539, 541  
  DISPLAY command  
  description 701, 707  
  examples 710

status, library information 624

storage pool  
  status 663

STRUCTURE keyword  
  CQCHKPT command 169  
  CQQUERY command 171

STRUCTURE keyword (*continued*)  
  CQSET command 176  
  DISPLAY command  
  description 714  
  examples 715

SUBSYS keyword  
  DISPLAY command  
  description 716  
  examples 649, 718

SURVEILLANCE keyword  
  CHANGE command  
  ALL 146  
  INTERVAL 146  
  LNK 146  
  LOG 146  
  RDS 146  
  TIMEOUT 146

SUSPEND keyword  
  DEQUEUE command 475  
  syntax 281, 358, 366, 371, 457, 471, 967  
  syntax diagram  
  how to read viii

SYSID keyword  
  BROADCAST command 123  
  DISPLAY command  
  description 719  
  examples 719  
  MSASSIGN command 970  
  MSVERIFY command 973  
  system initialization parameters,  
  displayed 769

SYSTEM keyword  
  CQCHKPT command 170  
  system logging  
  status 649  
  system messages  
  CHECKPOINT command 157

## T

terminal command  
  examples 15

terminal security  
  defaults for LU 6.2 devices 44  
  defaults for OTMA 44

terminal security defaults 27

time stamp  
  defined by COMM macro 12  
  in command responses 12

TIMEOUT keyword  
  CHANGE command 130

TIMEOUT value  
  CHANGE command 147

TIMEOVER keyword  
  DISPLAY command  
  description 720  
  examples 720

TMEM keyword  
  DEQUEUE command 475

TMEMBER keyword  
  DISPLAY command  
  description 721  
  examples 721

TMEMBER TPIPE keyword  
  EXIT command 784

TPIPE keyword  
  DEQUEUE command 475

TPNAME keyword  
 ALLOCATE command 100  
 DEQUEUE command 473

TRACE keyword  
 DISPLAY command  
 description 734  
 examples 737

TRACKING STATUS keyword  
 DISPLAY command 744

trademarks 977, 979

TRAN keyword 457  
 /DISPLAY command  
 QCNT 753  
 ASSIGN command 110  
 DELETE type-2 command 406  
 DEQUEUE command 476  
 DISPLAY command  
 description 752  
 examples 755  
 LOCK command 950  
 MSASSIGN command 970

TRANAUTH keyword  
 ERESTART command 769

TRANSACTION keyword  
 DISPLAY command  
 with Q keyword 686  
 with SYSID keyword 719

TRANSACTION/STEP keyword  
 /DISPLAY command  
 ACT 541

TRN= parameter  
 overriding with ERESTART COLDSYS  
 command 769

truncated commands 12

TSO SPOC  
 command example 15  
 entering 10  
 format 9  
 interface 9  
 issuing  
 commands 9  
 type-1 commands 9  
 type-2 commands 9  
 screen example 9

TYPE keyword  
 /DISPLAY command  
 ACT 540

type-1  
 command  
 entering 1  
 examples 15  
 format 16  
 responses  
 OM API 13

type-2  
 command  
 entering 1  
 command format 22  
 resource types 22  
 responses  
 OM API 13

type-2 commands  
 CREATE 177, 827  
 DELETE 357  
 EXPORT 787  
 IMPORT DEFN  
 SOURCE(CATALOG) 827

type-2 commands (*continued*)  
 IMPORT DEFN SOURCE(REPO |  
 RDDS) 848  
 INITIATE 891

## U

UNLOCK command  
 password 21

UOR keyword  
 CHANGE command  
 ABORT 150  
 COMMIT 150  
 PRTRKN 150  
 DISPLAY command  
 description 757  
 examples 759

user control block  
 preventing deletion of 110

USER keyword  
 ASSIGN command 110  
 BROADCAST command 123  
 CLSDST command 162  
 DEQUEUE command 476  
 DISPLAY command  
 description 760  
 examples 764  
 END command 768  
 ERESTART command 769  
 EXCLUSIVE command 781  
 EXIT command 784

USERID keyword  
 DELETE type-2 command 406

## V

VSAM (Virtual Storage Access Method)  
 buffer pool  
 /DISPLAY POOL command 669  
 enhanced buffer pool  
 /DISPLAY POOL command 671

VTAM (Virtual Telecommunications  
 Access Method)  
 ACB  
 close 702  
 attached terminals  
 LOOPTEST LINE command 957  
 MSC links  
 checkpoint shutdown 824  
 node  
 close 702  
 EXCLUSIVE NODE  
 command 781  
 IDLE NODE command 825  
 terminal  
 CLSDST command 163

VTAM ACB keyword  
 /DISPLAY command  
 ACT 539

VTAMPOOL keyword  
 ASSIGN command 111

## X

XRF (Extended Recovery Facility)  
 commands mirrored by the  
 alternate 37  
 commands supported on the  
 alternate 38  
 display screen  
 master terminal 2  
 system status 2  
 system-related status  
 information 599

XRF takeover 30

## Z

z/OS  
 command  
 entering 1  
 multiple console support 5  
 system console  
 maximum length of command  
 input 5  
 messages to 14  
 outstanding reply numbers 5





Product Number: 5635-A05  
5655-DSE  
5655-TM3

Printed in USA

SC19-4210-02



Spine information:

IMS Version 14

Commands, Volume 1: IMS Commands A - M

