IMS
15.4.0

Operations and Automation
(2024-08-30 edition)

Note

Before you use this information and the product it supports, read the information in “Notices” on page 293.

2024-08-30 edition.

Contents

About this information

These topics provide guidance information for selecting tools and options for operating an IMS system, for recovering an IMS system, and for automating IMS operations and recovery tasks. The topics also describe how to develop procedures for the master terminal operator and the end user.

This information is available in IBM® Documentation.

Prerequisite knowledge

Before using this information, operators and system programmers need to have some foundation understanding of basic z/OS® and IMS concepts, the IMS environment, and your installation's IMS system. Additionally, system programmers should understand administration of the IMS system and databases, z/OS control programs, and VSAM Access Service Methods.

To learn about z/OS, see z/OS Basic Skills. For more resources, see IBM Z Education and Training.

To learn about IMS, see the IBM Press publication *An Introduction to IMS*, the resources listed for IBM Information Management System, and the variety of options available in IBM Training.

How new and changed information is identified

For most IMS library PDF publications, information that is added or changed after the PDF publication is first published is denoted by a character (revision marker) in the left margin. The *Program Directory* and *Licensed Program Specifications* do not include revision markers.

Revision markers follow these general conventions:

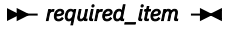
- Only technical changes are marked; style and grammatical changes are not marked.
- If part of an element, such as a paragraph, syntax diagram, list item, task step, or figure is changed, the entire element is marked with revision markers, even though only part of the element might have changed.
- If 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.

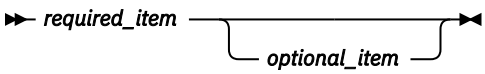
How to read syntax diagrams

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

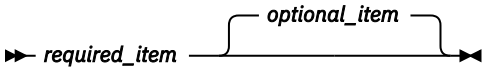
- Read the syntax diagrams from left to right, from top to bottom, following the path of the line. The following conventions are used:
  - The >>--- symbol indicates the beginning of a syntax diagram.
  - The ---> symbol indicates that the syntax diagram is continued on the next line.
  - The >--- symbol 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.

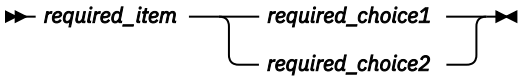


If an optional item appears above the main path, that item has no effect on the execution of the syntax element and is used only for readability.

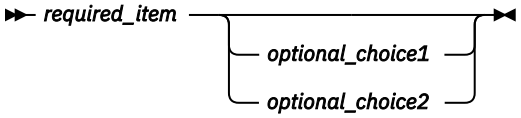


- If you can choose from two or more items, they appear vertically, in a stack.

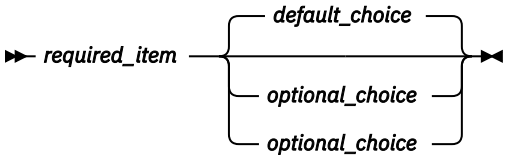
  If you *must* choose one of the items, one item of the stack appears on the main path.



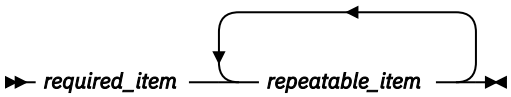
  If choosing one of the items is optional, the entire stack appears below the main path.



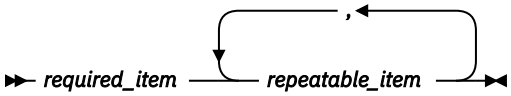
  If one of the items is the default, it appears above the main path, and the remaining choices are shown below.



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

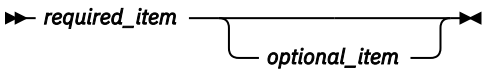


  If the repeat arrow contains a comma, you must separate repeated items with a comma.



  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 15.4

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

Accessibility features

The following list includes the major accessibility features in z/OS products, including IMS 15.4. These features support:

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

Keyboard navigation

You can access IMS 15.4 ISPF panel functions by using a keyboard or keyboard shortcut keys.

For information about navigating the IMS 15.4 ISPF panels using TSO/E or ISPF, refer to the *z/OS TSO/E Primer*, the *z/OS TSO/E User's Guide*, and the *z/OS ISPF User's Guide Volume 1*. These guides describe how to navigate each interface, including the use of keyboard shortcuts or function keys (PF keys). Each guide includes the default settings for the PF keys and explains how to modify their functions.

Related accessibility information

Online documentation for IMS 15.4 is available in IBM Documentation.

IBM and accessibility

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

How to send your comments

About this task

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:

Procedure

- Submit a comment by using the DISQUS commenting feature at the bottom of any IBM Documentation topic.
- Send an email to imspubs@us.ibm.com. Be sure to include the book title.
- Click the **Contact Us** tab at the bottom of any IBM Documentation topic.

What to do next

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. Controlling IMS

You can use the TSO Single Point of Control (TSO SPOC) to issue commands to control IMS and IMS resources.

Controlling IMS with the TSO SPOC application

You can use the TSO SPOC to issue operator commands in an IMSplex. The TSO SPOC application uses an ISPF panel interface and communicates with the IMS Operations Manager (OM). OM then communicates with all of the other address spaces in the IMSplex (for example, IMS) as required for operations.

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.
- Allows users to create, query, update, and delete various IMS resources online.
- Displays the OM audit trail log which records command input, command responses, and selected system messages from across the IMSplex.
- Allows users to define input user exits to modify or reject command parameters, set return and reason codes, and send text messages to the TSO SPOC session.

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.

Important: TSO SPOC applications issue the CSLOMCMD request. You therefore need to be aware of the parameters and return and reason codes of the CSLOMCMD request.

The following screen sample shows the format of the TSO SPOC screen:

```
File   Action   Manage resources   SPOC   View   Options   Help
-------------------------------------------------------------------
PLEX1                              IMS Single Point of Control
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 1. TSO SPOC screen format

You can issue both IMS type-1 commands and type-2 commands by using the TSO SPOC interface. Enter the command next to the command prompt (`Command ==>` in the previous screen sample). 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. If you specify an asterisk (*) in the Route field, the command is routed to all registered command processing clients in the IMSplex. If you specify a percent symbol (%) in the Route field, 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. If no route list is specified, the default routing is to all registered command processing clients in the IMSplex.

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.

Tip: If you want to change the default PF key settings on SPOC ISPF panels, use the **KEYS** or **KEYLIST** ISPF command.

For more information about the TSO SPOC application, see the IMS TSO SPOC online tutorial. To see the IMS TSO SPOC online tutorial, select **Help > Tutorial** in the application.

Related concepts

A single point of control (SPOC) program in CSL (System Administration)

Related reference

CSLOMCMD: command request (System Programming APIs)

ISPF keylist settings

Starting and setting up the TSO SPOC

After you start the TSO SPOC through the IMS Application menu, you must add the IMS distribution libraries to the TSO user's environment.

About this task

To start the TSO SPOC, use the IMS Application menu.

Use one of the following methods to set up the TSO SPOC program:

- Edit the logon procedure to include the IMS distribution libraries or to issue ALLOCATE commands to make the data sets available to the TSO user.

  The **TSOLIB** command of TSO can be used to make the load module data set available to the TSO user. **TSOLIB** is a command that establishes a STEPLIB-like data set without having to modify the logon procedure.

  The data set allocated to ISPTABL is unique to the user. Because of the way ISPF uses tables, the same data set allocated to ISPTABL must also be allocated in the ISPTLIB concatenation and ahead of the IMS.SDFSTLIB data set. The data sets are shown in the following table.

Table 1. Data sets for SPOC setup

Usage	Data set
TSOLIB command or STEPLIB	IMS.SDFSRESL
FILE(ISPPLIB)	IMS.SDFSPLIB
FILE(ISPMLIB)	IMS.SDFSMLIB
FILE(ISPTLIB)	*user*.ISPTLIB IMS.SDFSTLIB
FILE(ISPTABL)	*user*.ISPTLIB
FILE(SYSPROC)	IMS.SDFSEXEC

  After the data sets are added to your TSO environment, you can call the TSO SPOC from other applications through a command interface.

  An example of calling TSO SPOC from a rexx program is:

```
/* rexx */Address ISPEXEC "SELECT CMD(DFSSPOC CMD(qry tran) PLEX(plex1))"
```

Additionally, you can call the TSO SPOC by entering:

- TSO DFSSPOC in any ISPF command line.
- DFSSPOC in the ISPF option 6 command line followed by any of the optional parameters that appear in the DFSSPOC syntax diagram.

- Use the TSO ALTLIB command and ISPF's LIBDEF service. The DFSSPSRT exec provides an example of starting the DFSSPOC program using ALTLIB and LIBDEF.

  You can invoke DFSSPSRT from ISPF option 6 by entering:

```
EXEC 'imslib.SDFSEXEC(DFSSPSRT)' 'HLQ(imslib)'
```

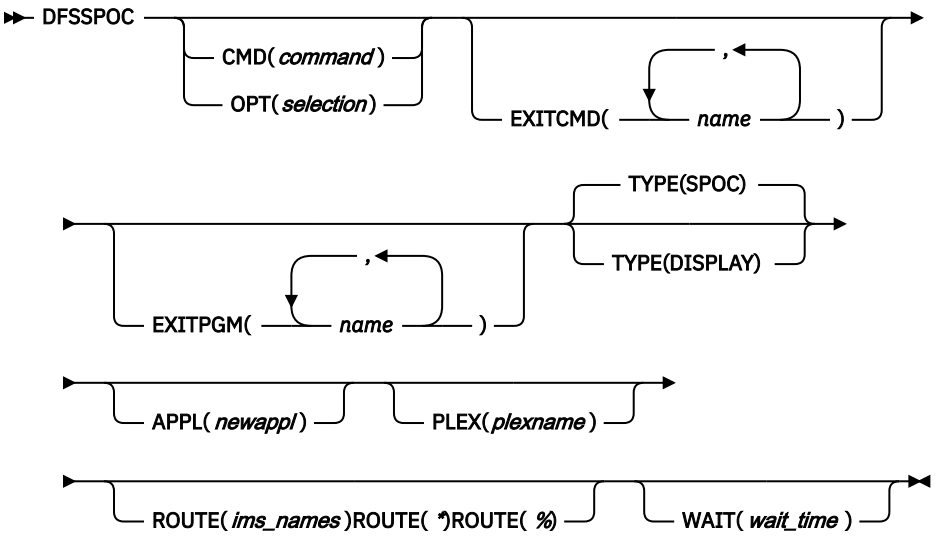
Related concepts

IMS Application Menu (System Administration)

Syntax of the DFSSPOC command

The **DFSSPOC** command supports several parameters. External programs can issue IMS operator commands and display the command response. You can scroll through the data, but no other SPOC interactions are allowed.

The DFSSPOC command accepts the following parameters:



DFSSPOC keywords

DFSSPOC
Specifies the TSO SPOC command name.

CMD(*command*)
Specifies that a command issue immediately. The response displays in the first SPOC panel. Only commands supported by the OM API can be used.

OPT(*selection*)
Jumps to a specific Manage Resource menu without viewing intermediate menus where (*selection*) is a value (1, 2, 3, and so on) for a particular menu.

EXITCMD
Specifies the user exit that TSO SPOC invokes before sending the command to OM. Can be either a single or a list of user exits. Command exits are called with a TSO command processor parameter list. All command exits are called before the program exits. On entry to the routine, register 1 points to the Command Processor Parameter List (CPPL), and is defined by macro IKJCPPL.

EXITPGM
Specifies the user exit that TSO SPOC invokes before sending the command to OM. Can be either a single or a list of user exits. Program exits are called with a z/OS batch program parameter list. Command exits are called before the program exits. On entry to the routine, register 1 points to a standard parameter list. Register 1 points to a full word, which points to a half-word length followed by the parameter string.

TYPE(DISPLAY)
Displays the command response only, with no other SPOC interactions.

TYPE(SPOC)
Indicates that the normal TSO SPOC display be used. This is the default if TYPE is not specified.

APPL(*newappl*)
Indicates a user specified application ID that is used by ISPF to fence off different applications. This is a one to four character value. The first character is alphabetic and the remaining characters are alphanumeric. If no value is specified, a default of 'CSLU' is used.

PLEX(*plexname*)
Specifies the name of the IMSplex to which to issue the command. If no IMSplex name is specified, the user's default IMSplex name is used.

ROUTE(*ims_names*)
Specifies which members of the IMSplex to route the command to. Names must be separated by commas.

ROUTE(*)
Specifies that the command is routed to all registered command processing clients in the IMSplex.

ROUTE(%)
Specifies that 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(*wait_time*)
Specifies, in MM:SS format, how long OM waits for member responses before returning a response. The default wait time is 5 minutes.

Related reference
TSO SPOC user exit routines (Exit Routines)

Starting the TSO SPOC for the first time

When you start the TSO SPOC for the first time (after setup is complete), you must set the user preferences.

About this task

Perform the following tasks in the TSO SPOC application:

Procedure

1. Select **Options** > **Preferences** to display the IMS Single Point of Control Preferences panel.
2. Set the default IMSplex value (this step is required).
3. Optionally, you can specify values for the other preferences or accept the default values. The Preferences panel provides field-level help for each field.
4. Press **Enter**.

Results

You can now enter commands at the TSO SPOC command line.

Type-1 and type-2 command responses from the TSO SPOC

When you issue a type-1 command in the TSO SPOC, the command response is displayed in sequential format. The command response consists of messages prefixed by the member name. Information from each member is grouped together in the command response.

When you issue a type-2 command, the command response is displayed in tabular format, with the data displayed in columns. You can sort the list of messages in the command response by positioning the cursor on a column heading (for example, you can sort by member) and pressing Enter. You can also sort the messages by selecting **View > Sort** and selecting the column heading by which you want to sort.

Restriction: The OM API supports most type-1 commands.

Related reference

IMS command language overview (Commands)

Displaying command status in the TSO SPOC

You can use the TSO SPOC to display command status, reissue commands that were previously issued, or edit and delete commands.

In the command status panel of the TSO SPOC application, you can:

- Display the commands that you have previously issued in the same TSO SPOC session.
- Display the responses of commands that you have previously issued, if the responses are available.
- Reissue commands that you have previously issued and view the command responses.
- Delete commands.
- Edit commands.

To open the command status panel, select **SPOC** > **Command status** in the TSO SPOC application.

```
File   Action   Manage resources   SPOC   View   Options   Help
-------------------------------------------------------------------------------
PLEX1                    IMS Single Point of Control
Command ===> ____________________________________________________________
_____________________________________________________________________________
 ------------------------------- Plex . _____ Route . _______  Wait . ____
Enter '/' to view command response, 'i' to reissue a command, 'd' to delete
a command, and 'e' to edit a command.

Act   Status      Command
___   Complete    DIS STATUS
___   Complete    QRY TRAN NAME(SKS*) SHOW(ALL)
___   Complete    QRY IMSPLEX SHOW(TYPE,STATUS,SUBTYPE)
___               /NRE CHKPT 0 FMT  ALL
___               QRY IMSPLEX SHOW(ALL)
___               QRY TRAN NAME(CDEBTRN3) SHOW(STATUS)
___               START TRAN CDEBTRN3
___               QRY TRAN NAME(CDEBTRN3) SHOW(ALL)
___               DIS   TRAN  CDEBTRN3
___               DIS   TRAN  CDEBTRN3 ALL
___               DIS ACT
```

Figure 2. TSO SPOC Command Status panel

Related concepts

“Reissuing commands in the TSO SPOC” on page 7

When you issue a command, the command is saved in the TSO SPOC. You can reissue commands in several ways.

Command shortcuts in the TSO SPOC

You can use the command shortcuts option of the TSO SPOC if you prefer to use shortened versions of commands or nicknames for commands. Define the shortcuts (the short commands or nicknames) in the TSO SPOC.

Use the ampersand (&) character as the first character of the shortcut if you want to use it as a nickname.

When you issue a short version of a command, the TSO SPOC appends the additional parameters to the short command. When you issue a nickname for a command, the TSO SPOC replaces the nickname with the full version of the command. The following example shows the SPOC Command Shortcuts panel with some user-defined command shortcuts.

```
File   Action   Manage resources   SPOC   View   Options   Help
 ------------------------------------------------------------
                    SPOC Command Shortcuts
 Command ===> ____________________________________________________
_____________________________ _____ _____ _____ _____ _____ _____
 ---------------------- Plex . _____ Route . _________ Wait . ____
 Act   Command         Additional Parameters
 ____  ______________  ___________________________________________
                       ___________________________________________
 ____  &QRY1_________  QRY IMSPLEX SHOW(STATUS)___________________
                       ___________________________________________
 ____  QRY PGM______   SHOW(ALL)_____________________________________
                       ___________________________________________
 ____  QRY TRAN______  SHOW(CLS PSB QCNT STATUS)___________________
                       ___________________________________________
 ____  MON LINE ABC07  PTERM ALL_____________________________________
                       ___________________________________________
 ************************** Bottom of data **********************
```

Figure 3. TSO SPOC Command Shortcuts panel

In this example, the first command shortcut is **&QRY1** (a nickname). When you issue this command shortcut, the TSO SPOC replaces the nickname with the entire command, **QRY IMSPLEX SHOW(STATUS)**. The second command shortcut is **QRY PGM**. When you issue this command shortcut, the TSO SPOC appends the additional parameters (in this example, **SHOW(ALL)**) to the short version of the command when it is issued. The full command **QRY PGM SHOW(ALL)** is issued.

Related concepts

“Reissuing commands in the TSO SPOC” on page 7
When you issue a command, the command is saved in the TSO SPOC. You can reissue commands in several ways.

Entering command shortcuts in the TSO SPOC

You can enter command shortcuts in the TSO SPOC instead of using the full command name. When you issue a short version of a command, the TSO SPOC appends the additional parameters to the short command.

About this task

To use command shortcuts in the TSO SPOC:

Procedure

1. Specify the preference for command shortcuts in the IMS Single Point of Control Preferences panel of the TSO SPOC. Select **Options** > **Preferences** to display this panel.
2. Define your command shortcuts in the SPOC Command Shortcuts panel by selecting **SPOC** > **Command shortcuts**. To define a nickname, ensure that the first character of the command shortcut is an ampersand (&).

Defining groups of IMS systems in the TSO SPOC

You can use the **Route** field in the TSO SPOC application to issue commands to a specific IMS or IMS systems. However, the **Route** field is long enough for only two IMS systems. To work around this restriction, you can define groups of IMS systems in the group definitions panel of the TSO SPOC.

About this task

Procedure

1. Select **Options** > **Set IMS groups** in the TSO SPOC to display the group definitions panel.
2. Specify the name of the group in the **Route** field to route commands to those specific IMS systems.

Reissuing commands in the TSO SPOC

When you issue a command, the command is saved in the TSO SPOC. You can reissue commands in several ways.

- After entering a command, position the cursor on the entered command and press **Enter**. The command is moved into the TSO SPOC command line, and you can edit the command or press **Enter** to issue the command again.
- Use the ISPF retrieve key to display commands that were previously entered. Because the ISPF retrieve key is an ISPF function, all commands entered from other applications are also retrieved.
- Use the ISPF **RETP** command to list previously entered commands. Because the ISPF retrieve key is an ISPF function, all commands entered from other applications are also retrieved. Select the number of the command shown in the list of commands and press Enter.
- Use the TSO SPOC command status panel. Enter "i" in the **Act** column to reissue the command. If you enter "i" next to several commands and press Enter, the commands are reissued in the order in which they are listed.
- Use the TSO SPOC command shortcuts panel. Enter "i" in the **Act** column to reissue the command. If you enter "i" next to several commands and press Enter, the commands are reissued in the order in which they are listed.

Related concepts

“Displaying command status in the TSO SPOC” on page 5
You can use the TSO SPOC to display command status, reissue commands that were previously issued, or edit and delete commands.

“Command shortcuts in the TSO SPOC” on page 6
You can use the command shortcuts option of the TSO SPOC if you prefer to use shortened versions of commands or nicknames for commands. Define the shortcuts (the short commands or nicknames) in the TSO SPOC.

Managing IMS resources by using TSO SPOC

The TSO SPOC application provides a set of resource management panels to help you manage IMS resources. You can use TSO SPOC to query IMS resources and, if dynamic resource definition (DRD) is enabled in your IMS systems, you can also use the TSO SPOC panels to create, delete, export, import, and update IMS resources.

You can access the panels from the Manage Resources menu in the TSO SPOC application or by selecting the **Manage Resources** option directly from the IMS Application Menu.

Querying IMS resource information by using TSO SPOC

You can query the attributes and status of IMS resources as maintained in each IMS system or across the IMSplex. The information returned for IMS resources can include:

- The status of resources as set by commands and maintained locally by IMS systems

- The status of resources as set by commands and maintained globally across the IMSplex by Resource Manager (RM)
- The status of resources as set in the DBRC RECON data set
- The attributes of various IMS resources as set by system definition macros or by DRD commands
- The attributes status and attributes information about an IMSRSC repository

Creating, deleting, exporting, importing, and updating IMS resources by using TSO SPOC

Before you can create, delete, export, import, or update IMS resources through TSO SPOC by using the dynamic resource definition commands, you should be aware of the following points:

- You must enable DRD in the target IMS systems.
- Changes made to IMS resources are not saved across an IMS cold start until they are exported to the resource definition data set (RDDS) or the repository. If a cold start or emergency restart with the COLDSYS parameter occurs before an export is performed, any changes to resource definitions are lost.
- After creating an application program or database by using the **CREATE** command, you must further define the application program or database, as well as its relationships to other resources, by running the appropriate generation utility, such as the Database Description Generation (DBDGEN) utility or the Program Specification Block Generation (PSBGEN) utility. You must perform an ACBLIB or ACBMBR online change to ensure that the created or modified DBD or PSB resources are available to the IMS system.

The IMS resources and resource descriptors that you can create, delete, export, import, and update in DRD-enabled systems include:

- Application programs
- Databases
- Fast Path routing codes
- Transactions
- Change list (delete only)

Related tasks

Enabling dynamic definition for IMS resource groups (System Definition)

Viewing the OM audit trail log by using TSO SPOC

You can view the OM audit trail log by using the TSO SPOC application if the CSL Operations Manager (OM) is configured to produce an audit trail log. The OM audit trail log contains log records of commands, command responses, and system messages routed through OM.

OM audit trail logging is enabled by the AUDITLOG parameter in the CSL OM initialization parameters PROCLIB member (CSLOI*xxx*).

You can filter the records returned by TSO SPOC by specifying in the Preferences panel one or more IMSplex member names for which you want to see records. Alternately, you can specify the type of IMSplex member, such as IMS systems, automated operator programs, CQS, RM, and so forth.

When you are viewing the OM audit trail in TSO SPOC, a log record for command input, command output, or system message appears on a separate line. You can obtain additional information by:

- Clicking on a log entry for a system message, which produces a link that you can follow to a description of the message.
- Clicking on a log entry for a command response, which produces information about that command.

To search for a text string in the TSO SPOC Audit Trail panel, type `find` on the command line or click **View** > **Find** from the menu bar. A pop-up panel is displayed, prompting you to enter the text string. To find the same text string again, press PF5 or type `rfind` on the command line.

```
File  Display  View  Options  Help
------------------------------------------------------------
PLEX1               IMSplex Audit Trail
Command ===> ____________________________________________
---------------- Members. . ____________ Type . . ____________
MbrName Time                       Message
IMS1    2006.086 14:04:21.148419 DFSxxxxI message text
IMS1    2006.086 14:08:18.898122 DFSxxxxW message text
IMS1    2006.086 14:10:08.752387 DFSxxxxE message text
OMUSER1 2006.086 14:15:08.752387 Response for: QRY IMSPLEX
IMS1    2006.086 15:08:21.148419 DFSxxxxI message text
IMS1    2006.086 15:09:18.898122 DFSxxxxW message text




 F1=Help    F7=Up    F8=Down    F12=Cancel
```

Figure 4. TSO SPOC IMSplex Audit Trail panel

When you are using the TSO SPOC ISPF interface, the log entries for system messages are color coded in the TSO SPOC display. Error messages are displayed in red. Warning messages are displayed in yellow. All other messages are displayed in white.

Enabling the OM audit log display

You can enable the OM audit log display in order to display the current records that are being logged.

About this task

To enable the OM audit log display, follow these steps:

Procedure

1. Optionally define your audit trail preferences in the TSO SPOC Preferences panel.
2. Select **Audit Trail** from the panel selection screen.
3. Enter the z/OS log data set name that contains the OM audit log. This name must match the z/OS log data set name specified in the AUDITLOG subparameter of the IMSPLEX parameter in the CSLOIxxx PROCLIB member.
4. Enter the earliest date and time for which you want to see records.
5. Enter the latest data and time for which you want to see records. Enter an asterisk in the end date and end time fields to see the current records being logged.

Issuing Batch SPOC commands

You can use the Batch SPOC utility to submit IMS operator commands to an IMSplex. The Batch SPOC utility accepts any commands that are supported by the OM API.

The Batch SPOC utility uses:

- Program parameters to define the IMSplex environment, including IMSplex name, routing and the wait time.
- The SYSIN file as input for IMS operator commands.
- The SYSPRINT file to show SPOC-like formatted command response.

You can invoke the Batch SPOC utility using standard JCL statements. The following example shows a simple invocation, but you can call the utility using other valid JCL.

Sample batch job with multiple commands

```
//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 following program parameters define the IMSplex environment:

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 execute. The commands are executed serially. When one command completes, the next command is executed 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 nonblank 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 with the following format:

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

The SYSPRINT file will have the formatted command response. If more than one command is issued, the responses 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 style specified by the user. 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.

Sample batch job output

The following example shows sample batch job output:

```
============================================================
Log for. . : QRY IMSPLEX SHOW(JOB,TYPE,STATUS)

IMSplex . . . . . : 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

IMSplex  MbrName    CC Member   JobName  Type  Status
CSLPLEX1 OM1OM       0 USRT002  USRT002  AOP   ACTIVE
CSLPLEX1 OM1OM       0 OM1OM    OM1      OM    READY,ACTIVE
CSLPLEX1 OM1OM       0 RM1RM    RM1      RM    READY,ACTIVE
CSLPLEX1 OM1OM       0 SCI1SC   SCI1     SCI   READY,ACTIVE
CSLPLEX1 OM1OM       0 IMS1     IMS1     IMS   READY,ACTIVE
CSLPLEX1 OM1OM       0 SYS1     SYS1     IMS   READY,ACTIVE
============================================================
```

Sample batch job output with no response

If no wait time, WAIT=0, is specified, the command response is not available and therefore will not be printed. The SYSPRINT file will only have short summary information for each command:

```
============================================================
Log for. . : QRY IMSPLEX SHOW(JOB,TYPE,STATUS)

IMSplex . . . . . : PLEX1
Routing . . . . . :
Start time. . . . : 2006.075 15:36:28.11
============================================================
```

Related reference

Batch SPOC utility (CSLUSPOC) (System Utilities)

Modifying and controlling system resources

You establish the initial settings of IMS resources during IMS system definition.

Modifying system resources online

IMS supports two separate functions that support modifying IMS resources online: dynamic resource definition (DRD) and the online change function. Authorized system operators and database administrators (DBAs) can change various system resources using IMS commands.

Dynamic resource definition

With DRD, you can perform the following actions on IMS system resources:

- Use commands to create, update, and delete MODBLKS and MSC resources dynamically.
- Use the enhanced Destination Creation exit routine (DFSINSX0), formerly called the Output Creation exit routine, to create transactions (and, if necessary, the programs that are associated with the transactions) instead of using the online change process.

- Use the Program Creation user exit routine (PGMCREAT) to dynamically create the runtime program resourece for a program that is to be scheduled in a BMP or JBP dependent region.

DRD enables you to avoid the online change process for MODBLKS resources. To change resources other than MODBLKS and MSC resources online (for example, resources in IMS.ACBLIB), you still need to use the online change process.

DRD is enabled by specifying MODBLKS=DYN in the DFSDFxxx member or the DFSCGxxx member. To enable DRD for MSC resources, you must also specify MSCRSCS=DYN in the DFSDFxxx member.

When dynamic resource definition (DRD) is enabled in your IMS system, you can use the following type-2 commands to create, modify, and delete runtime resource and descriptor definitions for application programs, databases, routing codes and transactions.

- **CREATE**
- **UPDATE**
- **DELETE**
- **IMPORT**

The **CREATE**, **UPDATE**, **DELETE**, and **IMPORT** commands in a DRD-enabled system are essentially the same as adding, modifying, or deleting the system definition macros for IMS runtime resource definitions.

The **EXPORT** command is used to export MODBLKS runtime resource and descriptor definitions from the online IMS system to a resource definition data set (RDDS) or IMSRSC repository. The MODBLKS resource and descriptor definitions can then be added to IMS dynamically through the IMPORT command. For dynamically defined MSC resources, use the automatic export function to export the resources to the IMSRSC repository. The MSC resource definitions can then be added to IMS dynamically at IMS cold start by using the automatic import function.

By exporting runtime resource and descriptor definitions to a repository, you can create resources for another IMS in the IMSplex.

Modifications to IMS resources that are made by use of type-2 commands are recoverable across a warm or emergency restart. Runtime MODBLKS descriptor definitions and MSC definitions are lost when an IMS system is cold started unless you perform the following actions:

- For runtime MODBLKS descriptor definitions, export MODBLKS resource and descriptor definitions to the RDDS or repository while IMS is running. MODBLKS resource and descriptor definitions can be exported either explicitly by use of the **EXPORT DEFN** command, or automatically if AUTOEXPORT is enabled in the DFSDFxxx PROCLIB member.
- For MSC definitions, use the automatic export function to export the definitions to the IMSRSC repository.

If you want to modify resources, use the **QUERY** command to view the resource information before making the change.

You must ensure that:

- Commands are routed to and executed on the systems where you want the changes to be applied.
- Any changes made dynamically with the DRD commands are recovered across a cold start. Here are some examples of possible methods:
  - Using automatic export to export updated resource definitions to an RDDS or the IMSRSC repository, and then using automatic import during IMS cold start to retrieve the definitions from the RDDS or the IMSRSC repository.
  - Issuing the **EXPORT DEFN TARGET(REPO)** command while IMS is running and then using automatic import from the repository.

    You can periodically issue the **EXPORT DEFN TARGET(REPO) OPTION(CHANGESONLY)** command to write changes to the repository.

The following IMS resources and resource descriptors can be defined dynamically:

- Application programs

- Databases
- Fast Path routing codes
- Transactions
- MSC resources, including logical links, physical links, logical link paths (MSNAMEs), and remote logical terminals (LTERMs)

You can enter the **CREATE**, **UPDATE**, **DELETE**, and **MODIFY** type-2 commands directly through:

- TSO SPOC
- Batch SPOC
- REXX automation (through the REXX SPOC API)
- Your own programs that use the OM API

Similarly, you can use the Manage Resource panels to enter DRD commands by selecting an option from the IMS Application Menu. The Manage Resource panels are a hierarchy of ISPF panels that offer an alternative way to enter commands. From the Manage Resource panels, you can:

- Define and control IMS resources
- Select an action that you want to perform (for example, CREATE)
- Manage attributes of a resource

```
--------------------------------------------------------------------------------
                                                                        MORE: +
 * NAME   Transaction name . . 4NEWTRAN_____________________________
 SET
   AOCMD    AOI command option . . . . . . . N___       CMD, N, Tran, Y
   CLASS    Class. . . . . . . . . . . . . . 1__        1-999
   CMTMODE  Commit mode. . . . . . . . . . . SNGL       Sngl, Mult
   CONV     Conversational . . . . . . . . . N          Y, N
   DCLWA    Log write-ahead option . . . . . Y          Y, N
   DIRROUTE MSC direct routing option. . . . N          Y, N
   EDITRTN  Input edit routine . . . . . . . __________
   EDITUC   Edit to uppercase. . . . . . . . Y          Y, N
   EMHBSZ   EMH buffer size. . . . . . . . . ________   12-30720
--------------------------------------------------------------------------------
```

Figure 5. Example of a Manage Resource panel

You can use many IMS commands to perform similar control functions for different types of resources.

Online change function

There are two variations of online change:

- Local online change, which enables you to make online changes to IMS resources for one IMS.
- Global online change, which enables you to coordinate online changes to IMS resources across all IMS systems in an IMSplex.

Global online change is not supported in all IMS environments that support local online change. Environments that require the MODSTAT/MODSTAT2 data sets do not support global online change. FDBR, XRF alternate, and DBCTL-standby do not participate in global online process, however they keep track of the online change from the log records.

Table 2. IMS environments that support online change

Environment	Supports local online change?	Supports global online change?
DB/DC	Yes	Yes
DBCTL	Yes	Yes
DBCTL-standby	No	No
DCCTL	Yes	Yes

Table 2. IMS environments that support online change (continued)

Environment	Supports local online change?	Supports global online change?
FDBR	No	No
XRF active	Yes	Yes
XRF alternate	No	No

Related concepts

Overview of the IMSRSC repository (System Definition)

"Creating, updating, deleting, and querying resource definitions dynamically" on page 24

If dynamic resource definition (DRD) is enabled in your IMS system, you can create, update, delete, and query certain IMS resources online through several type-2 commands. You can also create runtime descriptor definitions to use as templates for creating additional resources.

Overview of the local online change function (System Administration)

Overview of dynamic resource definition (System Definition)

Related tasks

Exporting resource and descriptor definitions (System Definition)

Importing resource and descriptor definitions (System Definition)

Related reference

"List of commands with similar functions for multiple resources" on page 14

These tables show the IMS commands and how they affect certain resources.

List of commands with similar functions for multiple resources

These tables show the IMS commands and how they affect certain resources.

Note: In the following tables, a blank indicates that the command is not applicable for the resource.

Subsections:

Telecommunication line, physical terminal, or node

The following table shows the IMS commands that affect the telecommunication line, physical terminal, or node resources. This table indicates whether each resource can perform the following functions after the command is issued:

- Receive input
- Send output
- Output message queuing

Table 3. IMS commands that affect telecommunications line, physical terminal, or node resources

IMS command	Receive input	Send output	Output message queuing
/ASSIGN	Y	Y	Y
/LOCK	N	N	Y
/MONITOR	Y	N	Y
/PSTOP	N	N	Y
/PURGE	N	Y	Y
/RSTART	Y	Y	Y
/START	Y	Y	Y
/STOP	N	N	Y
/UNLOCK	Y		Y

Note: /MONITOR, **/PSTOP**, **/PURGE**, and **/RSTART** refer to the telecommunication line or physical terminal, not to the node.

Logical terminal

The following table shows the IMS commands that affect logical terminal resources. This table indicates whether these resources can perform the following functions after the command is issued:

- Receive input
- Send output
- Queuing from other terminals

Table 4. IMS commands that affect logical terminal resources

IMS command	Receive input	Send output	Queuing from other terminals
/ASSIGN	Y	Y	Y
/LOCK	N	N	N
/PSTOP	N	N	Y
/PURGE	N	Y	N
/RSTART			
/START	Y	Y	Y
/STOP	N	N	N
/UNLOCK		Y	Y

MSC logical link (MSLINK) commands

Several commands are used to control logical links to other systems (MSLINKs). Logical link definitions are defined with the MSLINK macro or the CREATE MSLINK command, which enable you to name the link, to associate the logical link with a logical link defined in a partner system, and to define the type of physical link the logical link can be used with.

The following table describes the effect of each of the commands that you can use to control logical link definitions.

Table 5. Commands used to control MSC logical links definitions

MSC use and effect	Type-1 command	Type-2 command
Set or reset the ASR.	**/CHANGE LINK***link* **ASR ON|OFF**	**UPDATE MSLINK SET(ASR(ON|OFF))**
Change the session resynchronization option.	**/CHANGE LINK** *link* **FORCSESS|SYNCSESS|COLDSESS**	**UPDATE MSLINK SET (SYNCOPT(***FORCSESS*|*SYNCSESS*|*COLSDESS***))**
Change the default modetable name.	**/CHANGE LINK** *link* MODE *name*	**UPDATE MSLINK SET(MODETBL(***name***)**
Assign to a new physical link.	**/MSASSIGN LINK** *link* **MSPLINK** *name*	**UPDATE MSLINK SET(MSPLINK(***name***)**
Stop the current link and sending and receiving of messages.	**/PSTOP LINK** *link*	**UPDATE MSLINK STOP(COMM)**
Force the stoppage of a link.	**/PSTOP LINK** *link* **PURGE|FORCE**	**UPDATE MSLINK STOP(COMM) OPTION(FORCE)**
Start a previously stopped MSLINK and start the queuing of or sending of messages to logical links on another system.	**/RSTART LINK** *link*	**UPDATE MSLINK START(COMM)**
Start a previously stopped MSLINK and start the queuing of or sending of messages to logical links on another system, but use modetable for this session only.	**/RSTART LINK** *link* **MODE** *name*	**UPDATE MSLINK START(COMM) SET(MODETBL(***name***))**
Start or stop an internal trace.	**/TRACE SET ON|OFF LINK** *link*	**UPDATE MSLINK START(TRACE)** or **UPDATE MSLINK STOP(TRACE)**
Start or stop an XRF takeover trace.	**/TRACE SET ON|OFF LINK** *link* **TAKEOVER**	**UPDATE MSLINK START(TKOTRC)** or **UPDATE MSLINK STOP(TKOTRC)**
Start or stop the bandwidth mode.	Not applicable	**UPDATE MSLINK SET(BANDWIDTH(ON|OFF))**
Change the send/receive buffer size.	Not applicable	**UPDATE MSLINK SET(BUFSIZE(***size***))**
Change the logical link name.	Not applicable	**UPDATE MSLINK SET(MSLINK(***name***))**
Change the partner id.	Not applicable	**UPDATE MSLINK SET(PARTNER(***id***))**

MSC physical link (MSPLINK) commands

Several commands are used to control physical links (MSPLINKs). The MSPLINK macro or the type-2 CREATE MSPLINK command can define the following types of connections between two systems:

- Channel-to-channel (CTC)

- Memory-to-memory (MTM)
- TCP/IP
- VTAM®

The following table describes the effect of each of the commands that you can use with to control physical link paths to IMS systems.

Table 6. Commands used to control MSC physical links

MSC use and effect	Type-1 command	Type-2 command
Disable logons to this physical link.	**/PSTOP MSPLINK**	**UPDATE MSPLINK STOP(LOGON)**
Enable logons to this physical link.	**/RSTART MSPLINK**	**UPDATE MSPLINK START(LOGON)**
Change the address of the channel-to-channel adapter.	Not applicable	**UPDATE MSPLINK SET(ADDR(***addr***))**
Set or reset ASR for all assigned logical links.	Not applicable	**UPDATE MSPLINK SET(ASR(ON|OFF))**
Change the order in which IMS restarts TCP/IP and VTAM links after an XRF takeover.	Not applicable	**UPDATE MSPLINK SET(BACKUP(***n*|NO**))**
Change the input and output buffer sizes for each logical link that is assigned to a physical link.	Not applicable	**UPDATE MSPLINK SET(BUFSIZE(***size***))**
Change the default modetable name for all assigned logical links.	Not applicable	**UPDATE MSPLINK SET(MODETBL(***name***))**
Change the physical link name.	Not applicable	**UPDATE MSPLINK SET(MSPLINK(***name***))**
Change the VTAM node name.	Not applicable	**UPDATE MSPLINK SET(NODE(***name***))**
Change the number of parallel sessions that can be active for TCP/IP and VTAM physical link types.	Not applicable	**UPDATE MSPLINK SET(SESSION(***n***))**

MSC logical link path (MSNAME) commands

Several commands are used to control logical link paths (MSNAMEs). Logical link paths are defined by system identifier (SYSID) pairs that identify sending and destination systems. Logical link paths are named by the MSNAME macro or by the CREATE MSNAME command.

The following table describes the effect of each of the commands that you can use to control logical link paths resources.

Table 7. Commands used to control MSC logical link paths

MSC use and effect	Type-1 command	Type-2 command
Change the local system identification SIDs.	Not applicable	**UPDATE MSNAME SET(SIDL(id)**
Change the remote system identification SIDs.	Not applicable	**UPDATE MSNAME SET(SIDR(id)**
Assign a new logical link path to a logical link.	**/MSASSIGN MSNAME**	**UPDATE MSNAME SET(MSLINK(name))**
Halt the queuing of primary requests for all remote terminals and programs represented by the MSNAME. Continued conversations and secondary requests are handled. Primary requests entered through an input terminal receive message DFS065. Requests from other systems that require use of the logical link path for a response are not accepted but remain queued in the sending system.	**/PURGE MSNAME**	**UPDATE \MSNAME STOP(Q) START(SEND)**
Start a previously stopped MSNAME and start the queuing of or sending of messages to logical link paths.	**/START MSNAME**	**UPDATE MSNAME START(Q,SEND)**

Table 7. Commands used to control MSC logical link paths (continued)

MSC use and effect	Type-1 command	Type-2 command
Stop the sending and receiving of primary request messages associated with the logical link path. When an MSNAME is stopped by an input system, primary requests for remote programs or terminals associated with the stopped logical link path are canceled, and message DFS065 is returned to the input terminal. Conversations in progress are allowed to continue. When an MSNAME is stopped by a destination system, messages received from other systems over a stopped logical link path cause a logical link path to be stopped in the sending system (input or intermediate), and MTOs of the sending and receiving systems to be notified by messages DFS2140 and DFS2142, respectively. The message remains queued in the sending system until the logical link is subsequently started in both systems.	**/STOP MSNAME**	**UPDATE MSNAME STOP(Q,SEND)**

Transaction

The following table shows the IMS commands that affect transaction resources. This table indicates whether these resources can perform the following functions after the command is issued:

- Message scheduling by transaction
- Message queuing by transaction

Table 8. IMS commands that affect transaction resources

IMS command	Message scheduling by transaction	Message queuing by transaction
/ASSIGN or **UPDATE TRAN SET**	Y	Y

Table 8. IMS commands that affect transaction resources (continued)

IMS command	Message scheduling by transaction	Message queuing by transaction
/LOCK or **UPDATE TRAN SET(LOCK|ON|OFF)**	N	Y
/MSASSIGN or **UPDATE TRAN SET(MSNAME)**	Y	Y
/PSTOP or **UPDATE TRAN STOP(SCHD) START(Q)**	N	Y
/PURGE or **UPDATE TRAN START(SCHD) STOP(Q)**	Y	N
/START or **UPDATE TRAN START(Q,SCHD)**	Y	Y
/STOP or **UPDATE TRAN STOP(Q,SCHD)**	N	N
/UNLOCK or **UPDATE TRAN SET(LOCK|ON|OFF**)	Y	Y

Transaction class

The following table shows the IMS commands that affect transaction class resources. This table indicates whether these resources can perform transaction scheduling by class after the command is issued.

Table 9. IMS commands that affect transaction class resources

IMS command	Transaction scheduling by class
/ASSIGN	Y
/MSASSIGN	Y
/START	Y
/STOP	N
UPDATE TRAN	Y or N[1]
UPDATE TRAN SET	Y

Table 9. IMS commands that affect transaction class resources (continued)

IMS command	Transaction scheduling by class
UPDATE TRAN START (Q)	Y
UPDATE TRAN STOP (Q)	N

[1] Whether the transaction class can perform transaction scheduling by class after the UPDATE command is issued depends on the parameters and keywords specified in this command.

Program

The following table shows the IMS commands that affect program resources. This table indicates whether the program can be started after the command is issued.

Table 10. IMS commands that affect program resources

IMS command	Can the program be started?
/ASSIGN	Y
/LOCK	N
/START	Y
/STOP	N
/UNLOCK	Y
UPDATE PGM SET(LOCK(ON))	N
UPDATE PGM SET(LOCK(OFF)	Y
UPDATE PGM START(SCHD)	Y
UPDATE PGM STOP(SCHD)	N

Database

The following table shows the IMS commands that affect database resources. This table indicates whether the database can be accessed after the command is issued.

Table 11. IMS commands that affect database resources

IMS command	Can the database be accessed?
/DBDUMP	N
/DBRECOVERY	N
/LOCK	N
/START	Y
/STOP	N
/UNLOCK	Y
UPDATE	Y or N[1]
UPDATE DB START(ACCESS)	Y
UPDATE DB STOP(ACCESS)	N
UPDATED DB STOP(UPDATES)	Y
UPDATE DB STOP(SCHD)	N

Table 11. IMS commands that affect database resources (continued)

IMS command	Can the database be accessed?
UPDATE DB SET(LOCK(ON))	N
UPDATE DB SET (LOCK(OFF))	N

[1] Whether the database can be used after the UPDATE command is issued depends on the parameters and keywords specified in this command.

Area

The following table shows the IMS commands that affect area resources. This table indicates whether the area can be accessed after the command is issued.

Table 12. IMS commands that affect area resources

IMS command	Can the area be accessed?
/DBRECOVERY	N
/START	Y
/STOP	N
UPDATE	Y or N[1]
UPDATE AREA START(ACCESS)	Y
UPDATE AREA STOP(ACCESS)	N
UPDATE AREA STOP(SCHD)	N

[1]Whether the database can be used after the UPDATE command is issued depends on the parameters and keywords specified in this command.

Subsystem

The following table shows the IMS commands that affect subsystem resources. This table indicates whether the subsystem can be attached after the command is issued.

Table 13. IMS commands that affect subsystem resources

IMS command	Can the subsystem be attached?
/START	Y
/STOP	Y
/CHANGE	N

User

The following table shows the IMS commands that affect user resources. This table indicates whether the user resource can perform the following functions after the command is issued:

- Receive input
- Send output
- Output message queuing

Table 14. IMS commands that affect user resources

IMS command	Receive input	Send output	Output message queuing
/ASSIGN	Y	Y	Y
/RSTART	Y	Y	Y
/START	Y	Y	Y
/STOP	N	N	Y

Related concepts

"Commands for IMS operations tasks" on page 48
You can use various commands to complete IMS operation tasks.

"Modifying system resources online" on page 11
IMS supports two separate functions that support modifying IMS resources online: dynamic resource definition (DRD) and the online change function. Authorized system operators and database administrators (DBAs) can change various system resources using IMS commands.

Modifying dependent regions

To modify the assignment of classes to regions or to adjust the processing load among message regions, you can use the type-1 **/ASSIGN** command or the type-2 **UPDATE** command.

About this task

Use the **/ASSIGN TRAN** or **UPDATE TRAN SET(CLASS(***new_class_number***))** command to modify the assignment of classes to regions.

Modifying telecommunication lines

To discard response-mode output messages, you can use the **/DEQUEUE** command

About this task

Use the **/DEQUEUE** command to discard response-mode output messages before you enter an **/RSTART LINE** command.

How to modify terminals

Terminals and links to terminals can be modified through various type-1 commands. You can use these commands and their variations to manage terminals.

Use the **/DISPLAY PTERM** and **/DISPLAY NODE** command to display the status of terminals and nodes. You can also use the **/DISPLAY STATUS** command, which shows all resources that have a status requiring operator intervention, including terminals and nodes.

Use the **/ASSIGN LTERM** command to modify the assignment of logical terminals to physical terminals or nodes. The new assignment remains in effect until the next cold start or until you issue another **/ASSIGN** command.

Use the **/DEQUEUE** command to discard full-function response-mode output so that the **/RSTART** command can reset terminal response mode.

You can reset static nodes that are hung in Fast Path input response mode by issuing the **/STOP NODE** and **/START NODE** commands in sequence.

Use the **/COMPT** command for VTAM terminals (nodes) to notify IMS that a terminal component is operable or inoperable.

IMS provides a VTAM I/O Timeout facility to detect VTAM hung nodes and determine what action, if any, should be taken. Use the **/TRACE** command to start and stop the VTAM I/O Timeout facility. Use the **/IDLE** command to deactivate a node and the **/ACTIVATE** command to activate a node. Use the **/DISPLAY** command to display all nodes that have I/O outstanding for a time period greater than that specified during system definition.

Modifying and controlling transactions

You can use the **/ASSIGN TRAN** or **UPDATE TRAN SET(CPRI(***new_current_priority***))** command to reassign the scheduling priorities established for transactions during system definition. The new assignments remain in effect until the next cold start or until you issue another **/ASSIGN** or **UPDATE** command.

In a shared-queues environment, you can use the **/ASSIGN TRAN** or **UPDATE TRAN SET(CLASS(***new_class_number***))** command to control which IMS subsystems can run certain types of transactions by assigning transactions to a particular class.

For example, you can define TRANA to class 4 on IMSA and to class 255 on IMSB and IMSC, so that only IMSA can run TRANA. If IMSA fails, you can reassign TRANA on either IMSB or IMSC to a class that these IMS subsystems can run.

Attention: Do not use the **/STOP TRAN** or **UPDATE TRAN STOP(Q,SCHD)** command to control which IMS subsystems can run certain types of transactions.

Database control

You can use the type-1 **/DBDUMP DB** or type-2 **UPDATE DB** commands to control databases and access to databases.

Use the **/DBDUMP DB** or **UPDATE DB STOP(UPDATES)** command to stop online update access to a database so that you can produce an offline dump of the database.

Use the **/DBRECOVERY DB** or **UPDATE DB STOP(ACCESS)** command to stop all online access to a database. Use it to recover a database offline.

Normally, IMS switches to using the next OLDS when you enter the **/DBDUMP** or **/DBRECOVERY** command. This switch does not occur if you specify the NOFEOV keyword on either command. When you enter the **UPDATE DB STOP(ACCESS)** command to stop online access to the database, IMS does not switch to the next OLDS. You can specify OPTION(FEOV) if you want IMS to switch to the next OLDS.

Specify the GLOBAL keyword on the **/DBDUMP** or **/DBRECOVERY** command to have the command apply to all subsystems sharing the database. The IRLM must be active if you use this keyword. The default is LOCAL, which specifies that the command applies only to the subsystem on which you enter the command.

Creating, updating, deleting, and querying resource definitions dynamically

If dynamic resource definition (DRD) is enabled in your IMS system, you can create, update, delete, and query certain IMS resources online through several type-2 commands. You can also create runtime descriptor definitions to use as templates for creating additional resources.

You can use the following type-2 commands:

- **CREATE**
- **UPDATE**
- **DELETE**
- **QUERY**
- **IMPORT**

You can also create *runtime descriptor definitions*, which are a set of resource definitions that you can use as templates to create additional resources.

The following IMS resources and resource descriptors can be dynamically defined:

- Application programs
- Databases
- Fast Path routing codes
- Transactions

You can use Fast Path routing codes and transactions as soon as they are correctly defined by the **CREATE** command.

After creating an application program or database by using the **CREATE** command, you must further define the application program or database, as well as its relationships to other resources, by running the appropriate generation utility, such as the Database Description Generation (DBDGEN) utility or the Program Specification Block Generation (PSBGEN) utility.

You can also create and modify these resources by using the Managing Resources series of ISPF panels which are invoked through the IMS applications menu.

Related concepts

"Modifying system resources online" on page 11
IMS supports two separate functions that support modifying IMS resources online: dynamic resource definition (DRD) and the online change function. Authorized system operators and database administrators (DBAs) can change various system resources using IMS commands.

Dynamic resource definition (System Definition)

Creating runtime resource definitions or runtime descriptor definitions dynamically

The **CREATE** commands can be used to add resources dynamically without having to go through the online change process. You can use the **CREATE** command with the appropriate parameters to define a new runtime resource definition or runtime descriptor definition.

About this task

Runtime resource definitions and runtime descriptor definitions that are created by using the **CREATE** command exist until the next IMS cold start, unless they are deleted by using the **DELETE** command.

Runtime resource definitions and runtime descriptor definitions created dynamically by using the **CREATE** command are recoverable across an IMS warm start or an emergency restart. Resources and runtime descriptor definitions are lost if an IMS system is cold started, unless the resource and descriptor definitions were previously exported to a resource definition data set while IMS was running.

The **CREATE** command can only be issued through the Operations Manager API. The **CREATE** command can be used in all IMS environments, although some keywords are restricted. For example, you cannot use the **CREATE DB** command in an IMS DCCTL environment, and you cannot use the **CREATE TRAN** command in IMS DBCTL environment.

The **CREATE** command is not valid in an XRF alternate IMS system or an FDBR region.

The **CREATE** command is only valid if DRD is enabled. To enable DRD, specify MODBLKS=DYN in either the DFSDFxxx or the DFSCGxxx PROCLIB member.

Creating databases and database descriptors dynamically

Use the type-2 command **CREATE DB** to dynamically create a database. Use the type-2 command **CREATE DBDESC** to create a database descriptor. Optionally, you can use the Program Creation user exit routine (PGMCREAT) to dynamically create a database, but only one database can be created.

About this task

You can reference existing databases or database descriptors to use as a model by specifying the LIKE() parameter.

The **CREATE DB** and **CREATE DBDESC** commands are valid only in DB/DC and DBCTL systems.

The PGMCREAT user exit is valid in DB/DB, DBCTL, and DCCTL systems.

Related reference

CREATE DB command (Commands)

CREATE DBDESC command (Commands)

PGMCREAT user exit routine type (Exit Routines)

Creating application programs and application program descriptors dynamically

Use the type-2 command **CREATE PGM** to dynamically create an application program. Use the type-2 command **CREATE PGMDESC** to create an application program descriptor. Use the Program Creation user exit routine (PGMCREAT) to dynamically create an application program that is scheduled in a BMP or JBP dependent region.

About this task

You can reference existing application programs or application program descriptors to use as a model by specifying the LIKE() parameter.

The PGMCREAT user exit routine, the **CREATE PGM** command, and the **CREATE PGMDESC** command are valid in DB/DC, DBCTL, and DCCTL systems.

In a shared EMH queues environment, **CREATE PGM** creates a program even if there are messages on the shared EMH queues for the program. If the messages were placed on the shared EMH queues because the program was defined as Fast Path in another IMS system, but the program being created is defined as non Fast-Path, the IMS system in which the new program is being defined will be unable to access the messages on the shared EMH queues.

Related reference

CREATE PGM command (Commands)

CREATE PGMDESC command (Commands)

PGMCREAT user exit routine type (Exit Routines)

Creating Fast Path routing codes and Fast Path routing code descriptors dynamically

Use the type-2 command **CREATE RTC** to dynamically create a Fast Path routing code. Use the type-2 command **CREATE RTCDESC** to create a Fast Path routing code descriptor.

About this task

You can reference existing Fast Path routing codes or Fast Path routing code descriptors to use as a model by specifying the LIKE() parameter.

The **CREATE RTC** and **CREATE RTCDESC** commands are valid only in DB/DC and DCCTL systems in which Fast Path is installed.

Creating transactions and transaction descriptors dynamically

Use the type-2 command **CREATE TRAN** to dynamically create a transaction. Use the type-2 command **CREATE TRANDESC** to create a transaction descriptor.

About this task

You can reference an existing transaction or transaction descriptor to use as a model by specifying the LIKE() parameter.

The **CREATE TRAN** and **CREATE TRANDESC** commands are valid only in DB/DC and DCCTL systems.

In a shared message queues environment, the **CREATE TRAN** command creates a transaction, even if there are messages on the shared message queues for the transaction. If the messages were placed on the shared message queues using conversation, Fast Path, response, or serial attributes that differ from the attributes defined for the new transaction, the IMS system in which the new transaction is being created will be unable to access the messages on the shared message queues.

Updating runtime resource definitions or runtime descriptor definitions dynamically

Use the **UPDATE** command with the appropriate parameters to dynamically update a runtime resource definition or resource descriptor. Changes to resource definitions or runtime descriptor definitions made by using the **UPDATE** command exist until the next IMS cold start, unless they are modified by another **UPDATE** command or the runtime resource definition or runtime descriptor definition is deleted by using the **DELETE** command.

About this task

Updates to runtime resource definitions and runtime descriptor definitions are recoverable across an IMS warm start or an emergency restart. Updates to resources and runtime descriptor definitions are lost if an IMS system is cold started, unless the resource and descriptor definitions were previously exported to the resource definition data set while IMS was running.

The **UPDATE** command can be used to update runtime descriptor definitions dynamically without having to go through the online change process. The **UPDATE** command can only be issued through the Operations Manager API. Each resource or descriptor is updated individually. DRD does not work like the online changes where either all resources are created, updated, and deleted, or none of the resources are created, updated, or deleted.

You cannot update the attributes of runtime resource definitions that are currently in use or that have work in progress. If a resource is in use, the **UPDATE** command fails. To ensure the success of an **UPDATE** command, stop the resource and let the work complete before you issue the **UPDATE** command. In an IMSplex environment, the **UPDATE** command might succeed on some IMS systems and fail on others.

To minimize the likelihood that an update of a resource definition will fail, perform the following steps before issuing the **UPDATE** command:

Procedure

1. Stop the resource
2. Query the resource to check for work in progress
3. Dequeue transactions with the **QUEUE** command
4. Complete the work (if any)

Results

If all of the attributes specified by an **UPDATE** command are already defined for a resource, IMS takes no action and returns a completion code of zero.

Except for a few resource-specific exceptions, the **UPDATE** command is not valid with the SET( ) parameter or with any of the resource descriptor parameters if DRD is not enabled in either the DFSDFxxx or the DFSCGxxx PROCLIB member.

You cannot modify resource definitions for resources that are delivered with the IMS product. For example, you cannot modify the Fast Path utility program DBF#FPU0. You can, however, use the **UPDATE** command to change the status of certain IMS-delivered resources. For example, you might start or stop DBF#FPU0.

You also cannot modify the runtime descriptor definitions that are delivered with the IMS product, other than to select them as your default resource descriptor for a given resource. DFSDSDB1, DFSDSPG1, DBFDSRT1, and DFSDSTR1 are examples of the descriptors delivered with the IMS product.

Related reference

UPDATE PGM command (Commands)

Updating database attributes and database descriptors dynamically

Use the type-2 command **UPDATE DB** to update certain attributes of a database definition dynamically. Use the type-2 command **UPDATE DBDESC** to update a database descriptor.

About this task

The **DB** and **DBDESC** parameters of the **UPDATE** command are valid only in DB/DC and DBCTL systems.

You cannot specify the **UPDATE DB** command or the **UPDATE DBDESC** command in an XRF alternate IMS system or in an FDBR region.

If you use the **UPDATE DB** command to change the status of a database that has a MODBLKS definition type, the definition type of the database changes to UPDATE, as displayed by the **QUERY DB** command.

If DRD is not enabled, you cannot use the **UPDATE** command to update runtime resource definition attribute values.

If a database is in use, you cannot update the status of the database definition.

Changing RESIDENT(N) to RESIDENT(Y) takes effect at the next restart.

Updating application programs and application program descriptors dynamically

Use the type-2 command **UPDATE PGM** to update certain attributes of the definition of an application program dynamically. Use the type-2 command **UPDATE PGMDESC** to update an application program descriptor.

About this task

The **PGM** and **PGMDESC** parameters of the **UPDATE** command are valid in DB/DC, DBCTL, and DCCTL systems.

You cannot specify the **UPDATE PGM** command or the **UPDATE PGMDESC** command in an XRF alternate IMS system or in an FDBR region.

If there is work in progress for the application program, you cannot update application program definitions. In a local queues environment, if there are queued messages for transactions associated with the program being updated, the update fails.

In a shared-queues environment, the **UPDATE PGM** command updates a program even if there are messages queued for transactions that reference the program. If there are conflicts within a single IMS between the program attributes being changed and the attributes of the transactions and routing codes that reference the program, the update fails. However, if there are conflicts between the program attributes being changed and the attributes of messages queued for transactions that reference the program, this will not cause the update to fail. You must be careful when updating a program in a shared-queues environment to set the attributes consistently between programs, transactions, and routing codes across all IMS systems in the shared queues group. In a sysplex environment with multiple IMS systems, the update might succeed on some IMS systems and fail on others.

Changing RESIDENT(N) to RESIDENT(Y) takes effect at the next restart.

You must stop an application program before issuing an **UPDATE PGM** command if you are updating any of the following attributes of the application program definition:

- DOPT
- FP
- GPSB
- LANG
- PGMTYPE
- RESIDENT
- SCHDTYPE

If DRD is not enabled, the LOCK and TRANSTAT attributes are the only application program attributes that you can set by using the **UPDATE PGM** command.

For application programs delivered as part of the IMS product, you can issue the **UPDATE PGM** command with the following parameters:

- **START**
- **STOP**
- **SET(LOCK(ON|OFF))**
- **SET(TRANSTAT(Y|N))**

If you use the **UPDATE PGM** command to change any of the following attributes of an application program that has a MODBLKS definition type, the definition type of the application program changes to UPDATE, as displayed by the **QUERY PGM** command:

- DOPT
- FP
- GPSB
- LANG
- PGMTYPE
- SCHDTYPE
- TRANSTAT

You cannot update any of the following attributes of an application program if DRD is not enabled:

- DOPT
- FP
- GPSB
- LANG
- PGMTYPE
- RESIDENT
- SCHDTYPE

By using the **UPDATE PGM START(REFRESH)** command, you can roll out the application program change easily without having to figure out all the regions the program is scheduled in and stopping the regions manually.

The **UPDATE PGM START(REFRESH)** command is supported for programs scheduled in the following region types:

- MPP pseudo-wait-for-input (PWFI) regions in which the program is scheduled and the program is not preloaded by the DFSMPLxx PROCLIB member
- JMP PWFI regions in which the specified program name is scheduled

- MPP, JMP, and message-driven BMP regions in which the program is scheduled and that are running a transaction defined as WFI=YES

The **UPDATE PGM START(REFRESH)** command is not supported for MPP regions where the program is loaded by the DFSMPLxx PROCLIB member, IFP regions, JBP regions, and non-message driven BMP regions.

Updating Fast Path routing codes and Fast Path routing code descriptors dynamically

Use the type-2 command **UPDATE RTC** to dynamically update the definition of a Fast Path routing code. Use the type-2 command **UPDATE RTCDESC** to update a Fast Path routing code descriptor.

About this task

The **RTC** and **RTCDESC** parameters of the **CREATE** command are valid only in DB/DC and DCCTL systems in which Fast Path is installed.

If the Fast Path routing code is in use, the **UPDATE RTC** command fails if you are modifying either the INQ attribute or the PGM attribute of the definition of a Fast Path routing code.

You cannot update the following attributes of the definition of a Fast Path routing code if DRD is not enabled:

- INQ
- PGM

If you use the **UPDATE RTC** command to change either the INQ attribute or the PGM attribute in the definition of a Fast Path routing code that has a MODBLKS definition type, the definition type of the Fast Path routing code changes to UPDATE, as displayed by the **QUERY RTC** command.

Updating transactions and transaction descriptors dynamically

Use the type-2 command **UPDATE TRAN** to dynamically update the definition of a transaction. Use the type-2 command **UPDATE TRANDESC** to update a transaction descriptor.

About this task

The **TRAN** and **TRANDESC** parameters of the **UPDATE** command are valid only in DB/DC and DCCTL systems.

If DRD is not enabled, you cannot use the **UPDATE** command to update definitional attributes. You can, however, use the **UPDATE** command when DRD is not enabled to control the locking, tracing, queuing, and scheduling of transactions.

In a local queues environment, you cannot update the definitional attributes of a transaction if there are queued messages for the transaction being updated. In a shared queues environment, the **UPDATE TRAN** command updates the definitional attributes of a transaction, even if there are messages queued for the transaction on the shared queue.

You must be careful when updating the definitional attributes of a transaction to set attributes consistently across all IMSs in a shared queues group. Updating a transaction does not remove queued messages that are associated with the transaction from the queue. Also, queued messages are not automatically updated to be consistent with the changes made to associated transactions. Inconsistencies between queued messages and updated transactions can cause problems. In a sysplex environment with multiple IMS systems, an update might succeed on some IMS systems and fail on others.

If a transaction is in use or has work in progress, the **UPDATE TRAN** command fails if you are modifying any of the following attributes of the transaction definition:

- AOCMD
- CMTMODE
- CONV

- DCLWA
- DIRROUTE
- EDITRTN
- EDITUC
- EMHBSZ
- EXPRTIME
- FP
- INQ
- MSGTYPE
- MSNAME
- PGM
- PLCTTIME
- RECOVER
- REMOTE
- RESP
- SERIAL
- SIDL
- SIDR
- SPASZ
- SPATRUNC
- TRANSTAT
- WFI

If you use the **UPDATE TRAN** command to change any of the following attributes in the definition of a transaction that has a MODBLKS definition type, the definition type of the transaction changes to UPDATE, as displayed by the **QUERY TRAN** command:

- AOCMD
- CMTMODE
- CONV
- DCLWA
- DIRROUTE
- EDITRTN
- EDITUC
- EMHBSZ
- FP
- INQ
- MSGTYPE
- MSNAME
- PGM
- RECOVER
- RESP
- SERIAL
- SIDL
- SIDR
- SPASZ

- SPATRUNC
- WFI

Deleting runtime resource definitions or runtime descriptor definitions dynamically

Use the **DELETE** command to dynamically delete runtime resource definitions such as databases, application programs, Fast Path routing codes, or transactions. Also use the **DELETE** command to dynamically delete any runtime resource definition descriptors.

About this task

If you delete resource definitions or runtime descriptor definitions from all IMSs in an IMSplex, and the IMSRSC repository is used, you must also issue a **DELETE DEFN** command to delete the resource definitions or runtime descriptor definitions from the repository.

Recommendation: Delete the resource definitions or runtime descriptor definitions from all IMSs first, and then delete them from the repository.

Deleted runtime resource definitions and runtime descriptor definitions reappear if an IMS system is cold started, unless all resource and descriptor definitions are exported to a resource definition data set before IMS terminates and are imported from the resource definition data set when IMS cold starts. Deleted resources and descriptors remain deleted across either a warm start or an emergency restart. The **DELETE** command is recoverable.

In a shared message queues environment, **DELETE TRAN** dynamically deletes the transaction even if there are messages on the shared MSG queues for the transaction. If the transaction is deleted from all IMS systems, no IMS can access the messages on the shared MSG queues. Similarly, in a shared EMH queues environment, **DELETE PGM** deletes an application program even if there are messages on the shared EMH queues for the program. If the program is deleted from all IMS systems, no IMS system can access the messages on the shared EMH queues.

The **DELETE** commands are similar to local online change (**/MODIFY**) or global online change (**INITIATE OLC**) for IMS resources in the MODBLKS data set, except that the **DELETE** command works without requiring any offline procedures such as a MODBLKS system definition or the execution of the online change copy utility. Moreover, the **DELETE** command deletes resources and runtime descriptor definitions individually, while the **/MODIFY** and **INITIATE OLC** commands either delete a list of resources as a group or deletes no resources at all if an error occurs.

You cannot delete resources that are currently in use or that have work in progress. If a resource is in use, the delete fails. In a sysplex environment, the delete might succeed on some IMS systems and fail on others.

To minimize the likelihood that a delete of a resource will fail, perform the following steps before issuing the **DELETE** command:

Procedure

1. Stop the resource
2. Query the resource to check for work in progress
3. Dequeue transactions with the QUEUE command
4. Complete the work (if any)

Results

Examples of resources in use include a database being accessed by an application program, a **/STOP** or **UPDATE** command in progress for the resource, a scheduled application program, an active routing code, a transaction with messages queued.

You cannot issue the **DELETE** command against resources or runtime descriptor definitions that are identified with a definition type of IMS and delivered with the IMS product. For example, you cannot delete the IMS-defined application program DBF#FPU0 or the IMS-defined runtime descriptor definitions DFSDSDB1, DFSDSPG1, DBFDSRT1, and DFSDSTR1.

In a DB/DC environment, all of the parameters of the **DELETE** command are valid.

In a DBCTL environment, only the following forms of the **DELETE** command are valid:

- `DELETE DB`
- `DELETE DBDESC`
- `DELETE DEFN TYPE(DB | DBDESC | PGM | PGMDESC)`
- `DELETE PGM`
- `DELETE PGMDESC`

In a DCCTL environment, only the following forms of the **DELETE** command are valid:

- `DELETE DEFN TYPE(PGM | PGMDESC | RTC | RTCDESC | TRAN | TRANDESC)`
- `DELETE PGM`
- `DELETE PGMDESC`
- `DELETE RTC`
- `DELETE RTCDESC`
- `DELETE TRAN`
- `DELETE TRANDESC`

The commands **DELETE RTC** and **DELETE RTCDESC** are only valid in DB/DC or DCCTL environments in which Fast Path is installed.

Querying runtime resource definitions or runtime descriptor definitions dynamically

Use the **QUERY** command with the appropriate parameter to query database and program resources. All of the **QUERY** commands are type-2 commands that you can issue from the OM API.

About this task

These commands can be issued through the TSO SPOC or the Manage Resources options in the IMS Applications menu. These commands can also be issued to an IMSplex using the batch SPOC utility.

The **QUERY** commands return information based on the keyword specified. For example, issue the **QUERY DB** command to view the current database resource or database descriptor settings.

Several type-2 **QUERY** commands support dynamic resource definition (DRD):

- **QUERY DB**
- **QUERY DBDESC**
- **QUERY OLC**
- **QUERY PGM**
- **QUERY PGMDESC**
- **QUERY RTC**
- **QUERY RTCDESC**
- **QUERY TRAN**
- **QUERY TRANDESC**

When the IMSRSC repository is enabled, you can check for and identify which resource and descriptor definitions have not yet been exported to the IMSRSC repository by issuing the **QUERY** *rsc_type*

SHOW(EXPORTNEEDED) command, where *rsc_type* can be DB, DBDESC, PGM, PGMDESC, RTC, RTCDESC, TRAN, or TRANDESC.

Modifying ETO user IDs and assignments of ISC users

You can use the **/ASSIGN**, **/DISPLAY USER DEADQ**, **/DEQUEUE**, and the **/DISPLAY QCNT MSGAGE** commands to modify various aspects of assignments of ISC users.

About this task

For dynamic users, use the **/ASSIGN** command to change the assignment of an LTERM to another user (also called a subpool for ISC). The new assignment remains in effect until the next cold start or until you issue another **/ASSIGN** command. For static terminals that are not ISC, the LTERM can be reassigned to another terminal or node.

Use the **/DISPLAY USER DEADQ** command to list all message queues that are eligible for dead letter status. Use the **/ASSIGN** command to assign a dead letter queue to another user. Use the **/DEQUEUE** command to discard a dead letter queue.

In a shared-queues environment, use the **/DISPLAY QCNT MSGAGE** command to determine which messages, if any, are eligible for dead letter status.

Modifying Multiple Systems Coupling resources

After modifying MSC resources, use the **/MSVERIFY** command to ensure that the assignment produced a valid configuration. The **/MSVERIFY** command verifies the consistency of MSC system identifications (SYSIDs) and logical link paths (MSNAMEs) across two systems.

About this task

All changes made by the online commands remain in effect until the next cold start unless changes are exported to the IMSRSC repository or coded into IMS system generation.

If you use the IMSRSC repository to store dynamically defined MSC resources, ensure that automation and operational procedures that issue commands for MSC resources use type-2 commands, which specify link names, instead of type-1 commands, which specify link numbers. For example, instead of using the **/RSTART LINK 10** command to start a link, use the `UPDATE MSLINK NAME(`*`logicallinkname`*`) START(COMM)` command. During stage-1 system generation, the IMS system assigns numbers to logical links in the order in which the links are generated. However, the numbers for links are not stored in the IMSRSC repository. If logical links are referenced by using link numbers and are automatically imported from the IMSRSC repository, the numbers of the links are likely to change at the next IMS cold start.

Related concepts

"MSC operations" on page 197

In a non-sysplex environment, each IMS system in a Multiple Systems Coupling (MSC) configuration is operationally an independent unit. Each IMS system exclusively owns its own communication resources, and is controlled by its own master terminal.

Modifying security options

Commands to modify security options for RACF® are described.

About this task

Use the **/MODIFY PREPARE RACF**, **/DISPLAY MODIFY**, and **/MODIFY COMMIT** commands to reinitialize RACF information if you are not using a RACF data space. If you are using a RACF data space, use the RACF **SETROPTS RACLIST** command rather than the IMS **/MODIFY** command.

Use the **/SECURE APPC** command to control the RACF security level for input from LU 6.2 devices. Use the **/DISPLAY APPC** command to show the security level that is currently in effect. When IMS starts, the default is full security.

Use the **/SECURE OTMA** command to control the RACF security level for input from specified OTMA clients or for the entire OTMA z/OS cross-system coupling facility (XCF) group. Use the **/DISPLAY OTMA** command to show the security level that is currently in effect. When IMS starts, the default is full security.

Use the **/SECURE OTMA REFRESH** command to refresh security information cached in online memory after making a RACF change to a specific user ID. Cached OTMA security information can be refreshed globally by the OTMA TMEMBER, or by user ID, without impacting other cached user IDs.

Displaying and terminating conversations

Use the **/DISPLAY CONV** command to show the status of all conversations, held or active. You can terminate a conversation if necessary with the **/EXIT** command, but you should only do this after warning the end user.

About this task

Modifying and controlling subsystems

Use the **/CHANGE** command to delete an invalid network identifier (NID). If you need to disconnect from a specific subsystem, use the **/STOP** command. If the **/STOP** command does not work, use the z/OS **MODIFY** command.

About this task

Controlling OTMA input messages

Use the OTMA descriptor, the **/START TMEMBER INPUT** command, or a client-bid protocol message from an OTMA member to set the maximum transaction instance block (TIB) or maximum active input message count for an OTMA member.

About this task

The OTMA descriptor can be used to establish the initial maximum active input message count for an OTMA member. The client-bid protocol message can under-ride the value that is set by the OTMA descriptor, but not override it. The **/START TMEMBER INPUT** command can always be used to override the value that is specified by the OTMA descriptor or client-bid protocol message. If the value is not specified by any of these methods, the system default of 5000 TIBs is used.

OTMA monitors the growth of the active input messages from members. A warning message DFS1988W is sent to the console to indicate that a certain percentage of the TIB limit for the member has been reached. The message is sent when the member reaches 80% of the TIB limit and then for each additional 5% thereafter. When the maximum limit is reached, an error message DFS1989E is sent to the console. Any subsequent OTMA input messages are rejected with a new OTMA sense code X'30'.

Use the **/DISPLAY TMEMBER** or **/DIS OTMA** command to display the active input message count for an OTMA member.

To activate global flood control to suppress new OTMA input transactions, complete one of the following tasks:

- Specify the global flood limit value INPT for the DFSOTMA member in the OTMA client descriptor for all the OTMA members.
- Issue the IMS command **/START TMEMBER ALL INPUT *####***.

Recovery during the IMSRSC repository data set update process

The Repository Server (RS) uses a duplex pair of data sets for each IMSRSC repository, so that the RS can always recover data sets to the last completed and verified write activity.

The duplex pair of data sets is the primary repository index data set (RID) and repository member data set (RMD) (COPY1) and the secondary RID and RMD (COPY2). The RS always requires both the primary and secondary data set pairs to be available; the repository is otherwise stopped.

The repository can have an optional third defined data set pair, the spare RID and RMD (SPARE), which is used during the SPARE recovery process.

The RS writes to the repository in two phases:

1. In the first phase, the COPY1 RID and RMD are updated. At the crossover point, the change is considered completed. A successful return code is sent to the requesting client.
2. In the second phase, the same updates are written to the COPY2 RID and RMD. The request is completed.

The following sequence shows how the RS performs recovery of data sets if an I/O error occurs during the first phase of the update process (write to the COPY1 data set):

1. The request in progress fails with an error.
2. The repository becomes unavailable. The client is notified through any client connection exits with the `Repository unavailable` status.
3. If the RS verifies that a spare data set pair is unavailable, the repository is stopped. The client is notified that the repository is stopped. Administrator intervention is required to delete the data set in error and define a new data set and restart the repository. During repository start processing, data is copied from the valid data set to the newly added data set.
4. If the RS verifies that a valid spare data set pair is available and the other conditions for recovery are met, then spare recovery processing is started:
   a. Any client connection exits are driven with the `Repository recovery started` status.
   b. When recovery processing is completed, data is copied from the COPY2 data set to the SPARE data set pair and the status of the SPARE data set pair is changed to COPY1. The repository becomes available, and any client connection exits are driven with the `Repository recovery ended successfully` status.
   c. If an error occurs during recovery processing, any client connection exits are driven with the `Repository recovery error` status.

If the RS fails during the second phase of the update process (write to the COPY2 data set), the RS performs the same tasks, except that data is copied from the COPY1 data set to the SPARE data set pair and the status of the SPARE data set pair is changed to COPY2. As the first phase of the update process is completed, the request in progress is considered completed. A successful return code is returned to the caller.

If the automated SPARE recovery process fails, the repository might be in a CLOSE state.

If a repository remains in CLOSED state at the end of SPARE recovery processing, perform the following steps to make the repository available:

1. Re-allocate the discarded repository data sets.
2. Issue the **LIST** FRPBATCH command to check if the repository is in a STOP state. If it is not, issue the **STOP** FRPBATCH command to change the state to STOP.
3. Issue the **DSCHANGE** FRPBATCH command to change the re-allocated discarded data sets back to a SPARE state.
4. Issue the **START** FRPBATCH command to start the repository.

   This resumes the data set recovery process.

For automated recovery to take place, either the primary RID and RMD or the secondary RID and RMD must be valid and a SPARE repository data set pair must be available.

If a SPARE repository data set pair is not defined when the failure occurs, the repository is stopped. You must define new data sets for the primary or secondary data sets and start the repository. Perform the following procedure:

1. Allocate new repository data sets to take the place of the failed primary or secondary data sets and to create a SPARE repository data set pair.
2. Issue the **DSCHANGE** FRPBATCH command to discard the repository data sets in error.
3. Issue the **UPDATE** FRPBATCH command to replace the failed data sets with the newly allocated data sets for the existing repository.
4. Issue the **START** FRPBATCH command to start the repository.

   This resumes the data set recovery process.

You can also use the **F reposervername,ADMIN** command equivalents to the FRPBATCH commands.

The following topics describe the recovery procedures after the failure of both the primary and secondary RIDs and RMDs.

Related concepts

Overview of the IMSRSC repository (System Definition)

IMSRSC repository and RS catalog repository data sets (System Definition)

Recovery of changed runtime resource and descriptor definitions (System Definition)

Deleting runtime resource and descriptor definitions (System Definition)

Updating IMSRSC repository specifications in the RS catalog repository (System Administration)

Related reference

F reposervername,ADMIN (Commands)

Related information

Commands for FRPBATCH (System Programming APIs)

Recovering IMSRSC repository data sets when all IMS systems and an RM system are active in an IMSplex

You can recover IMSRSC repository data sets when all IMS systems and an RM system are active in an IMSplex.

About this task

Perform the following procedure:

Procedure

1. Define new primary and secondary repository data sets for the repository that failed, by using the FRPBATCH **UPDATE** command.
2. Start the repository by using the FRPBATCH **START** command or the **F** *reposervername*,ADMIN START command.
3. List the repository by using the FRPBATCH **LIST** command or the **F** *reposervername*,ADMIN DISPLAY command.
4. Write the resource definitions to the repository from each active IMS by using the IMS **EXPORT** command.
   - If your IMS systems are cloned, route the following command to one of the active IMS systems in the IMSplex and have it write its resource definitions to the repository for all of the IMS systems in the IMSplex.

     ```
     EXPORT DEFN TARGET(REPO) TYPE(ALL) NAME(*) SET(IMSID
     (IMSids_of_all_IMSs_sharing_repository))
     ```

- If your IMS systems are not cloned, route the following command to each of the IMS systems in the IMSplex so that each IMS writes its own runtime definitions to the repository.

  ```
  EXPORT DEFN TARGET(REPO) TYPE(ALL) NAME(*)
  ```

  This command writes the runtime resource definitions from each IMS to the repository.

IMS can continue processing while the repository data sets are being recovered.

Related concepts
Overview of the IMSRSC repository (System Definition)
Change lists for the IMSRSC repository (System Definition)
Related reference
EXPORT command (Commands)
F reposervername,ADMIN (Commands)
LIST command for FRPBATCH (System Programming APIs)
START command for FRPBATCH (System Programming APIs)
UPDATE command for FRPBATCH (System Programming APIs)

Recovering IMSRSC repository data sets when one or more IMS systems are down but an RM system is active

You can recover IMSRSC repository data sets when one or more IMS systems are down but IMS logs are available and an RM system is active.

About this task

Perform the following procedure:

Procedure

1. Define new primary and secondary repository data sets for a repository that failed, by using the FRPBATCH **UPDATE** command.
2. Start the repository by using the FRPBATCH **START** command or the **F** *reposervername*,ADMIN START command.
3. List the repository information by using the FRPBATCH **LIST** command or the **F** *reposervername*,ADMIN DISPLAY command.
4. Write the resource definitions to the repository by performing one of the following steps:
   - If your IMS systems are cloned, route the following command to one of the active IMS systems in the IMSplex and have it write its resource definitions to the repository for all of the IMS systems in the IMSplex.

     ```
     EXPORT DEFN TARGET(REPO) TYPE(ALL) NAME(*) SET(IMSID
     (IMSids_of_all_IMSs_sharing_repository))
     ```

   - If your IMS systems are not cloned, take one of the following steps:
     - Issue the **EXPORT DEFN TARGET(REPO) TYPE(ALL) NAME(*)** command at each active IMS system to route the command to each active IMS system in the IMSplex. This command writes the runtime resource definitions from each active IMS to the repository.
     - Generate a resource definition data set (RDDS) for each IMS system that is down. The RDDS can be generated from the IMS logs by using the DFSURCL0 utility. Then, run the CSLRUP10 utility to write the resource definitions from each generated RDDS to the repository.

Results

If an **IMPORT DEFN SOURCE(REPO) SCOPE(ALL)** command was issued before the data sets were recovered and it resulted in the IMS change lists to be created for the IMS systems that are down, no IMS

change lists will exist in the IMSRSC repository after the data sets are recovered. You must either reissue the **IMPORT DEFN SOURCE(REPO) SCOPE(ALL)** command to re-create the IMS change lists, or issue the **IMPORT** command after the IMS that was down and had a change list created is restarted.

Related concepts

Overview of the IMSRSC repository (System Definition)

Change lists for the IMSRSC repository (System Definition)

Related reference

EXPORT command (Commands)

Repository to RDDS utility (CSLURP20) (System Utilities)

Create RDDS from Log Records utility (DFSURCL0) (System Utilities)

F reposervername,ADMIN (Commands)

LIST command for FRPBATCH (System Programming APIs)

START command for FRPBATCH (System Programming APIs)

UPDATE command for FRPBATCH (System Programming APIs)

Recovering IMSRSC repository data sets when no IMS systems or RM systems are active

You can recover IMSRSC repository data sets when all IMS systems and RM systems are down and no IMS log is available but backup copies of the repository data sets are available.

About this task

Any IMS IDs that were deleted from the IMSRSC repository using the **DELETE DEFN TYPE(IMSIDMBR)** command after the repository backup was taken will now be in the IMSRSC repository. If the resource definitions for the IMS ID and the IMS IDs are no longer needed in the IMSRSC repository you must reissue the **DELETE DEFN TYPE(IMSIDMBR)** command.

Procedure

1. Define the primary and secondary data sets from the backup copies.
2. Start the repository by using the FRPBATCH **START** command or the F *reposervername*,ADMIN START command.

   The repository is started from the backup copies of data.
3. List the repository information by using the FRPBATCH **LIST** command or the **F** *reposervername*,ADMIN DISPLAY command.
4. Cold start IMS systems from the repository.
5. To bring the repository and IMS systems to the same state as the time of failure, reissue the commands that you used to create, update, or delete the resource definitions between the time the backup was generated and the IMS cold start.

   The type-1 or type-2 commands that were issued from OM API can be obtained from the OM audit log or command responses sent to OM API.

Results

Any new IMS change lists in the IMSRSC repository that are created after the backup time are lost. To re-create these IMS change lists, you must reissue the **IMPORT DEFN** commands that created them.

Any old IMS change lists in the backup data sets exist in the IMSRSC repository. If any of these lists are not needed because IMS was restarted or a **DEL DEFN TYPE(CHGLIST)** command was issued, you must issue the **DEL DEFN TYPE(CHGLIST)** command to delete the residual change lists that are no longer needed.

Related concepts

Overview of the IMSRSC repository (System Definition)

Related tasks

"Cold starting an IMS system that uses the IMSRSC repository" on page 78
If IMS is using the IMSRSC repository, and AUTOIMPORT=AUTO or REPO is defined in the DYNAMIC_RESOURCES section of the DFSDFxxx member of the IMS PROCLIB data set, the stored resource definitions are read from the repository during IMS cold start if the repository contains the resource definitions for the IMS that is cold starting.

Related reference

F reposervername,ADMIN (Commands)
LIST command for FRPBATCH (System Programming APIs)
START command for FRPBATCH (System Programming APIs)
UPDATE command for FRPBATCH (System Programming APIs)

Enabling and disabling IMS functions

If an IMS function is not enabled by default, you can enable the function dynamically while IMS is running by issuing the **UPDATE IMSFUNC** command. Or, you can enable the function statically by defining the function as enabled in the DFSDF*xxx* member of the IMS PROCLIB data set, and then cold starting IMS.

Before you begin

To enable IMS functions dynamically, ensure that OM and SCI are enabled.

About this task

Changes that you make by using the **UPDATE IMSFUNC** command are logged in the x'22' map byte x'31' log record and are recoverable across an IMS restart. For example, if a function is enabled in the DFSDF*xxx* member and later disabled by using the **UPDATE IMSFUNC**, the disabled value is recovered if IMS restarts.

If you change a function enablement value by using the **UPDATE IMSFUNC** command and IMS is cold started, one of the following situations occurs:

- For local functions, the enablement value is retrieved from the DFSDF*xxx* PROCLIB member during cold start. In this case, the **UPDATE IMSFUNC** command might need to be issued following the cold start to return the enablement value to a previous state.
- For global functions, if you use RM, CQS, and a resource structure, the enablement value is retrieved from the resource structure during cold start.
- For catalog functions, the enablement value is retrieved from the catalog during cold start.

Procedure

After installing a PTF that includes an IMS function that is not enabled by default, use one or more of the following steps to enable or disable the function:

- To enable IMS functions statically, define the enabling parameter for the function in the DFSDF*xxx* member, and then cold start IMS.
- To enable IMS functions dynamically, perform the following steps.

  a) Restart IMS.

  When IMS restarts, message DFS4878I is issued to display the current IMS function level.

  b) Issue the **UPDATE IMSFUNC** command with the **SET(ENABLED(Y))** option specified.

  For IMS catalog functions, the enablement value is stored in the IMS catalog after you issue the **UPDATE IMSFUNC** command.

  For all other global IMS functions, if you use RM, CQS, and a resource structure, the enablement value is stored in the resource structure.

c) Optional: For local functions, define the function as enabled in the DFSDF*xxx* member so that the function remains enabled at the next IMS cold start. For global functions, if you do not use RM, CQS, and a resource structure, define the function as enabled in the DFSDF*xxx* member so that the function remains enabled at the next IMS cold start.

- To disable the function statically, define the function as disabled in the DFSDF*xxx* member so that the function remains disabled at the next IMS cold start.
- To disable the function dynamically, perform the following steps:

  a) Issue the **UPDATE IMSFUNC** command with the **SET(ENABLED(N))** option specified.

  For IMS catalog functions, the enablement value is stored in the IMS catalog after you issue the **UPDATE IMSFUNC** command.

  For all other global IMS functions, if you use RM, CQS, and a resource structure, the enablement value is stored in the resource structure.

  b) Optional: For local functions, define the function as disabled in the DFSDF*xxx* member so that the function remains disabled at the next IMS cold start. For global functions, if you do not use RM, CQS, and a resource structure, define the function as disabled in the DFSDF*xxx* member so that the function remains disabled at the next IMS cold start.

Related concepts

IMS function levels overview (System Administration)

Related reference

QUERY IMSFUNC command (Commands)

UPDATE IMSFUNC command (Commands)

Controlling log data set characteristics

You might need to tune and modify log data set characteristics after monitoring or after changing your requirements for system availability, integrity, or operator handling. You can change characteristics for the OLDS, WADS, and RECON data sets through the IMS commands.

Changing online log data set characteristics

Because you can restart IMS (warm start or emergency restart) with all input on SLDS, you can reallocate the OLDSs between a shutdown (or failure) and a subsequent restart. To restart IMS using SLDSs as input, you must delete the PRIOLDS and SECOLDS records from the RECON data sets.

Changing block size for the OLDS

You can change the space, location, or block size of your OLDS or reallocate an OLDS on the same volume and with the same space.

Procedure

1. Shut down IMS.
2. Archive all OLDSs.
3. Delete PRIOLDS and SECOLDS records from the RECON data sets by using the **DELETE.LOG** command of the Data Base Recovery Control utility program (DSPURX00).
4. Either scratch all OLDSs and reallocate them with the BLKSIZE, or change the BLKSIZE= parameter in the LOGGER section of the DFSDF*xxx* PROCLIB member to the new block size.
5. Verify WADS space allocation.
6. Restart IMS (from SLDS).

What to do next

Changing the OLDS block size can affect the space required for the IMS write-ahead data set (WADS) . For example, if you have your WADS sized to hold a hundred 22K buffers, and you increase the OLDS block

size to 24K, you might need to increase the size of the WADS if you want it to be able to hold a hundred of the larger 24K buffers. If you reallocate the WADS, make sure that you restart IMS by using the **/NRE FORMAT WA** or **/NRE FORMAT ALL** command.

Related tasks

"Changing write-ahead data set characteristics" on page 44
The write-ahead data set (WADS) contains log records that reflect completed operations and are not yet written to an online log data set. You can change the write-ahead data set (WADS) mode, add or remove a WADS, or change the space, location, or allocation of the WADS.

Related reference

Execution data sets (Installation)

Changing the OLDS mode from single to dual

You must change your OLDSDEF specification in the LOGGER section of the DFSDF*xxx* member in IMS.PROCLIB to change the OLDS mode from single to dual. IMS initialization requires that at least three pairs of OLDSs be available. You must also reconsider data set placement.

About this task

When changing from single to dual OLDS, each data set in a pair of OLDSs must have the same space allocation (number of blocks).

Changing the mode from single to dual or from dual to single requires changes in the following operating procedures:

- Skeletal JCL for archive (ARCHJCL member)
- Skeletal JCL for log recovery (LOGCLJCL member)
- Batch JCL for Log Recovery utility
- Batch backout JCL for online transactions and BMPs

If you warm start the IMS subsystem after changing from single to dual OLDSs, the **/DISPLAY OLDS** command does not show the secondary OLDSs as IN USE until they have been archived once. The command does, however, show dual OLDS logging.

Procedure

1. Shut down IMS.
2. Archive all OLDSs.
3. Allocate dual OLDSs.
4. Delete the OLDS records from the RECON data sets, using the **DELETE.LOG** command. The primary OLDS records will be deleted.
5. Change OLDSDEF specification to dual.
6. Change IMS startup procedure (OLDS DD statements), if required.
7. Compile DFSMDA macros, if required.
8. Modify operating procedures.
9. Restart IMS.

Changing the OLDS mode from dual to single

You must change your OLDSDEF specification in the LOGGER section of the DFSDF*xxx* member in IMS.PROCLIB to change the OLDS mode from dual to single. IMS initialization requires that at least three pairs of OLDSs be available. You must also reconsider data set placement.

About this task

Changing the mode from single to dual or from dual to single requires changes in the following operating procedures:

- Skeletal JCL for archive (ARCHJCL member)
- Skeletal JCL for log recovery (LOGCLJCL member)
- Batch JCL for Log Recovery utility
- Batch backout JCL for online transactions and BMPs

Procedure

1. Shut down IMS.
2. Archive all OLDSs.
3. Allocate the single OLDS data set.
4. Delete the OLDS records from the RECON data sets, using the **DELETE.LOG** command.
5. Change OLDSDEF specification to single.
6. Change IMS startup procedure (OLDS DD statements), if required.
7. Compile DFSMDA macros, if required.
8. Modify operating procedures.
9. Restart IMS.

Changing the number of OLDS buffers

You must change BUFNO by changing the OLDSDEF specification for BUFNO in the LOGGER section of the DFSDFxxx member in IMS.PROCLIB to change the number of OLDS buffers.

About this task

When you modify BUFNO, you should consider also modifying the region size for the VSM common storage area (CSA). The amount of storage fixed for OLDS buffers is [BUFNO x BUFFERSIZE].

WADS space is also affected by BUFNO.

To change the number of OLDS buffers:

Procedure

1. Shut down IMS.
2. Change the OLDSDEF specification for BUFNO.
3. Verify CSA size.
4. Verify WADS space allocation.
5. Restart IMS.

Related tasks

“Changing write-ahead data set characteristics” on page 44

The write-ahead data set (WADS) contains log records that reflect completed operations and are not yet written to an online log data set. You can change the write-ahead data set (WADS) mode, add or remove a WADS, or change the space, location, or allocation of the WADS.

Related reference

Execution data sets (Installation)

Changing space, location, or allocation of an OLDS without shutting down IMS

You can change the space, location, or allocation of an OLDS by change the block size for the OLDS. If your IMS is operating in a non-XRF environment, you can modify space, location, or allocation without shutting down IMS.

Procedure

1. **/STOP OLDS***nn*
2. Archive all OLDSs.
3. Delete PRIOLDS and SECOLDS records in the RECON data sets, using the **DELETE.LOG** command.
4. Scratch and reallocate OLDSs.
5. **/START OLDS***nn*

Related concepts

Formatting newly initialized (reinitialized) volumes for an OLDS (System Definition)

Changing write-ahead data set characteristics

The write-ahead data set (WADS) contains log records that reflect completed operations and are not yet written to an online log data set. You can change the write-ahead data set (WADS) mode, add or remove a WADS, or change the space, location, or allocation of the WADS.

About this task

Related tasks

"Changing block size for the OLDS" on page 41
You can change the space, location, or block size of your OLDS or reallocate an OLDS on the same volume and with the same space.

"Changing the number of OLDS buffers" on page 43
You must change BUFNO by changing the OLDSDEF specification for BUFNO in the LOGGER section of the DFSDFxxx member in IMS.PROCLIB to change the number of OLDS buffers.

Changing the WADS mode from single to dual

You can change the write-ahead data set (WADS) mode from single to dual. The WADS can be dynamically allocated and deallocated.

Before you begin

To reflect the new mode for WADSs, you must update the following parameters:

- Skeletal JCL for the Log Recovery utility (LOGCLJCL member).
- LOGGER section of the DFSDFxxx member of the IMS.PROCLIB data set (WADSDEF statement).
- All recovery procedures implemented to recover WADS errors or to close unclosed OLDSs using WADS.

Procedure

1. Shut down IMS.
2. Allocate new WADS.

3. Define a DFSMDA member.
4. Add DD statement in IMS JCL, if necessary.
5. Update WADSDEF specification in the LOGGER section of the DFSDF*xxx* member of IMS.PROCLIB.
6. Modify operating procedures.
7. Restart IMS with FORMAT WADS keywords.

Changing the WADS mode from dual to single

You can change the write-ahead data set (WADS) mode from dual to single. The WADS can be dynamically allocated and deallocated.

Before you begin

To reflect the new mode for WADSs, you must update the following parameters:

- Skeletal JCL for the Log Recovery utility (LOGCLJCL member).
- LOGGER section of the DFSDF*xxx* member of the IMS.PROCLIB data set (WADSDEF statement).
- All recovery procedures implemented to recover WADS errors or to close unclosed OLDSs using WADS.

Procedure

1. Shut down IMS.
2. Update WADSDEF specification in the LOGGER section of the DFSDF*xxx* member of IMS.PROCLIB.
3. Delete DFSMDA member.
4. Remove DD statement in IMS JCL, if necessary.
5. Modify operating procedures.
6. Restart IMS.

Adding a spare WADS

The write-ahead data set (WADS) contains log records that reflect completed operations and are not yet written to an online log data set.

Procedure

1. Allocate a spare WADS.
2. Update WADSDEF specification in the LOGGER section of the DFSDF*xxx* member of IMS.PROCLIB
3. Define DFSMDA member.
4. Add DD statement in IMS JCL, if necessary.
5. Modify operating procedures.
6. **/START WADS***n* (or wait until IMS restart).

Removing a spare WADS

You might need to remove a spare write-ahead data set (WADS). WADS contains log records that reflect completed operations and are not yet written to an online log data set.

Procedure

1. Issue the **/STOP WADS***n* command (and wait for dynamic deallocation).
2. Scratch the spare WADS.
3. Update WADSDEF statement in the LOGGER section of the DFSDF*xxx* member of IMS.PROCLIB.
4. Remove DD statement in IMS JCL.
5. Modify operating procedures.

Changing space, location, or allocation of a WADS

All write-ahead data sets (WADS) must have the same space allocation (number of tracks) and be on the same type of device.

Procedure

1. Shut down IMS.
2. Scratch and reallocate WADSs.
3. Restart IMS with FORMAT WADS keywords.

What to do next

Changes in the WADS block size after IMS initialization are accomplished by changing the OLDS block size. If you are changing the OLDS block size from being a multiple of 2048 (running in non-z/Architecture mode) to being a multiple of 4096 (running in z/Architecture® mode), make sure that you run the **/NRE FORMAT WA** or **/NRE FORMAT ALL** command after changing the OLDS block size.

Changing system log data set characteristics

Converting from single to dual SLDSs requires modification in the skeletal JCL (ARCHJCL member) and in all your operational procedures using SLDSs. In the operational procedures, consider using the secondary SLDS when you experience errors in the primary one. No modification is required for online processing because IMS dynamically allocates SLDSs.

About this task

Changing the BLKSIZE requires modification in the skeletal JCL (ARCHJCL member). All SLDSs required for online processing must have the same BLKSIZE.

Changing RECON data set characteristics

You can change many aspects of the RECON data set by adding or removing spares, replacing the active data set, or changing it from single to dual mode.

Adding a spare RECON data set

IMS normally works with two active RECON data sets. If one RECON data set becomes unavailable, a spare RECON data set will be activated, if a spare is available.

About this task

Recommendation: For both online and batch, use dynamic allocation for RECON data sets, and run with at least three RECON data sets.

Note: The spare data set must be in VSAM CREATE mode.

Procedure

1. Delete the cluster.
2. Compile DFSMDA macro for the spare data set or add DD statement in IMS JCL and batch JCL if you do not use dynamic allocation.

Removing a spare RECON data set

There may be certain instances where you need to remove a RECON data set (for instance testing in a nonproduction system or maybe a damaged DASD).

About this task

Recommendation: For both online and batch, use dynamic allocation for RECON data sets, and run with at least three RECON data sets.

Procedure

1. Delete the cluster.
2. Delete DFSMDA member in IMS.SDFSRESL or remove DD statement from IMS JCL and batch JCL if you do not use dynamic allocation.

Replacing an active RECON data set

There might be circumstances for which you need to replace an active RECON data set, for instance on a test system.

Before you begin

Recommendations: Stop all IMS subsystems and batch jobs accessing the RECON data sets.

For both online and batch, use dynamic allocation for RECON data sets, and run with at least three RECON data sets.

Procedure

1. Define a spare data set with new space or allocate a spare data set at a new location.
2. Issue the **CHANGE.RECON REPLACE (RECON***n*) command.
3. Define a new spare data set.
4. Continue normal processing.

Changing the RECON data set mode from single to dual

IMS normally works with two active RECON data sets. If one RECON data set becomes unavailable, a spare will be activated if a spare is available.

Before you begin

Recommendations: Stop all IMS subsystems and batch jobs accessing the RECON data sets.

For both online and batch, use dynamic allocation for RECON data sets, and run with at least three RECON data sets.

Procedure

1. Define a spare RECON data set.
2. Issue the **CHANGE.RECON DUAL** command.
3. Define a new spare data set.
4. Continue normal processing.

Connecting and disconnecting subsystems

IMS can only connect to another subsystem if that subsystem is identified in the subsystem member in IMS.PROCLIB. You must specify the subsystem member name in the SSM EXEC parameter before an IMS

subsystem can access databases in an external subsystem (another program executing in a z/OS address space).

When specified to and accessed by IMS, the subsystem member name cannot be changed without stopping IMS.

Connections between an IMS subsystem and another subsystem can occur with or without operator intervention, and without regard to which subsystem is available first.

- Automatic connection: When the SSM EXEC parameter is specified, IMS automatically establishes the connection when it processes the parameter.
- Operator-controlled connection: When the SSM EXEC parameter is not specified, the connection is established when the **/START SUBSYS SSM** command is entered. The **/START SUBSYS SSM** command specifies the subsystem member in IMS.PROCLIB that IMS uses in connecting to subsystems.

Disconnecting IMS from another subsystem is initiated with the **/STOP SUBSYS** command, and normally completes without operator intervention. IMS will quiesce all dependent region external subsystem activity prior to stopping the subsystem.

If an external subsystem fails, IMS might not be able to disconnect properly from the failed subsystem. In this case, IMS might appear to hang in the disconnect process and the external subsystem might not be able to reconnect to IMS until the threads that were open at the time of the failure are properly closed.

To prevent this problem, issue the **/STOP SUBSYS** command before the failing external subsystem is shut down. After the external subsystem is brought back up, reconnecting to IMS is automatic.

If it is not possible to issue the **/STOP SUBSYS** command before the external subsystem is shut down, issue the **/DISPLAY ACTIVE** or **/DISPLAY ALL** command. These commands enable you to display the external subsystem threads that are still connected to IMS. Each thread has to be canceled individually. After all threads are cancelled and the external subsystem is brought back up, reconnecting to IMS is automatic.

Related concepts

Specifying the SSM= EXEC parameter (System Definition)

Commands for IMS operations tasks

You can use various commands to complete IMS operation tasks.

The following table describes commands for IMS operation tasks.

Table 15. Commands for IMS operations tasks

Operations task	Command
Activating VTAM nodes	/ACTIVATE
Activating MSC links	/ACTIVATE
Activating PSBs in select systems	IMPORT DEFN SOURCE(CATALOG)
Allocating a conversation with an LU name and TP name	/ALLOCATE
Altering assignments for IMS resources	/MSASSIGN
Altering relationships between IMS resources	/ASSIGN
Assigning a new logical link	/MSASSIGN MSNAME *name* LINK *link* UPDATE MSNAME SET(MSLINK(*name*))

Table 15. Commands for IMS operations tasks (continued)

Operations task	Command
Assigning a new physical link	/MSASSIGN LINK *link* MSPLINK *name* UPDATE MSLINK SET(MSPLINK(*name*))
Assigning a user structure to a nonexistent node	/OPNDST NODE x USER y
Beginning HALDB Online Reorganization for one or more partitions	INITIATE OLREORG /INITIATE OLREORG
Canceling an input message	/CANCEL
Canceling an output message	/DEQUEUE QUEUE
Changing applications	UPDATE PGM
Changing database buffers	UPDATE POOL
Changing databases	UPDATE DB
Changing default mode table name	UPDATE MSPLINK SET(MODETBL(*name*)) /CHANGE LINK *link* MODE *name* UPDATE MSLINK SET(MODETBL(*name*))
Changing HALDB Online Reorganization impact on overall system performance, for one or more partitions	UPDATE OLREORG SET(RATE(*rate*)) /UPDATE OLREORG SET(RATE(*rate*))
Changing IMS Fast DB Recovery surveillance	/CHANGE
Changing IMS resources	/CHANGE UPDATE
Changing Fast Path routing code definitions	UPDATE RTC
Changing local system identification (SIDs)	UPDATE MSNAME SET(SIDL(*id*))
Changing logical link name	UPDATE MSLINK SET(MSLINK(*name*))
Changing partner ID	UPDATE MSLINK SET(PARTNER(*id*))
Changing physical link name	UPDATE MSPLINK SET(MSPLINK(*name*))
Changing session resynchronization option	/CHANGE LINK *link* FORCSESS | SYNCSESS | COLDSESS UPDATE MSLINK SET(SYNCOPT(FORCESS | SYNCSESS | COLDSESS))
Changing transactions	UPDATE TRAN
Changing the address of the channel-to-channel adapter for a physical link	UPDATE MSPLINK SET(ADDR(*addr*))
Changing the input and output buffer sizes for each logical link that is assigned to a physical link.	UPDATE MSPLINK SET(BUFSIZE(*size*))
Changing the number of parallel sessions that can be active for TCP/IP and VTAM physical link types	UPDATE MSPLINK SET(SESSION(*n*))

Table 15. Commands for IMS operations tasks (continued)

Operations task	Command
Changing the order in which IMS restarts TCP/IP and VTAM links after an XRF takeover	UPDATE MSPLINK SET(BACKUP(*n*|N0))
Changing the RECON data set	/RMCHANGE
Changing the ready state for a terminal component	/COMPT /RCOMPT
Changing remote system identification SIDs	UPDATE MSNAME SET(SIDR(*id*))
Changing send or receive buffer size	UPDATE MSLINK SET(BUFSIZE(*size*))
Changing VTAM node name	UPDATE MSPLINK SET(NODE(*name*))
Checkpoint, taking	/CHECKPOINT
Class, assigning	/ASSIGN UPDATE TRAN SET(CLASS)
Cold start of components of IMS	/ERESTART
Cold start of IMS	/NRESTART
Connecting to a restarted IRLM	F job,RECONNECT
coupling facility structures, listing status	/CQQUERY
CQS checkpoint, initiating	/CQCHKPT
CQS structure checkpoint, requesting	/CQSET
Creating IMS resources online (application programs, databases, Fast Path routing codes, or transactions)	CREATE
Creating records in the RECON data set	/RMINIT
DBRC, commands for	/RMxxxxxx
Deallocating a user for an ISC node	/QUIESCE
Defining IMS resources online (application programs, databases, Fast Path routing codes, or transactions)	CREATE
Delivering asynchronous output from an LU 6.2 device	/ALLOCATE
Deleting affinities from VTAM	/CHECKPOINT
Deleting IMS resources	/CHANGE /CHECKPOINT DELETE
Deleting information in the RECON data set	/RMDELETE
Destination of messages, setting	/SET
Directing output to a component of a PTERM	/ASSIGN
Disabling logons to this physical link	/PSTOP MSPLINK *name* UPDATE MSPLINK STOP(LOGON)

Table 15. Commands for IMS operations tasks (continued)

Operations task	Command
Disconnecting a VTAM terminal	/CLSDST /RCLSDST
Displaying an LTERM	/DISPLAY LTERM *ltermname* QUERY LTERM NAME(*ltermname*)
Displaying a node	/DISPLAY NODE *nodename* QUERY NODE NAME(*nodename*)
Displaying a user	/DISPLAY USER *username* QUERY USER NAME(*username*)
Displaying a userid	/DISPLAY USER *useridname* QUERY USERID NAME(*useridname*)
Displaying assigned MSLINKS	/DISPLAY ASMT MSPLINK *name* QUERY MSPLINK SHOW(MSLINK) QUERY MSNAME SHOW(MSLINK)
Displaying assigned MSPLINKs	/DISPLAY ASMT LINK *link* MODE QUERY MSLINK SHOW(MSPLINK) QUERY MSNAME SHOW(MSPLINK)
Displaying assigned MSNAMEs	/DISPLAY ASMT LINK *link* QUERY MSLINK SHOW(MSNAME) QUERY MSPLINK SHOW(MSNAME)
Displaying bandwidth mode	/DISPLAY LINK OPTION BUFSIZE QUERY MSLINK SHOW(BANDWIDTH)
Displaying buffer size	/DISPLAY LINK OPTION BUFSIZE QUERY MSLINK SHOW(BUFSIZE)
Displaying CTC address	/DISPLAY ASMT MSPLINK *name* QUERY MSPLINK SHOW(ADDR)
Displaying database buffer pool usage statistics	QUERY POOL /DISPLAY POOL DBAS
Displaying usage statistics about storage pools managed by the IMS 64-bit storage manager	QUERY POOL TYPE(STG64) NAME(*name*)
Displaying global queue counts	/DISPLAY LINK *link* QCNT QUERY MSNAME SHOW(QCNT)
Displaying HALDB Online Reorganization status	QUERY DB STATUS(OLR) /DISPLAY DB OLR

Table 15. Commands for IMS operations tasks (continued)

Operations task	Command
Displaying IMS Fast DB Recovery status	F fdbrproc,STATUS
Displaying IMS resources and status	/DISPLAY QUERY
Displaying IMSplex member status or attribute information	QUERY MEMBER
Displaying IRLM status	F irlmproc,STATUS ,ALLD ,ALLI ,MAINT ,STOR ,TRACE
Displaying links with specified status	QUERY MSNAME STATUS(*status*)
Displaying MSNAMEs (logical link paths)	QUERY MSNAME QCNT(*condition*,*count*) QUERY MSLINK STATUS(*status*)
Displaying local queue counts	/DISPLAY LINK *link* QUERY MSNAME SHOW(QCNT)
Displaying local and remote system IDs	/DISPLAY ASMT MSNAME *msname* QUERY MSNAME SHOW(SYSID)
Displaying the LTERM name of the MTO	/RDISPLAY QUERY LTERM STATUS(*MTO,SMTO*)
Displaying maximum session count	/DISPLAY ASMT MSPLINK *name* QUERY MSPLINK SHOW(MSLINK)
Displaying members of the IMSplex	QUERY IMSPLEX
Displaying mode table name	/DISPLAY LINK *link* MODE [for MSC] /DISPLAY NODE *nodename* MODE [for non-MSC] QUERY MSLINK SHOW(MODETBL) QUERY NODE SHOW(MODETBL)
Displaying MSPLINKs with specified status	QUERY MSPLINK SHOW(MSNAME)
Displaying MSPLINKs of a specified type	QUERY MSPLINK TYPE(*type*)
Displaying OLCSTAT data set information	QUERY OLC LIBRARY (OLCSTAT)
Displaying partner ID	/DISPLAY LINK *link* MODE QUERY MSLINK SHOW(PARTNER)
Displaying physical link type	/DISPLAY ASMT MSPLINK *name* QUERY MSPLINK SHOW(TYPE)
Displaying send and receive counts	/DISPLAY LINK *link* /DISPLAY NODE *nodename* QUERY MSLINK SHOW(COUNT) QUERY NODE SHOW(COUNT)
Displaying status of assigned MSLINKs	QUERY MSPLINK SHOW(MSLINK)

Table 15. Commands for IMS operations tasks (continued)

Operations task	Command
Displaying status of a database or area	/DISPLAY DB /DISPLAY AREA QUERY DB QUERY AREA
Displaying status and rate information about HALDB Online Reorganizations that are in progress	QUERY OLREORG
Displaying status of MSLINKs	/DISPLAY LINK *link* /DISPLAY STATUS LINK QUERY MSLINK SHOW(STATUS)
Displaying status of MSNAMEs	/DISPLAY MSNAME *msname* /DISPLAY STATUS MSNAME QUERY MSNAME SHOW(STATUS)
Displaying status of MSPLINKs	/DISPLAY ASMT MSPLINK *name* QUERY MSPLINK SHOW(STATUS)
Displaying runtime resource and descriptor definitions that have not been exported to the IMSRSC repository since those resources and descriptors were created or last updated	QUERY DB SHOW(EXPORTNEEDED) QUERY DBDESC SHOW(EXPORTNEEDED) QUERY PGM SHOW(EXPORTNEEDED) QUERY PGMDESC SHOW(EXPORTNEEDED) QUERY RTC SHOW(EXPORTNEEDED) QUERY RTCDESC SHOW(EXPORTNEEDED) QUERY TRAN SHOW(EXPORTNEEDED) QUERY TRANDESC SHOW(EXPORTNEEDED)
Displaying VTAM node name	/DISPLAY ASMT MSPLINK *name* QUERY MSPLINK SHOW(NODE)
Dumping MSDBs	/DBDUMP
Enabling logons to this physical link	/RSTART MSPLINK *name* UPDATE MSPLINK START(LOGON)
Ending special modes	/END
Exclusive mode for a terminal, setting	/EXCLUSIVE
External subsystem, entering commands for	/SSR
Fast DB Recovery surveillance, changing	/CHANGE
Forcing stop link	/PSTOP LINK *link* PURGE|FORCE UPDATE MSLINK STOP(COMM) OPTION (FORCE)
Formatting a terminal screen	/FORMAT

Table 15. Commands for IMS operations tasks (continued)

Operations task	Command
Gathering diagnostic data while IMS is active	/DIAGNOSE
Generating JCL for utilities	/RMGENJCL
IMS monitor, starting or stopping	/TRACE
Initiating IMS Fast DB Recovery of tracked databases	F fdbrproc,RECOVER
Initiating a session with a terminal	/OPNDST
IRLM, abending	F irlmproc,ABEND
IRLM, changing MAXCSA (for IRLM Version 2.1)	F irlmproc,SET,CSA=nnn ,TRACE=nnn
IRLM, displaying status	F irlmproc,STATUS ,ALLD ,ALLI ,MAINT ,STOR ,TRACE
IRLM, reconnecting to	F proc, RECONNECT
IRLM, releasing locks	F irlmproc,PURGE,imsname
IRLM, setting timeout value	F irlmproc,START,TMOUT=n F irlmproc,STOP,TMOUT
IRLM, starting	S irlmproc
IRLM, stopping	P irlmproc
IRLM, tracing	S irlmproc,TRACE=YES TRACE CT
ISC node shutdown	/QUIESCE
ISC session cold start	/ASSIGN
Language Environment® (LE) attributes and parameters, changing	UPDATE LE DELETE LE
Limiting count for a transaction, setting	/ASSIGN UPDATE TRAN SET(LCT)
Listing all of the databases for which HALDB Online Reorganizations are in progress	QUERY DB STATUS(OLR)
Listing information in the RECON data set	/RMLIST
Logging off from a terminal	/RCLSDST
Logging on to IMS from a VTAM terminal	/OPNDST
LTERM, enqueuing messages to or dequeuing messages from	QUEUE LTERM
LTERM to node, assigning	/ASSIGN
LTERM to PTERM, assigning	/ASSIGN
LTERM to user or ISC half session, assigning	/ASSIGN
Message control error exit routine, activating	/DEQUEUE
Message Format Service (MFS) format, using on a terminal	/FORMAT
MFS test mode for terminals, setting	/TEST

Table 15. Commands for IMS operations tasks (continued)

Operations task	Command
Modifying applications	/MODIFY INITIATE UPDATE PGM
Modifying databases	/MODIFY INITIATE UPDATE DB
Modifying IMS resources	INITIATE /MODIFY UPDATE
Modifying IMS resources globally for an IMSplex	INITIATE OLC
Modifying Fast Path routing code definitions	/MODIFY INITIATE UPDATE RTC
Modifying transactions	UPDATE TRAN
Monitoring and displaying the status of the specified databases or partitions (including those HALDB Online Reorganizations that are in progress)	/DISPLAY DB OLR
MTO, displaying LTERM name, line, and node	/RDISPLAY QUERY LTERM STATUS(*mto,smto*) SHOW(*node*)
Notifying DBRC about additional information, which gets recorded in the RECON data set	/RMNOTIFY
Online change, performing	INITIATE /MODIFY
Output errors, testing for	/LOOPTEST
Output segments for application program, assigning	/ASSIGN
Output to secondary MTO, sending	/SMCOPY
Preventing updates to a database	/DBDUMP /DBRECOVERY UPDATE AREA STOP(ACCESS) UPDATE DATAGRP STOP(ACCESS) UPDATE DB STOP(UPDATES) UPDATE DB STOP(ACCESS)
Priority for a transaction, assigning	/ASSIGN UPDATE TRAN SET(NPRI)
Querying a logical terminal (LTERM)	/DISPLAY LTERM *ltermname* QUERY LTERM NAME(*ltermname*)
Querying a node	/DISPLAY NODE *nodename* QUERY NODE NAME(*nodename*)

Table 15. Commands for IMS operations tasks (continued)

Operations task	Command
Querying a user	/DISPLAY USER *username* QUERY USER NAME(*username*)
Querying a user ID	/DISPLAY USER *useridname* QUERY USERID NAME(*useridname*)
Querying assigned Multiple Systems Coupling (MSC) logical links (MSLINKs)	/DISPLAY ASMT MSPLINK *name* QUERY MSPLINK SHOW(MSLINK) QUERY MSNAME SHOW(MSLINK)
Querying assigned Multiple Systems Coupling (MSC) physical links (MSPLINKs)	/DISPLAY ASMT LINK *link* MODE QUERY MSLINK SHOW(MSPLINK) QUERY MSNAME SHOW(MSPLINK)
Querying assigned MSNAMEs	/DISPLAY ASMT LINK *link* QUERY MSLINK SHOW(MSNAME) QUERY MSPLINK SHOW(MSNAME)
Querying bandwidth mode	/DISPLAY LINK OPTION BUFSIZE QUERY MSLINK SHOW(BANDWIDTH)
Querying buffer size	/DISPLAY LINK OPTION BUFSIZE QUERY MSLINK SHOW(BUFSIZE)
Querying CTC address	/DISPLAY ASMT MSPLINK *name* QUERY MSPLINK SHOW(ADDR)
Querying global queue counts	/DISPLAY LINK *link* QCNT QUERY MSNAME SHOW(QCNT)
Querying HALDB Online Reorganization status	QUERY DB STATUS(OLR) /DISPLAY DB OLR
Querying IMS Fast DB Recovery status	F fdbrproc,STATUS
Querying IMS resources and status	/DISPLAY QUERY
Querying IMSplex member status or attribute information	QUERY MEMBER
Querying IRLM status	F irlmproc,STATUS ,ALLD ,ALLI ,MAINT ,STOR ,TRACE
Querying links with specified status	QUERY MSNAME STATUS(*status*)
Querying MSNAMEs (logical link paths)	QUERY MSNAME QCNT(*condition*,*count*) QUERY MSLINK STATUS(*status*)

Table 15. Commands for IMS operations tasks (continued)

Operations task	Command
Querying local queue counts	/DISPLAY LINK *link* QUERY MSNAME SHOW(QCNT)
Querying local and remote system IDs	/DISPLAY ASMT MSNAME *msname* QUERY MSNAME SHOW(SYSID)
Querying the LTERM name of the MTO	/RDISPLAY QUERY LTERM STATUS(*MTO,SMTO*)
Querying maximum session count	/DISPLAY ASMT MSPLINK *name* QUERY MSPLINK SHOW(MSLINK)
Querying members of the IMSplex	QUERY IMSPLEX
Querying mode table name	/DISPLAY LINK *link* MODE [for MSC] /DISPLAY NODE *nodename* MODE [for non-MSC] QUERY MSLINK SHOW(MODETBL) QUERY NODE SHOW(MODETBL)
Querying MSPLINKs with specified status	QUERY MSPLINK SHOW(MSNAME)
Querying MSPLINKs of a specified type	QUERY MSPLINK TYPE(*type*)
Querying OLCSTAT data set information	QUERY OLC LIBRARY (OLCSTAT)
Querying partner ID	/DISPLAY LINK *link* MODE QUERY MSLINK SHOW(PARTNER)
Querying physical link type	/DISPLAY ASMT MSPLINK *name* QUERY MSPLINK SHOW(TYPE)
Querying send and receive counts	/DISPLAY LINK *link* /DISPLAY NODE *nodename* QUERY MSLINK SHOW(COUNT) QUERY NODE SHOW(COUNT)
Querying status of assigned MSLINKs	QUERY MSPLINK SHOW(MSLINK)
Querying status of a database or area	/DISPLAY DB /DISPLAY AREA QUERY DB QUERY AREA
Querying status and rate information about HALDB Online Reorganizations that are in progress	QUERY OLREORG

Table 15. Commands for IMS operations tasks (continued)

Operations task	Command
Querying status of MSLINKs	/DISPLAY LINK *link* /DISPLAY STATUS LINK QUERY MSLINK SHOW(STATUS)
Querying status of MSNAMEs	/DISPLAY MSNAME *msname* /DISPLAY STATUS MSNAME QUERY MSNAME SHOW(STATUS)
Querying status of MSPLINKs	/DISPLAY ASMT MSPLINK *name* QUERY MSPLINK SHOW(STATUS)
Querying VTAM node name	/DISPLAY ASMT MSPLINK *name* QUERY MSPLINK SHOW(NODE)
Queuing or dequeuing messages	/DEQUEUE QUEUE
RACF, controlling	/SECURE
Reconnecting to a restarted IRLM	F job,RECONNECT
Reconnecting to coupling facility structures	F job,RECONNSTR
Recovering databases by using the Database Recovery Facility	/RECOVER
Releasing IRLM locks	F irlmproc,PURGE,imsname
Remote takeover, initiating	/RTAKEOVER
Removing IMS from a VTAM generic resources group	/CHECKPOINT /STOP
Removing an IMS from the IMSplex	/CHE FREEZE or DUMPQ or PURGE LEAVEPLEX
Resetting values set during IMS system definition	/ASSIGN UPDATE
Restarting a component of IMS	/OPNDST /RELEASE /RSTART /START /UNLOCK UPDATE TRAN START(Q,SCHD,TRACE) UPDATE AREA START(ACCESS) UPDATE DATAGRP START(ACCESS) UPDATE DB START(ACCESS) UPDATE DB START(ACCESS) AREA(*) UPDATE DB SET(LOCK(OFF))

Table 15. Commands for IMS operations tasks (continued)

Operations task	Command
Restarting IMS after failure	/ERESTART
Restarting XRF alternate subsystem	/ERESTART
Resuming a conversation	/RELEASE
Resuming HALDB Online Reorganization for one or more partitions	INITIATE OLREORG /INITIATE OLREORG
Scheduling a suspended transaction	/DEQUEUE UPDATE TRAN START(SUSPEND)
Secondary MTO, sending output to	/SMCOPY
Security, controlling	/SECURE
Sending messages to terminals or other IMS subsystems	/BROADCAST LTERM QUEUE LTERM
Setting the RATE for a HALDB Online Reorganization	INITIATE OLREORG SET(RATE(*rate*)) /INITIATE OLREORG SET(RATE(*rate*)) UPDATE OLREORG SET(RATE(*rate*)) /UPDATE OLREORG SET(RATE(*rate*))
Setting or resetting ASR	/CHANGE LINK *link* ASR ON|OFF UPDATE MSLINK SET(ASR(ON|OFF)) UPDATE MSPLINK SET(ASR(ON|OFF))
Setting timeout value for IRLM	F irlmproc,START,TMOUT=n F irlmproc,STOP,TMOUT
Sharing startup JCL between IMS dependent regions	/START
Shutting down Common Queue Server (CQS)	P cqsproc
Shutting down the Common Service Layer (CSL)	F sciproc, SHUTDOWN CSLLCL/CSLPLEX
Shutting down IMS	/CHECKPOINT
Shutting down an ISC node	/QUIESCE
Shutting down Operations Manager (OM)	P omjob
Shutting down Resource Manager (RM)	P rmjob
Shutting down Structured Call Interface (SCI)	P scijob
Signing on or off IMS terminals	/SIGN
Specifying that a HALDB master is capable of being reorganized online	INIT.DB OLRCAP CHANGE.DB OLRCAP
Specifying whether to delete the inactive data sets after the copying phase of a HALDB online reorganization completes	UPDATE OLREORG OPTION(DEL | NODEL) /UPDATE OLREORG OPTION(DEL | NODEL)

Table 15. Commands for IMS operations tasks (continued)

Operations task	Command
Specifying whether to release ownership of OLR at normal or abnormal shutdown of IMS	UPDATE OLREORG OPTION(REL | NOREL) /UPDATE OLREORG OPTION(REL | NOREL)
Starting bandwidth mode	UPDATE MSLINK SET(BANDWIDTH(ON|OFF))
Starting Common Queue Server (CQS)	S cqsproc
Starting HALDB Online Reorganization for one or more partitions	INITIATE OLREORG /INITIATE OLREORG
Starting IMS	/NRESTART /ERESTART
Starting IMS Fast DB Recovery	S fdbrproc
Starting IMS resources	/START UPDATE TRAN START(Q,SCHD,TRACE) UPDATE AREA START(ACCESS) UPDATE DATAGRP START(ACCESS) UPDATE DB START(ACCESS) UPDATE DB START(ACCESS) AREA(*)
Starting an internal trace	/TRACE SET ON|OFF LINK *link* UPDATE MSLINK START(TRACE)
Starting IRLM	S irlmproc
Starting a line, link, node, PTERM, or user	/RSTART UPDATE MSLINK(*linkname*) START(COMM)
Starting msname queuing or sending	/START MSNAME *name* UPDATE MSNAME START(Q,SEND))
Starting an XRF takeover trace	/TRACE SET ON|OFF LINK *link* TAKEOVER UPDATE MSLINK START(TKOTRC)
Statistics, recording to IMS log	/CHECKPOINT
Stopping bandwidth mode	UPDATE MSLINK SET(BANDWIDTH(ON|OFF))
Stopping a line, a logical link, or a logical terminal	/PSTOP UPDATE MSLINK STOP(COMM)
Stopping database access	/LOCK /STOP UPDATE AREA STOP(SCHD) UPDATE DATAGRP STOP(SCHD) UPDATE DB SET(LOCK(ON)) UPDATE DB STOP(SCHD)
Stopping HALDB Online Reorganization for one or more partitions	TERMINATE OLREORG /TERMINATE OLREORG

Table 15. Commands for IMS operations tasks (continued)

Operations task	Command
Stopping IMS resources	/STOP UPDATE AREA STOP(SCHD) UPDATE DATAGRP STOP(SCHD) UPDATE DB STOP(SCHD) UPDATE TRAN STOP(Q,SCHD)
Stopping input to a line, link, or terminal	/PURGE UPDATE MSNAME STOP(Q),START(SEND)
Stopping input messages with a transaction code	/PURGE UPDATE TRAN START(SCHD)STOP(Q)
Stopping IMS Fast DB Recovery	F fdbrproc,DUMP F fdbrproc,STOP
Stopping IMS Fast DB Recovery tracking	F fdbrproc,TERM
Stopping input for msname	/PURGE MSNAME *name* UPDATE MSNAME STOP(Q) START(SEND)
Stopping an internal trace	/TRACE SET ON|OFF LINK *link* UPDATE MSLINK STOP(TRACE)
Stopping IRLM	P irlmproc
Stopping an LTERM, node, program, PTERM, or transaction	/LOCK SET(LOCK(ON))
Stopping a transaction	/STOP UPDATE TRAN STOP(Q,SCHD)
Stopping msname queuing or sending	/STOP MSNAME *name* UPDATE MSNAME STOP(Q,SEND)
Stopping output to a line, link, or terminal	/PSTOP UPDATE MSLINK STOP(COMM)
Stopping output for msname	UPDATE MSNAME START(Q) STOP(SEND)
Stopping output to a programmable remote station	/MONITOR
Stopping scheduling of a message with a transaction code	/PSTOP UPDATE TRAN STOP(SCHD) START(Q)
Stopping an XRF takeover trace	/TRACE SET ON|OFF LINK *link* TAKEOVER UPDATE MSLINK STOP(TKOTRC)
Subsystem, external, entering commands for	/SSR
Suspending a conversation	/HOLD
Switching data sets for XRF	/SWITCH
Terminating a conversation	/EXIT

Table 15. Commands for IMS operations tasks (continued)

Operations task	Command
Terminating global online change on all IMS systems in an IMSplex	TERMINATE OLC
Terminating IMS without a dump	F job,STOP F job,STOPxxx F job,FORCExxx
Terminating IMS with a dump	F job,DUMP F job,DUMPxxx
Terminating IRLM abnormally	F irlmproc,ABEND
Terminating I/O for terminals	/IDLE
Terminating preset mode	/RESET
Terminating special modes	/END
Test mode for terminals, setting	/TEST
Testing for output errors	/LOOPTEST
Tracing IMS resources	/TRACE UPDATE TRAN START(TRACE) UPDATE MSLINK START(TRACE)
Tracing IRLM	S irlmproc,TRACE=YES TRACE CT
Transactions, enqueuing messages to or dequeuing messages from	QUEUE TRAN
Undoing /ASSIGN	/ASSIGN /NRE CHECKPOINT 0 UPDATE TRAN
Undoing /EXCLUSIVE	/END /START
Undoing /HOLD	/RELEASE /EXIT
Undoing /IDLE	/ACTIVATE
Undoing /LOCK	/UNLOCK UPDATE DB SET(LOCK(OFF))
Undoing /LOOPTEST	/END
Undoing /MSASSIGN	/MSASSIGN /NRE CHECKPOINT 0 UPDATE MSLINK NAME(*linkname*) SET(MSPLINK(*plinkname*) UPDATE MSNAME NAME(*msname*) SET(MSLINK(*linkname*)

Table 15. Commands for IMS operations tasks (continued)

Operations task	Command
Undoing /SET	/RESET
Undoing /START, /RSTART, /PSTART, /PURGE, or /MONITOR	/STOP UPDATE
Undoing /START, /STOP, /COMPT, /RCOMPT, /PSTART, /PURGE, or /MONITOR	/RSTART UPDATE
Undoing /START, /MONITOR, /RSTART, /STOP, or /PSTART	/PURGE UPDATE
Undoing /START, /MONITOR, /RSTART, /STOP, or /PURGE	/PSTOP UPDATE
Undoing /START, /RSTART, /STOP, /PSTOP, or /PURGE	/MONITOR UPDATE
Undoing /TEST	/END /START
Unloading an area from a data space	/VUNLOAD
Verifying local transactions and LTERMs with remote transactions and LTERMs	/MSVERIFY
Verifying /MSASSIGN	/MSVERIFY
Warm start of IMS	/NRESTART
Writing a message to the IMS log	/LOG
XRF, switching data sets	/SWITCH

Important: In an IMSplex, type-2 commands are issued to the Operations Manager, which then routes the commands to the subsystem IMSplex members. When commands are issued globally in an IMSplex, they can behave differently from commands that are issued to single IMS systems.

Related concepts

Restrictions for HALDB Online Reorganization (Database Administration)

Related reference

Equivalent IMS type-1 and type-2 commands (Commands)

“List of commands with similar functions for multiple resources” on page 14
These tables show the IMS commands and how they affect certain resources.

Chapter 2. Starting or restarting IMS

Starting or restarting an IMS subsystem means initializing the IMS control region. You must first restore the operating environment and the master terminal operator must insure that critical applications, such as any dependent regions, are initialized.

About this task

To restore the operating environment, the MTO must ensure the following are also initialized:

- All IRLMs
- All CQSs
- Any dependent regions (including utilities)
- All CSL address spaces (ODBM, OM, RM, and SCI), if operating as an IMSplex using CSL
- Any connections to VTAM
- All communication lines and terminals

The term MTO used in this information refers to the IMS master terminal operator for the DB/DC and DCCTL environments, and to the operator for the DBCTL environment. All commands shown assume that the command-recognition character is the forward slash (/).

Starting an IMSplex

Starting an IMSplex involves starting required CSL components, such as SCI and OM, starting the IMS control regions, and starting any other optional IMSplex components, such as CSL RM, CSL ODBM, IMS Connect, and so on.

About this task

The following briefly summarizes starting an IMSplex:

Procedure

1. Start all SCIs.
2. Start all OMs.
3. If using a resource structure, start CQS.
4. If using RM, start all RMs.
5. Start all IMS control regions with the following parameters:
   - CSLG= parameter specified on the control region execution or the DFSPBxxx member of the IMS PROCLIB data set.
   - Parameters specified in the DFSDCxxx member of the IMS PROCLIB data set if you do not want to use the defaults:
     - RCVYSTSN
     - RCVYFP
     - SRMDEF
   - OCMD and CSLT parameters specified in the DFSVSMxx member of the IMS PROCLIB data set.
   - Parameters specified in the DFSCGxxx member of the IMS PROCLIB data set:
     - CMDSEC
     - IMSPLEX
     - NORSCCC

  - OLC
  - OLCSTAT
- Appropriate parameters specified in the BPE and CQS PROCLIB member data sets.

6. Start any other optional IMSplex components, such as ODBM and IMS Connect.
7. If you use a SPOC, you can log on to TSO, allocate the appropriate files to the TSO user ID, and issue the DFSSPOC command from ISPF option 6.

Starting the CSL

The Common Service Layer (CSL) is consisted of multiple address spaces. The CSL is considered "started" after each of the CSL manager address spaces you require is started.

If you are running IMS™ as an IMSplex with CSL, start the SCI, OM, and RM address spaces before starting the IMS control region. Start ODBM after the SCI startup procedure is completed. You can start the CSL manager address spaces by using the z/OS **START** command from a z/OS system console, JCL, or a procedure that is in another IMS control region.

Initialization parameters must be defined for each address space before CSL can be started. To customize the parameters of a specific CSL manager address space, use the following members of the PROCLIB data set:

- CSLDIxxx, which initializes the Open Database Manager
- CSLOIxxx, which initializes the Operations Manager
- CSLRIxxx, which initializes the Resource Manager
- CSLSIxxx, which initializes the Structured Call Interface

The parameters that are defined in the PROCLIB data set vary on the address space that you are customizing, but can include the IMSplex name, the address space name, the use of Automatic Restart Manager (ARM), and more.

For ODBM, IMS also provides a CSLDCxxx member of the IMS.PROCLIB data set, which contains parameters that define the IMS data store connections to ODBM.

ODBM, OM, RM, and SCI are initially started by running their respective startup procedures. To run an ODBM, OM, RM, or SCI procedure, you can either issue a z/OS **START** command or code and run JCL. ODBM, OM, RM, and SCI can be restarted using the z/OS Automatic Restart Manager (ARM). You can specify the parameter ARMRST either in the startup procedures or in the execution parameters of ODBM, OM, RM, and SCI. The default is ARMRST=Y, so if you do not specify the parameter, all of the CSL managers try to register with ARM.

Before you can start any of the CSL components, they must be defined to the IMS system.

The IMS PROCLIB data set members DFSCGxxx and DFSDFxxx define how the IMS control region interacts with the CSL.

Related concepts

CSL definition and tailoring (System Definition)

Related reference

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

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

Starting the CSL SCI

The system operator can start the CSL SCI in one of three ways: as a z/OS started procedure, as JCL, or as a started procedure in an IMS control region.

About this task

You can start SCI in the following ways:

Procedure

- The system operator can use a z/OS started procedure. CSLSC000, a sample started procedure that is shipped with IMS, can be modified for your installation, or you can identify your own started procedure. To start an SCI address space with a started procedure, issue the z/OS **START** command as follows: **S scijobname**.scijobname is the job name of the SCI address space to be started.
- The system operator can submit JCL that starts the started procedure.
- The IMS control region can use a started procedure by using the **SCIPROC** parameter in the DFSCG*xxx* or DFSDF*xxx* member of the IMS PROCLIB data sets. If there is no SCI address space active in the IMSplex, IMS will start the specified procedure.

Restarting the CSL SCI

After the Structured Call Interface (SCI) is started, if it is abnormally terminated, it can be restarted by using the z/OS Automatic Restart Manager (ARM). SCI must complete initialization for ARM to restart the address space if an abend occurs. Use of ARM to restart SCI is the default.

About this task

Starting the CSL ODBM

You can start ODBM either before or after starting IMS, IMS Connect, or z/OS Resource Recovery Services (RRS).

About this task

If ODBM and an IMS system are not members of the same IMSplex, ODBM automatically connects to the IMS system only if the IMS system is started before ODBM.

To enable ODBM to connect to the IMS system automatically regardless of the order in which ODBM and IMS are started, use either the DFSCGxxx member of the IMS.PROCLIB data set or the Common Service Layer section of the DFSDFxxx PROCLIB member to define the IMS system as a member of the same IMSplex as ODBM.

The options for starting ODBM include:

- The system operator can use a z/OS started procedure. For example, CSLODBM, a sample started procedure that is shipped with IMS, can be modified for your installation, or you can identify your own started procedure. To start an ODBM address space with a started procedure, issue the z/OS **START** command as follows:

  ```
  S odbmjobname
  ```

  odbmjobname is the job name of the ODBM address space to be started.
- The system operator can submit JCL that starts the started procedure.

Restarting the CSL ODBM

In the event that ODBM terminates abnormally, it can be restarted by using the z/OS Automatic Restart Manager (ARM).

About this task

ODBM must complete initialization for ARM to restart the address space if an abend occurs. Use of ARM to restart ODBM is the default.

Starting the CSL OM

The system operator can start the CSL OM three ways: using a z/OS started procedure, or by submitting JCL, or by using a started procedure in an IMS control region.

About this task

You can start OM in the following ways:

Procedure

- The system operator can use a z/OS started procedure. For example, CSLOM000, a sample started procedure that is shipped with IMS, can be modified for your installation, or you can identify your own started procedure. To start an OM address space with a started procedure, issue the z/OS **START** command as follows: **S omjobname**

  omjobname is the job name of the OM address space to be started.
- The system operator can submit JCL that starts the started procedure.
- The IMS control region can use a started procedure by using the **OMPROC** parameter in the DFSCG*xxx* or DFSDF*xxx* member of the IMS PROCLIB data sets. If there is no OM address space active in the IMSplex, IMS will start the specified procedure.

Restarting the CSL OM

If, after OM is started, if it is abnormally terminated, it can be restarted using the z/OS Automatic Restart Manager (ARM). OM must complete initialization for ARM to restart the address space if an abend occurs. Use of ARM to restart OM is the default.

About this task

Starting the CSL RM

The system operator can use a z/OS started procedure. CSLRM000, a sample started procedure that is shipped with IMS, can be modified for your installation, or you can identify your own started procedure.

About this task

You can start RM in the following ways:

- The system operator can use a z/OS started procedure. CSLSC000, a sample started procedure that is shipped with IMS, can be modified for your installation, or you can identify your own started procedure. To start an RM address space with a started procedure, issue the z/OS **START** command as follows:

  ```
  S rmjobname
  ```

  rmjobname is the job name of the RM address space to be started.
- The system operator can submit JCL that starts the started procedure.
- The IMS control region can use a started procedure by using the **RMPROC=** parameter in the DFSCG*xxx* or DFSDF*xxx* member of the IMS PROCLIB data sets. If there is no RM address space active in the IMSplex, IMS will start the specified procedure.

You can also add a resource structure to your CFRM policy.

Related tasks

Defining the CFRM policy (System Administration)

Restarting the CSL RM

If the resource manager (RM) terminates abnormally, it can be restarted using the z/OS Automatic Restart Manager (ARM). RM must complete initialization for ARM to restart the address space if an abend occurs. Use of ARM to restart RM is the default.

About this task

Starting and stopping the IMSRSC repository

An IMSRSC repository can have a status of closed, opening, open, stopping, or stopped. Start or stop a repository to make it available or unavailable for client connections.

CLOSED
The repository is defined to the Repository Server (RS) and it is started, but its associated data sets are closed. The RS accepts client connections to the repository, but because the repository has not yet been opened and its associated data sets have not been allocated by the RS, you cannot write or read from the repository in this state.

If a repository remains in CLOSED state at the end of SPARE recovery processing, perform the following steps to make the repository available:

1. Re-allocate the discarded repository data sets.
2. Issue the **LIST** FRPBATCH command to check if the repository is in a STOP state. If it is not, issue the **STOP** FRPBATCH command to change the state to STOP.
3. Issue the **DSCHANGE** FRPBATCH command to change the re-allocated discarded data sets back to a SPARE state.
4. Issue the **START** FRPBATCH command to start the repository.

   This resumes the data set recovery process.

OPENING
The repository is in the process of being opened.

OPEN
Started and opened. The repository is available for client connections and read and update requests.

Starting a repository makes it possible for connections to be made to it. The following functions are performed when a repository is started:

- The repository becomes available for client connections.
- If the repository is set with the **ADD** or **UPDATE** FRPBATCH commands to AUTOOPEN=Y, or the repository was started with the OPEN=YES option in the **START** FRPBATCH command, the RS opens the repository, readying it for client function requests.
- If the repository is not set to open either explicitly or automatically, the repository and RS catalog repository data sets are allocated when a client first connects to the repository.

STOPPING
The repository is in the process of being stopped.

Stopping a repository has the following results:

- The repository can no longer accept new client connections.
- If the repository data sets are allocated, the RS tries to close and deallocate them. The RS waits until all active client request threads are completed before closing and deallocating the data sets.

STOPPED
Stopped and closed. The RS does not accept client connections to the repository. The repository data sets are closed and deallocated.

The start and stop state of a repository is persistent. For example, a repository that is stopped remains stopped after the RS is restarted, regardless of the setting for the AUTOOPEN option. The AUTOOPEN option only determines when data sets are allocated and opened if the repository is started.

Related concepts

Overview of the IMSRSC repository (System Definition)

IMSRSC repository and RS catalog repository data sets (System Definition)

Related information

Commands for FRPBATCH (System Programming APIs)

Opening the IMSRSC repository

You can control when an IMSRSC repository is opened with the AUTOOPEN option for the **ADD** or **UPDATE** FRPBATCH commands, or with the OPEN option for the **START** FRPBATCH command.

- If the repository is set to AUTOOPEN=Y, its data sets are allocated and opened when it starts. If the repository was in the started state when the Repository Server (RS) last shut down, the data sets are allocated and opened when the RS is restarted.
- If the repository is set to AUTOOPEN=N and it is started with OPEN=YES, it is opened immediately.
- If the repository is set with AUTOOPEN= N and it is not started with OPEN=YES, it is opened at the time of the first client connection.

The RS performs basic validation of IMSRSC repository data sets at open time. A repository cannot be opened unless there is both a valid COPY1 and COPY2 of the IMSRSC repository data set.

If there is one valid copy and a valid SPARE IMSRSC repository data set, automated SPARE recovery occurs. In the SPARE recovery process, the SPARE IMSRSC repository data set is reassigned to take the place of the missing COPY1 or COPY2, and its data sets are populated from the valid copy.

At the conclusion of data set validation:

- The available IMSRSC repository data sets are allocated and open.
- Empty data sets are identified.
- All data sets have valid characteristics for their identified function as either a repository member data set (RMD) or repository index data set (RID).
- Populated data sets (with resource definitions) are considered to be valid repository data sets.
- The repository's set of data sets is known to be consistent.

If any data set fails to meet these basic requirements, an error message is issued and the repository open process terminates.

During repository open processing, either with the AUTOOPEN option or during the first access, all of the repository data sets are opened using VSAM local shared resources (LSR). As the data sets are opened using VSAM LSR, if the repository data sets are empty, informational messages are issued that indicate that the data sets are empty. An informational messages received for the spare data set pair requires no user action. The spare data set is opened to ensure that it is empty, so that there are no errors during SPARE recovery process.

The RS closes the IMSRSC repository data sets when the repository is stopped, during the SPARE recovery process and during RS shutdown.

Related concepts

Overview of the IMSRSC repository (System Definition)

IMSRSC repository and RS catalog repository data sets (System Definition)

Recovery in an IMSplex (System Administration)

Related information

Commands for FRPBATCH (System Programming APIs)

Starting the IMSRSC repository

To start the IMSRSC repository, you can either use a **START** command issued from the z/OS MODIFY (F) interface or issue **START** FRPBATCH command.

About this task

To start the repository, do one of the following:

- Issue the following command from the z/OS MODIFY (F) interface:

```
F reposervername,ADMIN START repositoryname
```

- Issue the **START** FRPBATCH command.

Related concepts

Overview of the IMSRSC repository (System Definition)

Related reference

F reposervername,ADMIN (Commands)

START command for FRPBATCH (System Programming APIs)

Stopping the IMSRSC repository

To stop the IMSRSC repository, you can either use a **STOP** command issued from the z/OS MODIFY (F) interface or issue **STOP** FRPBATCH command.

About this task

To stop the repository, do one of the following:

- Issue the following command from the z/OS MODIFY (F) interface:

```
F reposervername,ADMIN STOP repositoryname
```

- Issue the **STOP** FRPBATCH command.

Related concepts

Overview of the IMSRSC repository (System Definition)

Related reference

F reposervername,ADMIN (Commands)

STOP command for FRPBATCH (System Programming APIs)

Starting and stopping the Repository Server

To start the Repository Server (RS), submit a customized RS startup procedure. To stop the RS, use an RS command issued from the z/OS MODIFY (F) interface.

Related concepts

Overview of the IMSRSC repository (System Definition)

Related reference

F reposervername,ADMIN (Commands)

Starting the Repository Server

To start the Repository Server (RS), submit a customized RS startup procedure.

About this task

Perform the following procedure:

Procedure

1. Make sure that you have performed system definition for the IMSRSC repository.
2. Customize the following startup procedure:

```
//FRP12A   PROC RGN=0M,SOUT=A,
//*
//IEFPROC  EXEC PGM=BPEINI00,REGION=0M,
//  PARM='BPEINIT=FRPINI00,BPECFG=BPECONFG,FRPCFG=FRPCONFG'
//*
//STEPLIB  DD  DSN=RESLIB_dataset_name,DISP=SHR
//*
//PROCLIB  DD  DSN=PROCLIB_dataset_name,DISP=SHR
//*
//FRPPRINT DD  SYSOUT=&SOUT
//SYSPRINT DD  SYSOUT=&SOUT
//SYSUDUMP DD  SYSOUT=&SOUT
//*
```

   Make sure that the values specified in this JCL for the BPECFG= and FRPCFG= parameters match the values specified in the FRP and BPE configuration members of the IMS PROCLIB data set.
3. Submit the RS startup procedure.

Results

If the RS is started successfully as the active RS, then the FRP2002I message is issued.

If another RS is already started in the same z/OS cross-system coupling facility group, a FRP2001I message is issued, indicating that the subordinate RS is started.

During startup of both the active and subordinate RS, the FRP2006I message is issued.

When the RS has started successfully and is ready to accept client connections, the FRP2025I message is issued.

Related concepts

Overview of the IMSRSC repository (System Definition)

Related tasks

“Starting the Repository Server” on page 71
To start the Repository Server (RS), submit a customized RS startup procedure.

Defining the IMSRSC repository (System Definition)

“Cold starting an IMS system that uses the IMSRSC repository” on page 78
If IMS is using the IMSRSC repository, and AUTOIMPORT=AUTO or REPO is defined in the DYNAMIC_RESOURCES section of the DFSDFxxx member of the IMS PROCLIB data set, the stored resource definitions are read from the repository during IMS cold start if the repository contains the resource definitions for the IMS that is cold starting.

Related reference

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

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

FRP messages (Repository Server) (Messages and Codes)

Related information

FRP2002I (Messages and Codes)

FRP2001I (Messages and Codes)

FRP2006I (Messages and Codes)

FRP2025I (Messages and Codes)

Starting subordinate Repository Servers

To start a Repository Server (RS) as a subordinate, you only need to define it with the same z/OS cross-system coupling facility group as an RS that is already running.

However, all parameters in the FRPCFG member for both the master RS and its subordinates should be the same, except for the RSNAME= parameter. RSNAME= should be unique for each RS, including the master RS and its subordinates. RSNAME= is used to register to the Structured Call Interface (SCI) if IMSPLEX() is specified. SCI requires unique names for the clients.

Make sure that the subordinate RSs use the same RS catalog repository data sets as the master RS.

Related concepts

Overview of the IMSRSC repository (System Definition)

Recovery in an IMSplex (System Administration)

Stopping the Repository Server

To stop the Repository Server (RS), you can use an RS command issued from the z/OS MODIFY (F) interface.

About this task

To stop the RS, issue the following command from the z/OS MODIFY (F) interface:

```
F reposervername,SHUTDOWN
```

To stop all the RS address spaces, issue the following command from the z/OS MODIFY (F) interface. The command can be directed to a master RS or subordinate RS:

```
F reposervername,SHUTDOWN ALL
```

You can also stop the RS by issuing the z/OS **STOP (P)** command.

Related concepts

Overview of the IMSRSC repository (System Definition)

Related tasks

“Starting the Repository Server” on page 71

To start the Repository Server (RS), submit a customized RS startup procedure.

Related reference

F reposervername,SHUTDOWN (Commands)

How to start the IMS control region

You usually start the IMS control region with a z/OS **START** command from a z/OS system console.

If, depending on the size of your installation, you have an alternate system console for the IMS MTO, be sure the z/OS system operator relays all IMS messages (that is, those beginning with BPE, CQS, DFS, DSP, and DXR) to the IMS MTO.

If the z/OS system console is physically separate from the IMS MTO, you should have a local telephone or paging line to enable communication between the z/OS operator and the IMS MTO. The MTO should also have easy access to the operators responsible for DASD and tape units.

When the IMS control region is started, IMS issues two DFS1929I messages. Message DFS1929I lists the system parameters and their values defined for the IMS system. This information will help you understand the configuration of a particular system. Understanding the configuration will be helpful in diagnosing problems. The system parameters are also listed in the job log.

IMS issues message DFS1929I once during initialization and again when restart completes. The first DFS1929I message reflects the system parameter values that are defined in the DFSPBxxx PROCLIB

member. The second DFS1929I message reflects the system parameter values that are actually in effect for the current execution of IMS.

For online IMS systems, the control region automatically starts the DBRC address space. If you are running an IMS DB/DC or DBCTL environment, you can specify LS0=S as an EXEC parameter for the control region to start the DL/I address space. The following rules apply:

- IMS initialization waits until there is a separate address space available for DBRC.
- You can start DBRC manually (if, for example, a JCL error prevented DBRC from starting).
- When you restart IMS, you can change the specification of LS0=; IMS does not use any prior specifications.

After the control region has started, the MTO enters an IMS command to start the system. Select one of three ways of starting IMS:

1. Cold start
2. Warm start (also referred to as normal restart)
3. Emergency restart

Your operating procedures should include guidelines for the MTO in selecting the right type of startup.

You can alternatively use automatic restart instead of having the MTO enter an IMS command to restart the system. Specify AUT0=Y in the control region JCL to request automatic restart. IMS selects a suitable checkpoint for restart and notifies the MTO. You cannot use automatic restart if you:

- Want to perform a cold start
- Need to specify any options on the restart command (such as formatting system data sets, changing security options, or restarting only one portion of a DB/DC system)
- Need to override DBRC after a failure in which DBRC is unable to mark the subsystem record in the RECON data set as abnormally terminated, such as in the case of a power, CPC, z/OS, or DBRC failure

Setting the z/OS TOD clock

The z/OS time-of-day (TOD) clock setting is critical to the integrity of the IMS log. The TOD clock is also critical to the proper functioning of IMS restart, data sharing and XRF tracking and takeover.

Recommendations: Use one of the following procedures:

- Use an External Time Reference (ETR) device. Be sure to set it to use Universal Coordinated Time (also known as Greenwich Mean Time) with the appropriate offset for your local time zone.
- Set the z/OS TOD clock during IPL using Universal Coordinated Time, then set the local time.

Attention: During IPL, do not set the TOD clock to a time and date **earlier** than the immediate prior shutdown or failure. Setting the TOD clock back has severe IMS database integrity and recovery implications. If you do set the TOD clock back, you must reallocate the OLDSs using a different block size, reinitialize the DBRC RECON data sets, take image copies of all DBDSs, and cold start IMS.

Considerations for setting local time

Set your system's time zone offset (the difference between local time and Universal Coordinated Time (UTC)) to an integer multiple of 15 minutes. This is consistent with world time zone definitions and ensures that the local time reported by IMS matches the local time reported by the operating system.

IMS will function properly even if the time zone offset is not a multiple of 15 minutes. However, due to the way IMS stores time values internally, not following this recommendation might result in IMS reporting a local time that is inconsistent with the local time reported by the operating system.

More about IMS time stamps and setting the time zone offset

IMS stores all date/time stamps internally as 12 bytes in the following packed-decimal format:

```
YYYYDDDF HHMMSSth mijuAQQ$
```

The date and time fields (YYYYDDDF and HHMSSthmiju) are in Universal Coordinated Time (UTC). The A field consists of 4 bits of flags used internally by IMS. The QQ$ field contains the signed packed-decimal local time zone offset for when the time stamp was created. To get the local time, IMS adds the time zone offset—stored in quarter hours (fifteen-minute intervals)—to the UTC stamp. For example, the time stamp:

```
2000353F 06420588 4242032D
```

represents the UTC date and time of:

```
06:42:05.884242 on day 353 of 2000 (December 18th)
```

The QQ$ field of this time stamp is 32D, which represents a local time zone offset of minus 32 quarter hours (or minus 8 hours west of Greenwich). Thus, the local time represented by this time stamp is:

```
22:42:05.884242 on day 352 of 2000 (December 17th)
```

This format allows IMS internal time stamps to encode both the UTC time (to provide an always-incrementing time that is not subject to daylight savings time changes), and local time (to determine the “wall-clock” time for when the time stamp was generated). If your time zone offset is set to a multiple of fifteen minutes, IMS can accurately use the offset to convert from the UTC stored in the 12-byte time stamp into the local time.

In this example, the local time will be consistent with the time reported by the operating system. However, if your time zone offset is not set to a multiple of fifteen minutes, the local time derived from this conversion will not be consistent with the time reported by the operating system.

In the following example, the time zone offset is set to +01:55 (one hour and fifty-five minutes east of Greenwich). A time stamp is generated at 15:00:00.000000 UTC time (16:55:00.000000 local time, using the current time zone offset setting). IMS will construct an internal time stamp of:

```
2001240F 15000000 0000008C
```

The encoded time zone offset is 8C, or +8 quarter hours (2 hours). Because IMS cannot store a time zone offset with a precision greater than fifteen minutes, it rounds to the nearest fifteen minute boundary.

When this time stamp is converted back into local time, IMS reports a time of 17:00:00.000000, not 16:55:00.000000. This does not cause problems for IMS, which uses the more accurate UTC form of the time stamp internally. However, you will notice this rounding effect when you compare IMS-generated local times to the current operating system time. For example, if you issue an IMS checkpoint command on the system described above, the DFS058I message will indicate two different times, as shown below:

```
16:55:00 DFS058I 17:00:00 CHECKPOINT COMMAND IN PROGRESS
```

The first time value (16:55) is returned by the operating system; this value is generated by using the z/OS time zone offset of +01:55. The second time value (17:00) is returned by IMS; this value is generated from an internal IMS time stamp that uses a time zone offset of +08 quarter hours (2 hours).

This behavior may also be encountered by application programs that compare the local time stamp in the I/O PCB with the value returned from the z/OS TIME macro. The I/O PCB local time represents the time when the message was received by IMS. The data in this time field is derived from an internal IMS 12-byte time stamp stored with the message.

In the following example, a message is received by IMS at 12:42 UTC on the system described above. The local time is 14:37, based on the z/OS time zone offset of +01:55. However, the time presented to the application by IMS in the I/O PCB local time field is 14:42, due to the rounding of the time zone offset.

Now, suppose the message is scheduled at 12:43 UTC, one minute after it arrived. If the application program issues a z/OS TIME macro request, it will receive back a local time of 14:38. If the application were to compare this time to the I/O PCB time, it would appear as if the message was being processed before it was enqueued (enqueue time = 14:42; process time = 14:38). This is only a problem for an application program if it compares the I/O PCB time to a time generated from an operating system time service.

Related tasks

“Setting or changing local time” on page 76
You can reset local time whenever necessary, for example when changing from standard time to daylight savings (or summer) time.

Related reference

z/OS: External time reference (ETR)

Setting or changing local time

You can reset local time whenever necessary, for example when changing from standard time to daylight savings (or summer) time.

Procedure

1. To set or change the local time, edit the CLOCKxx member (the TIMEZONE keyword) of the SYS1.PARMLIB data set to set the current offset for local time.
2. If you have an ETR device, use it to set and change the time zone offset. If you do not have an ETR device, complete one of the following steps to set or change the local time:

Option	Description
Use the SET CLOCK command to set the time zone offset.	Using the **SET CLOCK** command does not allow you to set a precise value for your system's time zone offset. Consequently, using this method might cause IMS to report a local time that is inconsistent with the local time reported by the operating system.
Re-IPL your system with the updated CLOCKxx member.	Your time zone offset is set to the precise value specified in the member.

Related concepts

“Setting the z/OS TOD clock” on page 74
The z/OS time-of-day (TOD) clock setting is critical to the integrity of the IMS log. The TOD clock is also critical to the proper functioning of IMS restart, data sharing and XRF tracking and takeover.

Cold starting IMS in a non-shared queues environment

You perform a cold start when starting IMS for the first time, or after changing the nucleus in a system definition. IMS never uses information from a previous shutdown as input to a cold start.

About this task

In a nonshared-queues environment, a cold start assumes empty message queues, so IMS discards any messages that exist. You must close the last OLDS used by the online system before you attempt a cold start. You should also archive the last OLDS before a cold start, so the OLDS is available if a database recovery is required.

Cold starting IMS in a shared-queues environment

You perform a cold start when starting IMS for the first time, or after changing the nucleus in a system definition. IMS never uses information from a previous shutdown as input to a cold start.

About this task

In a shared-queues environment, IMS does not discard messages on shared queues. If you want to discard messages on shared queues, you must scratch the shared-queue structures on the coupling facility, and scratch and reallocate the CQS structure recovery data sets.

Cold starting IMS with dynamic resource definition

You perform a cold start when starting IMS for the first time, or after changing the nucleus in a system definition. IMS never uses information from a previous shutdown as input to a cold start.

During a cold start, resource and descriptor definitions are automatically imported from the IMSRSC repository, resource definition data set (RDDS), or the MODBLKS data set, or IMS comes up with no resources.

During a cold start, the partition status or access state will be copied from the HALDB master. If you export the HALDB master status to the RDDS or the repository, the status of the master and its partitions will be obtained from the RDDS or the repository.

Automatic import of resource and descriptor definitions is defined through the AUTOIMPORT parameter in the DYNAMIC_RESOURCES section of the DFSDFxxx member of the IMS PROCLIB data set

If AUTOIMPORT=AUTO is specified, these conditions determine the source of the imported resource and descriptor definitions:

If all of the following conditions are true, automatic import from the IMSRSC repository is enabled:

- For MODBLKS resources, MODBLKS=DYN is specified to enable MODBLKS DRD in IMS.
- For MSC resources, the MSC section of the DFSDFxxx member of the IMS PROCLIB data set is defined with MSCRSCS=DYN to enable MSC DRD in IMS.
- For MSC resources, the MSC section of the DFSDFxxx member is defined with MSCREPO=Y to enable MSC resources to be stored in the IMSRSC repository.
- The REPOSITORY section of the DFSDFxxx member is defined with TYPE=IMSRSC.
- IMS is enabled with RM services (RMENV=N is not specified in the DFSCGxxx member of the IMS PROCLIB data set or in the CSL section of the DFSDFxxx member of the IMS PROCLIB data set)
- The CSLRIxxx member of the IMS PROCLIB data set is defined for the repository with TYPE=IMSRSC.
- The repository contains stored resource definitions for the IMS system.
- RM is started with the repository enabled.
- The RS address space is started and available.

If all of the following conditions are true, automatic import from an RDDS is enabled:

- MODBLKS=DYN is specified to enable DRD in IMS.
- The REPOSITORY section of the DFSDFxxx member of the IMS PROCLIB data set is not present.
- Two or more system RDDSs are defined in the RDDSDSN() parameter of the DFSDFxxx member of the IMS PROCLIB data set.
- All of the defined RDDSs can be allocated and read.
- At least one of the RDDSs contains valid resource and descriptor definitions.

Related concepts

Overview of the IMSRSC repository (System Definition)

Related reference

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

Cold starting an IMS system that uses an RDDS

IMS systems that use dynamic resource definition to manage application programs, databases, Fast Path routing codes, and transactions can save resource definitions to a resource definition data set. During a cold start, IMS can use the definitions from the MODBLKS data set or the resource definition data set (RDDS) to create runtime resource definitions.

About this task

If your IMS system is not configured to import the resource definitions automatically at cold start, you can issue the **IMPORT** command after the restart process is complete. The **IMPORT** command, as well as all of the other type-2 dynamic resource definition commands, requires that the SCI and OM CSL components be started.

You can successfully start an IMS for the first time with DRD enabled and no resources defined yet. After the restart process completes, you can define resources dynamically by using the appropriate **CREATE** commands.

Before you shut down a running IMS system that uses dynamic resource definition for a cold start, you must export the resource and descriptor definitions that are currently defined in your online system to the RDDS. If you do not, any changes made to the resource definitions in the online system will be lost at the next cold start or emergency restart with the COLDSYS parameter.

During a cold start, the partition status or access state will be copied from the HALDB master. If you export the HALDB master status to the RDDS or the repository, the status of the master and its partitions will be obtained from the RDDS or the repository.

If an error occurs while trying to access an RDDS during the automatic import process, automatic import terminates abnormally. IMS either continues with the cold start process without any runtime resource definitions defined, or IMS stops the cold start process. If an error occurs during automatic import processing because of an invalid resource or descriptor definition, one of the following situations occurs:

- Automatic import processing terminates abnormally, and IMS cold start terminates abnormally with a U3397 abend.
- Automatic import processing continues, and the resource in error is marked and given a not-initiated status (NOTINIT).

The action IMS takes depends on the values specified for the RDDSERR and the IMPORTERR parameters in the DFSDFxxx member of the IMS PROCLIB data set.

Related concepts

Overview of the IMSRSC repository (System Definition)

Related tasks

Backing out DRD command-related changes made to your online system (System Definition)

Cold starting an IMS system that uses the IMSRSC repository

If IMS is using the IMSRSC repository, and AUTOIMPORT=AUTO or REPO is defined in the DYNAMIC_RESOURCES section of the DFSDFxxx member of the IMS PROCLIB data set, the stored resource definitions are read from the repository during IMS cold start if the repository contains the resource definitions for the IMS that is cold starting.

About this task

During a cold start, the partition status or access state will be copied from the HALDB master. If you export the HALDB master status to the RDDS or the repository, the status of the master and its partitions will be obtained from the RDDS or the repository.

The stored resource definitions from the repository are used to create the runtime resource definitions in the IMS system that is cold starting.

An IMS resource list in the repository is used during an IMS cold start to identify all the resource and descriptor definitions that are to be imported during the cold start.

The DFS3395I message is issued, indicating that the stored resource definitions are automatically imported from the repository. The DFS3396I message with a return code is issued when the automatic import from the repository is completed.

An RM error trying to read the stored resource definitions from repository results in the DFS4401E message with the RM return and reason code.

If REPOERR=ABORT is specified in the DFSDFxxx member, IMS terminates with a 3368 abend. The DFS3397E message with a return code of 16 is issued before the abend.

If REPOERR=NOIMPORT is specified in the DFSDFxxx member, IMS continues processing with no resources imported.

If there are no resource definitions in the repository for the IMS, the stored resource definitions from the resource definition data set (RDDS), MODBLKS data set, or the DFSCLL3x member of the IMS.SDFSRESL data set are read if AUTOIMPORT=AUTO is specified. If the RDDS, MODBLKS data set, or the IMS.SDFSRESL DFSCLL3x member is not defined or is empty, or AUTOIMPORT=REPO is specified, IMS will come up with no resources.

If there are no resources for the IMS in the repository, IMS issues the DFS4405W message, which indicates that the repository is empty.

If an RDDS was created by the Create RDDS from Log Records utility (DFSURCL0) to hold any new or modified resources that were not exported to the IMSRSC repository before IMS was shut down, you can either specify that RDDS on the **IMPORT** command after IMS cold starts from the IMSRSC repository, or use that RDDS as input to the CSLURP10 utility.

At the end of cold start processing, if IMS is not defined with AUTOEXPORT=AUTO or REPO, you must issue the **EXPORT DEFN TARGET(REPO)** command to export the runtime MODBLKS resource definitions imported from the RDDS or MODBLKS data set to the IMSRSC repository. If IMS has AUTOEXPORT=AUTO or REPO specified, it automatically exports any runtime resource definition imported from the RDDS, MODBLKS data set, or the IMS.SDFSRESL DFSCLL3x member to the repository at the end of the restart checkpoint.

If one exists, the change list in the repository for the IMS that is cold starting is deleted at the end of IMS cold start.

Related concepts

Overview of the IMSRSC repository (System Definition)

Resource lists for the IMSRSC repository (System Definition)

Related tasks

“Recovering IMSRSC repository data sets when no IMS systems or RM systems are active” on page 39
You can recover IMSRSC repository data sets when all IMS systems and RM systems are down and no IMS log is available but backup copies of the repository data sets are available.

Related reference

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

Create RDDS from Log Records utility (DFSURCL0) (System Utilities)

EXPORT command (Commands)

Related information

DFS4404I (Messages and Codes)

DFS3395I (Messages and Codes)

DFS3396I (Messages and Codes)

DFS4401E (Messages and Codes)

DFS3397E (Messages and Codes)

DFS4405W (Messages and Codes)

DFS3374W (Messages and Codes)

Warm starting IMS (or normal restart)

A warm start (or normal restart) is the most common way of initializing IMS, and is also the recommended way of restarting IMS after a controlled shutdown.

About this task

Use the **/NRESTART** command to perform a warm start or a normal restart. You can use the **FORMAT** parameter to format various system data sets (something you might want to do if you have reallocated any of the data sets). If you are running an IMS DB/DC or DCCTL environment, and you request reformatting of any of the message queue data sets, you must specify **BLDQ** on the **/NRESTART** command if IMS was previously shut down with either the DUMPQ or PURGE keyword specified.

If the MTO needs to dequeue transactions after a warm start, use the **/DEQUEUE SUSPEND** command.

If you want to reload MSDBs during restart, use the MSDBLOAD keyword on the **/NRESTART** command.

If IMS is enabled to use the IMSRSC repository, IMS calls Resource Manager (RM) during warm start 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.

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 system is deleted at the end of the warm start.

Related concepts

Resource lists for the IMSRSC repository (System Definition)

Emergency starting IMS

After IMS terminates without a controlled shutdown, you must perform an emergency restart. The one exception is if an emergency restart fails; in this case, you must perform a cold start with the **/ERESTART COLDSYS** command.

About this task

For an emergency restart, IMS restarts from the point of failure. Before restarting IMS, the MTO must know whether system data sets must be reallocated and reformatted during restart. The following table shows symptoms (a message or abend number) that indicate whether a system data set needs reallocation and reformatting.

Table 16. Symptoms indicating system data set reallocation is needed . **Key:** N/A=Not Applicable

System data set	Write error	Read error	Data set full	Open error
WADS	DFS0414I	DFS739I	N/A	DFS3256I
Message queue	U0759	U0759	U0758	DFS986A
Restart data set (RDS)	DFS3127I	U0970	N/A	U0970
MSDB checkpoint or MSDB dump	DFS2718I, DFS2722I	N/A	N/A	DFS2681I, DFS2713I, DFS2714I

The following situations cause the abends shown in the previous table.

U0758
An IMS internally generated **/CHECKPOINT DUMPQ** command failed after messages DFS206I, DFS207I, or DFS208I because one of the message queue data sets is full.

U0759
An error occurred while attempting to read or write the message queue data sets.

U0970
The restart data set (RDS) or checkpoint table was not initialized, or restart was unable to open the input log.

If the WADS has not previously been formatted, you should format it by specifying the FORMAT WADS keywords on the **/ERESTART** command. If you want to reformat any of the message queue data sets, specify the BLDQ keyword. If no checkpoint specifying either DUMPQ or SNAPQ has been taken since the last cold start, specify CHECKPOINT 0 on the **/ERESTART** command.

During Extended Specified Task Abnormal Exit (ESTAE) routine processing, IMS issues message DFS0617I if the RDS buffers have been purged and the RDS has been successfully closed. However, if you do not receive this message, you do not necessarily have to reformat or reallocate the RDS.

Requirement: You should reformat the RDS if you receive message DFS3127I or abend U0970, either of which indicate a read or write problem with the RDS. However, if the error has nothing to do with the RDS, you should not reformat the RDS. If errors occur on both MSDB checkpoint data sets:

Procedure

1. Issue the **/ERESTART** command.
2. Issue the **/DBDUMP DB** or **UPDATE DB STOP(UPDATES)** command for the MSDBs.
3. Shut down the system (in a controlled way).
4. Run the MSDB Dump Recovery utility.
5. Restart the system using an **/NRESTART** command, including the MSDBLOAD keyword so that IMS can reload the MSDBs.

Emergency restarts and dynamic resource definition

IMS systems that use dynamic resource definition automatically recover the resources and runtime descriptor definitions during emergency restart by processing the checkpoint records and X'22' log records during an emergency restart, unless the COLDSYS parameter is specified for the /ERESTART command.

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.

If the COLDSYS parameter is specified, the resource definitions are read from the resource definition data set (RDDS) or the IMSRSC repository.

If IMS is enabled to use the IMSRSC repository, the change list for the IMS, if one exists, is deleted at the end of COLDSYS processing.

Related concepts

"Emergency restart" on page 98
IMS requires an emergency restart whenever it terminates without a controlled shutdown. When restarted, IMS restarts from the point of failure. You can initiate an emergency restart using the **/ERESTART** command.

Overview of the IMSRSC repository (System Definition)

Resource lists for the IMSRSC repository (System Definition)

Related reference

/ERESTART command (Commands)

SLDS input to warm start and emergency restart

SLDS input might be required to perform a warm start or emergency restart. When the SLDS is required for restart, IMS dynamically allocates it. IMS also deallocates any data sets previously allocated with the dynamic allocation macro (DFSMDA) with IMSLOGR or IMLOGR2 as the DD name for the TYPE=SLDS statement.

DBRC provides the data set name and volume information for the dynamically allocated SLDSs. However, DBRC does not provide the unit type information, so a DFSMDA member with the name of IMSLOGR must provide the unit type.

In some cases, you may want to force IMS to read from the SLDS instead of the OLDS during restart—for example, to prevent IMS from reading an OLDS that you know is invalid or has been corrupted. To force IMS to read from the SLDS, you need to delete the OLDS entry from the PRIOLDS and SECOLDS records in the RECON data set. To delete an OLDS entry from the RECON data set, use the **DELETE.LOG OLDS(***ddname***)** command, where *ddname* is the name of the primary OLDS.

Attention: When image copy, change accumulation, and log data sets are listed in the RECON status record as being under the control of a catalog management system (CA | IC | LOG DATA SETS CATALOGED=YES), the restart of IMS will fail if the SLDS data sets are not actually cataloged. The volume serial number (VOLSER) is not passed back to IMS for data set allocation when the data sets are not cataloged. Likewise, if the data sets are SMS-managed and not cataloged (CA | IC | LOG DATA SETS CATALOGED=NO), the restart of IMS might fail if SMS migrated the SLDS data sets to a different volume serial number. In this case, the volume serial number will be passed back to IMS but it does not match the volume serial number where the data set resides. Therefore, it is important to set the CATDS keyword on the **CHANGE.RECON** or **INIT.RECON** command correctly.

Specifying security options at startup

Depending on your system definition, the MTO might be able to specify whether IMS activates various security options during startup. Each restart command options affects security differently, and can be used with the **/NRESTART** command or the **/ERESTART COLDSYS** command.

The following table shows the restart command options that affect security.

Table 17. Security options on the /NRESTART or /ERE COLDSYS command

JCL EXEC parameters	/NRE or /ERE command input	System security level after restart
SGN=Y	USER	Signon active. Single signon per user for static users.[1]
	TRANAUTH	Signon and transaction authorization active.
	CMDAUTH	Signon, static, and ETO command authorization active.
	CMDAUTHE	Signon and ETO command authorization active.
	MULTSIGN	Signon active. Multiple signon per user (static users).[1]
SGN=N	NOUSER	Signon not active.

Table 17. Security options on the /NRESTART or /ERE COLDSYS command (continued)

JCL EXEC parameters	/NRE or /ERE command input	System security level after restart
TRN=N	NOUSER NOTRANAUTH	Signon and transaction authorization not active. Transaction authorization not active. Signon active.
TRN=Y	TRANAUTH	Signon and transaction authorization active.
TRN=F	FORCTRAN	Signon and transaction authorization enforced.
SGN=F SGN=Z SGN=G	FORCSIGN SNGLSIGN	Signon enforced. Signon active. Single signon per user (static users).[1] Signon enforced. Multiple signons per user (static users).[1]
RCF=C	NOUSER NOCMDAUTHE NOCMDAUTH	Signon and command authorization for ETO not active. Command authorization for ETO not active. Command authorization for ETO/static users not active.
RCF=S	NOUSER NOCMDAUTHE NOCMDAUTH CMDAUTHE	Signon for ETO and static users not active. Command authorization for ETO/static users not active. Command authorization for ETO/static users not active. Command authorization for ETO active. Command authorization for static users not active.
RCF=N	NORACTRM+NORACFCM CMDAUTHE CMDAUTH	Signon and command authorization for ETO active. Signon and command authorization for ETO and static users active.

Note:

1. This also applies to dynamic users signing on to ETO terminals where the user structure name is different from the user ID.

Starting the IRLM

You use the internal resource lock manager (IRLM) to perform block-level or sysplex data sharing. If you are running an IMS DB/DC or DBCTL environment, start the IRLM from the z/OS system console using the z/OS **START** command.

About this task

You must start IRLM subsystems before starting the IMS batch or online subsystem. The IMS online subsystem identifies itself to the IRLM subsystem during IMS initialization or restart.

Related concepts

Defining an IRLM for sysplex data sharing (System Administration)

Starting the CQS

In an IMS shared-queues environment, IMS starts the CQS subsystem during IMS startup or restart if the CQS subsystem is not already running. You can use the z/OS **START** command to start the CQS subsystem if you want to ensure CQS is running before you start IMS.

About this task

After CQS is running, the IMS online subsystem identifies itself to the CQS subsystem during IMS initialization or restart.

If you register CQS with the z/OS Automatic Restart Manager, you do not have to restart CQS after a CQS failure. Because CQS and IMS are separate subsystems, you do not have to restart CQS after an IMS failure.

You can start CQS in one of three ways:

- As a z/OS task, using the z/OS **START** command
- As a z/OS batch job
- As a client task — Some clients (such as IMS) automatically start CQS, when appropriate

  Example: If IMS is the client, define the CQS name in the DFSSQxxx or the DFSDFxxx member in the IMS PROCLIB data set and specify the CQS name on the SHAREDQ parameter of the IMS procedure. IMS does not start CQS if the CQS is needed only to manage a resource structure.

A CQS that supports only a resource structure must be started manually because IMS does not start this CQS.

Related tasks

Preparing to start the CQS address space (System Definition)

Related reference

DCC procedure (System Definition)

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

z/OS: Defining Sysplex policies

Starting dependent regions

IMS dependent regions are z/OS regions that are separate from the IMS control region, and that are logically dependent upon the IMS control region.

About this task

The following types of programs execute in dependent regions:

- IMS batch message programs (BMPs) and message processing programs (MPPs)
- IMS Fast Path (IFP) message-driven application programs
- Fast Path DEDB online utility programs
- Java™ batch processing (JBP) and Java message processing (JMP) applications
- Coordinator controller (CCTL) application programs (in the IMS DBCTL environment)

You can start IMS dependent regions from the z/OS system console or the IMS master terminal. From the z/OS system console, use the z/OS **START** command to start a reader procedure (usually IMSRDR) to read the JCL for the region. From the IMS master terminal, use the IMS **/START REGION** command to start a region. If you do not specify a procedure name on the **START** or **/START REGION** command, IMS uses the IMSMSG procedure.

You should not start IMS message regions and Fast Path regions with automatic priority group (APG) initiators because IMS cannot control the dispatching priority of an APG-initiated job or uniquely identify such jobs. You might need to start and stop z/OS initiators for IMS message regions and Fast Path regions that have APG initiators.

IMS can dynamically allocate up to 255 regions. The total number of regions includes the number of regions allocated during IMS system definition plus the number of dynamically allocated regions or CCTL threads.

Message processing regions

One of the jobs in the IMS.JOBS data set contains the message region JCL. Parameters on the JCL EXEC statement define the transaction classes that are eligible for processing in the region. You can define more than one set of JCL if necessary.

The default job name is IMSMSG. After you start a message region, transactions associated with the transaction classes supported by the region can be scheduled. You can change these classes using the **/**

ASSIGN TRAN or **UPDATE TRAN SET(CLASS(***new_class_number***))** command after you have started the region.

Batch message processing regions

You can start BMP regions from the z/OS console or the IMS master terminal. Generally, you use the z/OS console and a z/OS reader procedure to start a BMP, unless the JCL for the job (using procedure IMSBATCH) is in the IMS.JOBS data set.

A JCL EXEC parameter indicates that the job is a BMP.

Fast Path message-driven regions

You can start IMS Fast Path message-driven regions from the z/OS console or the IMS master terminal. Generally, you use the z/OS console and a z/OS reader procedure to start a Fast Path message-driven region, unless the JCL for the job (using procedure IMSFP) is in the IMS.JOBS data set.

A JCL EXEC parameter indicates that the job is a Fast Path message-driven application program.

Java dependent regions

You can start two types of Java message-driven dependent regions, either a JMP or a JBP region.

The JMP region is similar to an MPP region and is started in the same manner as an MPP region. The default job name is IMSJMP.

The JBP region is similar to a BMP region and is started in the same manner as a BMP region. The default job name is IMSJBP.

DEDB online utility regions

You can start all DEDB online utilities using procedure FPUTIL. Specify the utility in the TYPE operand in the control data set.

CCTL regions

The CCTL region is not part of an IMS subsystem. See your CCTL documentation for information on how to start the CCTL.

Restarting CQS

After CQS completes the structure initialization, it continues with the restart. CQS can do either a cold restart or a warm restart. Restarting CQS affects only the units of work that this CQS manages. Restarting does not back out or restore any units of work owned by another CQS.

When you have completed restarting all structure pairs for a particular CQS, the CQS ready message is issued (CQS0020I).

Since changes to resource structures are not check pointed or logged, restarting CQS does not affect units of work for resource structures.

The frequency of system checkpoints affects restart. CQS must read more log records when checkpoints are infrequent than when checkpoints occur more often. Also, the amount of logging that one CQS performs can affect another CQS during restart. All CQSs write to the same log, so a CQS restarting must read all log records written by all CQSs.

CQS takes an initial system checkpoint at the end of a restart.

CQS warm start

During a warm start, CQS reads the log records from the last system checkpoint, restores the environment for committed data objects, and backs out uncommitted data objects to prepare CQS to regain

synchronization with the client and resume processing. Normally, a CQS warm start is automatic and you do not need to take any action.

About this task

When CQS warm starts, it reads the checkpoint data set to find the log token representing the last system checkpoint. When CQS finds this log token, it initiates a warm start. If CQS fails to find this log token in the checkpoint data set, it reads the log token from the structure. If CQS finds the log token, CQS issues WTOR CQS0031A to allow you to confirm the use of this token. At this point, you can do one of the following:

- Confirm the log token
- Cold start CQS
- Cancel CQS
- Specify a new log token

If you specify a new log token and CQS fails to find this log token, CQS issues WTOR CQS0032A. At this point, you can take one of the following actions:

- Cold start CQS
- Cancel CQS
- Specify a new log token

Sometimes, CQS purges log records that are required for restart. CQS purges log records in the following situations:

- During a structure checkpoint
- When the log becomes full and no more data sets are available for logging

Important: CQS might not have any log records if it is only managing resource structures.

Recommendation: If a CQS does not accept a log token during CQS restart, cold start the CQS. In cases where multiple CQSs are running, it is possible that log records for a CQS that previously failed and was not restarted are purged while another CQS performs a structure checkpoint.

Related concepts

Using CQS structure checkpoint (System Administration)

CQS cold start

When CQS restarts after a structure cold start, CQS cold start processing is automatic. You do not need to take any action. CQS takes a system checkpoint and then CQS restart is complete.

About this task

When CQS cold starts after a structure warm start or a structure recovery, CQS reads the structure to find unresolved work. CQS backs out requests to move data but completes requests to delete data. CQS performs a system checkpoint and restart is complete.

No log records are read or processed when CQS is cold started.

To cold start CQS, you must:

Procedure

1. Scratch the CQS system checkpoint data set for the structure.
2. Reply COLD to the CQS0031A WTOR.

CQS registration with the z/OS Automatic Restart Manager

CQS, if requested, can register with the z/OS Automatic Restart Manager (ARM). The ARM is a z/OS recovery function that can improve the availability of started tasks. When a task fails or the system on which it is running fails, the ARM can restart the task without operator intervention.

Recommendation: Register with the z/OS ARM regardless of the types of structures that CQS is using.

To enable the ARM, you can specify ARMRST=Y in one of two ways:

- In the CQSIPxxx member of the IMS PROCLIB data set
- As an execution parameter

An abend table exists in module CQSARM10. The table lists the CQS ends abnormally for which the ARM does not restart CQS after the abend occurs. You can modify this table.

IBM provides policy defaults for automatic restart management. You can use these defaults or define an ARM policy to specify how CQS should be restarted. The policy can specify different actions to be taken when the system fails and when CQS fails. When ARM is enabled, CQS registers to ARM with an ARM element name of CQS + *cqsssn* + CQS. Use this ARM element name in the ARM policy to define the ARM policy for CQS.

cqsssn is the CQS name that can be defined either as a CQS execute parameter, or with the **SSN**= parameter in the CQSIPxxx member of the IMS PROCLIB data set. For example, if **SSN**=CQSA, then the ARM element name is CQSCQSACQS.

Restarting CQS after CQS resource cleanup failures

If CQS abends and you receive message CQS0102E with module CQSRSM00, log records might be missing from the CQS log. Message CQS0102E indicates that a failure occurred during CQS resource cleanup. This failure might prevent CQS log records in internal buffers from being externalized to the CQS log.

About this task

If this situation occurs, perform one of the following actions:

- If the terminating CQS is the only CQS running for its set of structures, restart the CQS immediately.
- If other CQSs are running, either immediately restart the terminated CQS or initiate a structure checkpoint on one of the surviving CQSs.

Successfully restarting the failed CQS or taking a structure checkpoint is necessary to preserve the state of the data on the shared queues in the event that a structure rebuild is needed.

Restarting CQS queue structures

Restarting queue structures involves restarting a structure pair. The structure restart function ensures that the data in the structure is correct. Structure restart deals only with the status of data in a specific coupling facility structure, not units of work specific to a given CQS.

About this task

Before CQS can restart, CQS must recover each queue structure defined to CQS, if needed. CQS connects to both the primary and the overflow queue structures for each structure pair defined. CQS determines whether the structures need to be warm or cold started and performs the necessary recovery functions. If more than one structure pair is defined to CQS, one structure pair can be warm started and another can be cold started.

CQS structure allocation

A structure is allocated the first time a CQS connects to it and remains allocated until you explicitly delete it using the z/OS **SETXCF** command.

When a CQS connects to a structure, the structure might be empty; that is, it might contain no data, might contain only CQS control information, or it might contain client data. The structure can be empty if:

- This is the first time a CQS is accessing the structure.
- You scratched the structure to perform a structure cold start.
- A structure failure occurred and the structure must be recovered.

CQS structure warm start

CQS warm starts both a primary structure and its overflow structure if either the primary structure contains data, or one SRDS contains valid structure checkpoint data and the CQS log contains valid data.

About this task

During a structure warm start, CQS determines the status of the structure and initiates a structure recovery if necessary. If a structure recovery is needed, CQS allocates the structure and repopulates it from either the CQS log and the SRDS, which contains valid client data from a previous checkpoint, or from the CQS log by itself.

After a structure warm start has completed, CQS determines whether a future recovery is possible based on the status of the structure recovery data sets and the log stream. If the primary structure contains client data, but neither the SRDS nor the log can be used for future recovery, CQS issues a CQS0009W message.

Recommendation: If CQS issues a CQS0009W message, initiate a structure checkpoint as soon as possible. If a structure checkpoint does not complete successfully and the structure fails, CQS cannot recover the structure.

After a structure warm start, CQS can be cold started or warm started. If the log records needed for the CQS restart have been deleted, you might have to cold start the CQS.

CQS structure cold start

Common Queue Server (CQS) cold starts both a primary structure and its overflow structure if the primary structure is empty and both of the structure recovery data sets are empty.

About this task

During a structure cold start, CQS deletes:

- The overflow structure
- All the log records in the log stream for the structure

To cold start a structure, you must:

Procedure

1. Shut down any CQS systems that are connected to the structure.
   For example, issue the following command, where *cqs_name* is the CQS subsystem name.

   ```
   P cqs_name
   ```

   The response will be shown as follows:

   ```
   CQS0021I CQS SHUTDOWN COMPLETE
   CQS0101I CQS CLEANUP SUCCESSFUL
   ```

2. Delete the primary and overflow structures on the coupling facility.
   For example, issue the following command, where *structure_name* is the structure name.

   ```
   SETXCF FORCE,STRUCTURE,STRNAME=structure_name
   ```

   The response will be shown as follows:

   ```
   IXC353I THE SETXCF FORCE REQUEST FOR STRUCTURE
   structure_name WAS COMPLETED:
   STRUCTURE WAS DELETED
   ```

3. Scratch both structure recovery data sets (SRDS 1 and 2).
   For example, use the Access Method Services program IDCAMS to scratch the data sets.
4. Reallocate both structure recovery data sets (SRDS 1 and 2).
   For example, use the Access Method Services program IDCAMS to reallocate the data sets.
5. Restart a CQS. CQS initialization will reallocate the structure.
   For example, use the CQS startup procedure.
6. Verify that the structure is empty.
   For example,
   - Issue the following z/OS command, where *structure_name* is the structure name.

     ```
     D XCF,STR,STRNAME=structure_name
     ```

   The response for the z/OS command will be shown as follows:

   ```
   ...
   STORAGE INCREMENT SIZE: 256 K
   ENTRIES:  IN-USE:          3 TOTAL:       4238,   0% FULL
   ELEMENTS: IN-USE:          3 TOTAL:       4238,   0% FULL
   EMCS:     IN-USE:          0 TOTAL:       6634,   0% FULL
   ...
   ```

Results

When structure cold start completes, CQS automatically performs cold start restart processing.

Restarting resource structures

You can cold start a resource structure by shutting down any CQS systems that connect to the resource structure, deleting the resource structure on the coupling facility, and restarting a CQS to reallocate the resource structure.

About this task

To cold start a resource structure, you must:

Procedure

1. Shut down any CQS systems that connect to the resource structure.
   For example, issue the following command, where *cqs_name* is the CQS subsystem name.

   ```
   P cqs_name
   ```

   The response will be shown as follows:

   ```
   CQS0021I CQS SHUTDOWN COMPLETE
   CQS0101I CQS CLEANUP SUCCESSFUL
   ```

2. Delete the resource structure on the coupling facility.

For example, issue the following command, where *structure_name* is the resource structure name.

```
SETXCF FORCE,STRUCTURE,STRNAME=structure_name
```

The response will be shown as follows:

```
IXC353I THE SETXCF FORCE REQUEST FOR STRUCTURE
structure_name WAS COMPLETED:
STRUCTURE WAS DELETED
```

3. Restart a CQS. CQS initialization will reallocate the resource structure.
   For example, use the CQS startup procedure.
4. Verify that the resource structure is empty.
   For example, issue the following command, where *structure_name* is the resource structure name.

```
D XCF,STR,STRNAME=structure_name
```

The response will be shown as follows:

```
...
STORAGE INCREMENT SIZE: 256 K
ENTRIES:  IN-USE:          3 TOTAL:       4238,   0% FULL
ELEMENTS: IN-USE:          3 TOTAL:       4238,   0% FULL
EMCS:     IN-USE:          0 TOTAL:       6634,   0% FULL
...
```

Results

When structure cold start completes, CQS automatically performs cold start restart processing.

CQS structure recovery for restarting

A structure might need to be recovered if the structure is empty or if it contains only Common Queue Server (CQS) control information. Data from the last structure checkpoint and the z/OS log stream are used to recover a structure.

The structure is first repopulated with data objects from the structure recovery data sets. CQS then reads the log, starting at the time of the structure checkpoint, and updates the structure with changes that occurred after the structure checkpoint.

If the primary structure is empty and neither SRDS contains valid structure checkpoint data, CQS determines whether it can use just the CQS log for recovery. If the first log record in the log stream is the "Beginning of Log" record, the log stream contains all the log records required for recovery, and CQS can use the log records to complete the structure recovery.

If CQS finds that a previous structure rebuild did not complete successfully, it initiates another rebuild.

If the primary structure contains only CQS control information and the first CQS that connected to the structure (the one that allocated the structure) cannot determine whether a rebuild is needed, CQS initiates a rebuild if either SRDS is valid or if all the log records are available.

If neither SRDS is valid and the log records have been deleted by a previous structure checkpoint, CQS cannot rebuild the structure. When this happens and a rebuild is necessary, CQS issues write-to-operator-with-reply (WTOR) CQS0034A message asking what you want to do. You can cold start the structure or cancel this CQS.

Related concepts

CQS structure recovery (System Administration)

Starting Transaction Manager

For remote terminal operators to enter transactions, you must initialize the IMS Transaction Manager (for the IMS DB/DC and DCCTL environments).

The procedures are different for the different access methods and protocols:

- Virtual Telecommunications Access Method (VTAM)
- Communications with devices
- Advanced Program-to-Program Communications (APPC/MVS™)
- Intersystem Communication (ISC)

Connecting to the VTAM network

You use the **/START DC** command to establish communications between IMS and VTAM. The **/START DC** command tells VTAM to pass queued logon requests to IMS for all terminals defined to VTAM as belonging to IMS.

About this task

Before establishing a session with IMS for any terminal, verify that the following conditions exist:

- VTAM and the Network Control Program (NCP) are active.
- Controllers and logical units are active. If they are not automatically activated by VTAM, the VTAM network operator can use the VTAM VARY command to activate them.
- Controllers are powered on, appropriately configured, initialized, and activated for VTAM and NCP.

Recommendation: Start VTAM before starting the IMS control region.

You also need to use the **/START DC** command in either of the following situations:

- If VTAM is stopped and restarted while the IMS control region is running
- If you terminate communications between IMS and VTAM using the **/STOP DC** command

The /START DC command activates the following processes:

Initiating IMS Transaction Manager processing
Opening the VTAM access method control block (ACB)
Enabling the IMS VTAM logon exit routine
Opening the primary and secondary VTAM master terminals if these terminals were not opened during IMS initialization (due to the specification of VACBOPN=DELAY in the DFSDCxxx IMS.PROCLIB member)

If VTAM is active when IMS starts, the IMS VTAM ACB is opened; otherwise, the ACB is opened by the **/START DC** command. Any logon requests received by VTAM before the IMS **/START DC** command, but after the IMS VTAM ACB has been opened, are enqueued by VTAM until the **/START DC** command completes. If the queuing of logon requests by VTAM causes operational problems, the specification of VACOPN=DELAY in the DFSDCxxx PROCLIB member can be used to delay the opening of the VTAM ACB. When the master terminals are generated to use VTAM, the master terminal operator will be unable to enter either the **/NRE** or **/ERE** commands (including **/ERE BACKUP**) or the **/START DC** command. These commands can be issued using automatic restart (AUTO=Y) or automated operations.

Before you can establish a session between IMS and a logical unit or node, the logical unit or node must be active, connected, enabled, and available to VTAM and started in IMS. You can activate logical units or nodes when VTAM starts using VTAM start options, or a VTAM operator can activate them using the VTAM **VARY** command.

Normally after you start IMS, all VTAM terminals have a started status. If you stop a VTAM terminal with an IMS **/STOP** command, the terminal remains stopped after an IMS warm start or emergency restart. You must use the **/START NODE** command before issuing the **/OPNDST** (open destination) command to reactivate the terminal.

You must establish sessions between terminals and IMS before anyone can use them to transmit data. You can initiate sessions in several ways:

- VTAM can automatically log them on (based on the VTAM terminal definition).
- The end user can log on.
- You can issue an IMS **/OPNDST** command (if they are not automatically logged on).
- You can issue a VTAM **VARY** command.

Connecting to devices

When you start IMS, IMS automatically starts communications between IMS and devices such as spools, readers, printers, punches, tapes, and disks. After you start IMS, the telecommunication lines have a stopped status.

About this task

You can start the telecommunication lines, and devices if necessary, using the **/START** or **/RSTART** commands. Normally, you use the **/START** command after an IMS cold start and the **/RSTART** command after a warm start. The **/RSTART** command starts the devices so that they have the same mode they had when they were stopped at the previous system shutdown.

If you stop a device using the **/STOP** command, the device remains stopped after an IMS warm start or emergency restart.

Connecting to APPC/MVS

Before an LU 6.2 conversation can be established with IMS, the following must be active: APPC/MVS, VTAM, and NCP.

About this task

Recommendation: Start APPC/MVS and VTAM before starting IMS.

When you specify APPC=Y in the IMS procedure, IMS automatically starts communication between IMS and APPC/MVS. If you specify APPC=N in the IMS procedure, you can use the **/START APPC** command to establish communications.

You also need to use the **/START APPC** command in either of the following situations:

- If APPC/MVS is stopped and restarted while IMS is running
- If you terminate communication between IMS and APPC/MVS by using a **/STOP APPC** command

Connecting ISC sessions from CICS to IMS

You can connect CICS® ISC sessions to IMS in two distinct ways. You can initiate the session explicitly or the CICS operator can initiate the session through a command.

About this task

You can connect CICS ISC sessions to IMS in one of the following ways:

- You can initiate the session explicitly by specifying AUTOCONNECT YES on the **DEFINE CONNECTION** command in the CICS system definition (CSD) utility, or by specifying **CONNECT**=AUTO in the **DFHTCT TYPE**=TERMINAL program. When CICS is started, it attempts to establish all sessions for which AUTOCONNECT is specified.

  In order for session to be initiated, IMS must be active when CICS is started.
- The CICS operator can initiate a session by entering the following command:

  CEMT SET TERMINAL*(termid)* **[ACQUIRED | COLDACQ | RELEASED]**

where *termid* is the four-character session name (defined on the **DEFINE SESSIONS** command in the CSD utility), or the TRMIDNT in the DFHTCT TYPE=TERMINAL program. The following describes the effects of the keywords for the **CEMT SET TERMINAL** command:

ACQUIRED
Specifies normal resynchronization with the IMS half-session. CICS is in session with the logical unit represented by the terminal.

COLDACQ
Specifies that no resynchronization is done.

COLDACQ is a special form of ACQUIRED, in which no resynchronization is required. If the previous session ended abnormally, using COLDACQ overrides CICS integrity control, which can lead to integrity problems. Also, check the CSMT log for an activity keypoint (similar to an IMS checkpoint) after the restart of a session following a CICS failure. If there is no activity keypoint, reissue the **CEMT SET TERMINAL***(termid)* COLDACQ command after the next emergency restart.

RELEASED
Specifies that CICS is not in session with the logical unit represented by the terminal.

Specifying RELEASED for a terminal that has an active session causes the session to be terminated. Running transactions are allowed to finish unless you also specify PURGE or FORCEPURGE.

The display area of the **CEMT [INQUIRE | SET] TERMINAL** command shows the status of the CICS half-session. If the session initiates successfully, the status changes from REL (released) to ACQ (acquired). If the session does not initiate successfully, CICS writes an error message to the transient data destination CSMT.

- An application program can initiate a session implicitly by using the **ALLOCATE** command. A program can use the **ALLOCATE** command only for the SEND/RECEIVE interface.

  Recommendation: Use the **ALLOCATE SYSID** form of the command because it allows CICS to select an available session.

  If a session is not immediately available, control is returned to the application program if either of the following occur:

  – The program issues a **HANDLE CONDITION** command.
  – The program specifies NOQUEUE on the **ALLOCATE** command.

  Otherwise, the command is queued until a session becomes available.

  Each of the following conditions causes a session to be "not immediately available".

  – All sessions to the specified system (or the specified session) are in use.
  – The only available sessions are not bound.
  – The only available sessions are contention losers.
- CICS automatic task initiation (ATI) can initiate a session.

To determine the primary and secondary RU sizes, CICS uses the values for the RECEIVESIZE and SENDSIZE parameters on the **DEFINE CONNECTION | SESSION** command in the CSD utility or the values for the RUSIZE and BUFFER parameters on the DFHTCT TYPE=TERMINAL | SYSTEM program.

Related concepts

“Terminating an ISC session from CICS” on page 124
You can only use CICS control operator commands to terminate ISC sessions.

Related reference

CICS Transaction Server for z/OS

Starting IMS systems and global command status

Global command status is the status of a shared IMS resource as set by online commands and as seen by all IMSplex members that have the resource defined.

If you are operating in an IMSplex environment that specifies PLEXPARM(GSTSTRAN(Y), GSTSDB(Y) or GSTSAREA(Y)), you must understand how the status of IMS resources as set by online commands is maintained at both the global and the local level, as well as how resource status is recognized by IMSplex members as individual IMS systems are started and stopped across the IMSplex.

For example, if the Resource Manager (RM) maintains global command status information for a database and the database is stopped by a command, all members of the IMSplex, whether they were running at the time the command was issued or not, see that the database is stopped. That same database can have a local status in an IMS system that is different than the global command status maintained in RM. Stopping a database globally while allowing access locally can be useful for preventing updates to the database from other IMSplex members while a local batch program processes the database.

RM maintains global command status for IMS resources by resource type. The maintenance of global command status is enabled for a resource type by specifying the PLEXPARM= parameter in the DFSDFxxx or DFSCGxxx PROCLIB member during IMS initialization.

You can assign global status to a resource dynamically by using one of the following commands:

- **UPDATE IMS SET(PLEXPARM(GSTSDB(Y))** identifies that global database status is to be maintained on database commands.
- **UPDATE IMS SET(PLEXPARM(GSTSAREA(Y))** identifies that global area status is to be maintained on area commands.
- **UPDATE IMS SET(PLEXPARM(GSTSTRAN(Y))** identifies that global transaction status is to be maintained on transaction commands.

If global status is not explicitly specified for a resource type, RM does not maintain global status.

When RM maintains global status for a resource type, any commands issued against specific instances of the resource change the global status of the resource instance in RM. You can limit the affect of commands on the global status of a resources by using the SCOPE() parameter with the appropriate type-2 command:

- The SCOPE(ACTIVE) parameter limits the scope of commands to the active IMS systems that the command is routed to, and no global information is changed.
- The SCOPE(ALL) parameter specifies the scope of commands to the active IMS systems that the command is routed to and updates the RM status if applicable.

If a command is issued against a resource with global status before an IMS system is started, the starting IMS system can read the global status of resources in RM and apply them locally to the IMS system if the resources are valid.

The first IMS system to start in an IMSplex sets the PLEXPARM values that it knows about. IMS systems that start in the IMSplex subsequent to the first, use the PLEXPARM status definitions already in RM.

When an IMS system restarts and global status exists for the resource type in RM, the type of start determines which status takes precedence:

- For cold starts, the global status of databases, partitions, areas, and transactions takes precedence.
- For warm starts, the restarting IMS system first applies the local status saved in the log records and then applies the global status stored in RM only if it changed while the IMS system was down.

Before attempting to restart IMS again, complete one of the following actions:

1. Cancel IMS or shutdown IMS after restart completes. After the error condition reported in the DFS3308 message is fixed, restart IMS. IMS will be synchronized with RM status, and no additional action is required.
2. Allow the IMS restart to complete, then take the following actions:

- Issue the appropriate **UPDATE IMS SET(PLEXPARM)** command to enable global status at this IMS.
- Issue the appropriate **QRY TRAN|DB|AREA SHOW(GLOBAL)** commands to see the resources with global status.
- Issue appropriate command with the SCOPE(ACTIVE) parameter to ensure that the IMS is synchronized with RM status.

RM stores global status information in its resource structure. If the resource structure should fail, RM repopulates the resource structure by having each active IMS system in the IMSplex update the resource structure with its local resources that have global status. Any global status for resources owned by an IMS system that is not active is not restored to the resource structure.

Related concepts

"IMS restart and global resource status in an IMSplex that uses RM" on page 100
If you are operating in an IMSplex environment that uses the Resource Manager (RM) component of the Common Service Layer, you need to understand how restarting an IMS system affects both the global and local status information of the IMS resources that the restarting IMS system shares with the rest of the IMSplex.

Restarting a component of IMS

After performing the necessary recovery, you can reactivate the particular component that you shut down or detached. For example, if you take a database offline because of an I/O error and then recover it, you can again make it available to applications.

Use the appropriate IMS command to restart the component. Most often, this command is some form of the **/START** or **UPDATE** command. For example, use **/START DATABASE** or **UPDATE DB START(ACCESS)** to restart a database or **/START PROGRAM** to restart an application program.

A list of restart commands and their corresponding shutdown commands is shown in the following table:

Table 18. Commands for restarting components of IMS

Restart command	Corresponding shutdown command	Affected components
/OPNDST	**/CLSDST**	VTAM session
/START	**/STOP /IDLE /DBDUMP / DBRECOVERY**	Areas, databases, VTAM logons, lines, logical terminals, VTAM nodes, OLDS, programs, regions, external subsystem connection, transactions, users
/RSTART	**/IDLE**	Lines, physical terminals, logical links, VTAM nodes
/RELEASE	**/HOLD**	Conversations
/UNLOCK	**/LOCK**	Databases, logical terminals, nodes, programs, physical terminals, transactions
UPDATE	**UPDATE**	Areas, data groups, databases, MSC resources, and transactions.

Restarting IMS

You can restart IMS after you have completed any necessary recovery. You can perform a normal, emergency, or emergency with OVERRIDE restart.

- *Normal restart* initializes the IMS subsystem without reference to any previous execution of IMS (a cold start), or restarts a system that was terminated using a checkpoint command (a warm start).

- *Emergency restart* initializes an IMS subsystem after a system failure. During an emergency restart, IMS does the following:
  - Closes the OLDS from the WADS
  - Resets each active transaction to its last sync point
  - Resets each active region (BMPs, and CCTL or ODBA threads) to its last sync point

    You do not need to restart connected ODBA application programs or CCTLs. An IMS shutdown or failure simply disconnects from the ODBA application or CCTL and generally does not affect it.
  - Restores the databases to the last sync point
  - Restores local message queues to their condition at the time of failure

    An IMS failure does not generally affect shared message queues.
  - If dynamic resource definition is enabled in the IMS system, the runtime resource definitions are restored by processing the checkpoint and X'22' logs

  You must manually restart regions and BMPs. However, when restarting a batch message processing region, IMS automatically determines the last BMP sync point if you specified LAST as an EXEC parameter. Otherwise, IMS indicates what the last BMP sync point was so that you can specify it during restart.

  If you have an IMS DBCTL standby environment, you can send the **/ERESTART** command to the pre-initialized DBCTL subsystem. Doing this is faster than starting the IMS job and waiting for the DBCTL-ready message before sending the **/ERESTART** command.
- *Emergency restart with OVERRIDE* is necessary after a failure in which DBRC is unable to mark the SSYS record in the RECON data set as abnormally terminated, such as a power, CPC, z/OS, or DBRC failure.

Automatic restart reduces MTO intervention and makes restart faster. With automatic restart, IMS automatically chooses the appropriate restart command (either **/NRESTART** or **/ERESTART**). If restart processing abnormally terminates before initial checkpoint, the appropriate automatic restart command is the same type (either **/NRE** or **/ERE**) as the aborted restart. The operator does not enter a restart command. Specify automatic restart by including AUTO=Y in the JCL.

When AUTO=Y is specified in the startup procedure, you cannot cold start IMS or modify any restart options before IMS restarts.

You can implement normal restart procedures easily, and generally do not need to worry about it. However, an operator error could adversely affect system integrity in the following situations:

- Specifying cold start after a **/CHECKPOINT FREEZE | DUMPQ | PURGE** command when some messages are not completely processed
- Failing to supply the proper security options

Restriction: Do not change the IMS operating environment between shutdown and normal or emergency restart. For example, if you change from a nonshared-queues environment to a shared-queues environment, or change coupling facility structure names, IMS abends during normal or emergency restart. If you change the operating environment, you should cold start the IMS subsystem; or you can use IMS online change (**/MODIFY** command) to make changes while IMS is running.

Cold start

A cold start is required only when you first initialize IMS or after changing the nucleus during a system definition. You must also cold start IMS after scratching and reallocating shared-message-queue structures if these structures contained messages.

Request a cold start using the **/NRESTART CHECKPOINT 0** or **/ERESTART COLDSYS** command. When you specify the COLDSYS keyword, the **/ERESTART** command initiates a cold start, not an emergency restart.

Note that if AUTO=Y is specified in the startup procedure, you cannot cold start IMS or modify any restart options before IMS automatically executes either an emergency restart or warm restart, as determined by the circumstances of the prior termination.

In a non-shared-queues environment, a cold start assumes empty message queues: if any messages exist, IMS discards them. In a shared-queues environment, IMS does not discard messages on shared queues during a cold start. If any affinities exist for VTAM generic resource groups, IMS deletes them. IMS never uses information from a previous shutdown as input to a cold start.

Optionally, you can format the IMS message queue data sets, WADS, and restart data set (RDS) during a cold restart. During a cold start, IMS loads all control blocks from the system libraries.

In a shared-queues environment, any locked messages remain locked after the IMS cold start until IMS (or other Common Queue Server client) unlocks or deletes them.

IMS systems that use dynamic resource definition (DRD) can store resource definitions for application programs, databases, Fast Path routing codes, and transactions in a resource definition data set (RDDS) or the IMSRSC repository. During a cold start, IMS can be set up to automatically import the stored resource and descriptor definitions. If your IMS system is not configured to automatically import the resource definitions, you can issue the **IMPORT** command after the restart process is complete. The **IMPORT** command, as well as all of the other type-2 dynamic resource definition commands, require that the SCI and OM CSL components are started.

In order to recover changes to the runtime resource definitions across a cold start, you must export the resource definitions to an RDDS or the IMSRSC repository while IMS is up. If you do not, any changes made to resources and descriptors in the online system might be lost.

If the IMS resource definitions were not exported to the IMSRSC repository and IMS had to perform a cold start, you can create a non-system from RDDS from the IMS log using the DFSURCL0 utility with **EXPORTNEEDED** parameter. If such an RDDS was created then you can specify that RDDS on the **IMPORT** command after the cold start is complete.

During a cold start, the partition status or access state will be copied from the HALDB master. If you export the HALDB master status to the RDDS or the repository, the status of the master and its partitions will be obtained from the RDDS or the repository.

If an error occurs while trying to access an RDDS during the automatic import process, automatic import terminates abnormally. IMS either continues with the cold start process without any runtime resource definitions defined, or IMS stops the cold start process. If an error occurs during automatic import processing because of an invalid resource or descriptor definition, one of the following situations occurs:

- Automatic import processing terminates abnormally, and IMS cold start terminates abnormally with a U3397 abend.
- Automatic import processing continues, and the resource in error is marked and given a not-initiated status (NOTINIT).

The action IMS takes depends on the values specified for the RDDSERR and the IMPORTERR parameters in the DFSDFxxx member.

Related reference

Create RDDS from Log Records utility (DFSURCL0) (System Utilities)

Warm start

A warm start is the most common way to reinitialize IMS, and is the recommended way of restarting after a controlled shutdown of IMS. You should attempt a warm start only if IMS terminated in an orderly manner. You can specify a warm start by using the **/NRESTART** command.

Before starting IMS and entering the restart command, you must determine whether IMS should:

- Reload MSDBs
- Format specific system data sets during restart
- Activate specific security functions during restart

The following situations require you to reallocate and reformat specific system data sets after a controlled IMS shutdown:

- When a message queue data set is full (indicated by messages DFS206, DFS207, or DFS208) and IMS was shut down by an internally-generated **/CHECKPOINT DUMPQ** command.
- When a write error occurred on the restart data set (RDS) during checkpoint or restart (indicated by message DFS3127I).
- When you need to change the amount of space allocated to the WADS or a write error occurs on the WADS during normal operation. You should change the amount of space allocated to the WADS only after a controlled IMS shutdown. You can replace a WADS that encountered a write error during normal restart (**/NRE**) or emergency restart (**/ERE**). In any case, when IMS uses a new WADS, you must format it during restart.
- When read or write errors occur on one of the MSDB data sets during normal operation (indicated by messages DFS2718I or DFS2722I).

For other error situations that require reallocation and formatting of some or all of the system data sets, you must use the **/ERE** command.

During a warm start, IMS:

- Reestablishes the status of the control region using control blocks that were logged at termination
- Optionally restores local message queues if IMS logged them (you issued a **/CHE DUMPQ | PURGE** command)
- Restores in-doubt units of work so they can later be resolved

Unlike emergency restart, you might have to manually release any transactions on the suspend queue by:

- Re-executing the transaction by issuing a **/START TRAN** *tranname* command. This command releases a specific transaction.
- Issuing a **/DEQUEUE SUSPEND** command. This command releases all transactions on the suspend queue.

When using XRF with VTAM Multinode Persistent Session (MNPS), IMS operates in an XRF environment and must use two ACBs: the APPLID ACB and the MNPS ACB. During a controlled shutdown, IMS closes the MNPS ACB and the APPLID ACB. Closing the MNPS ACB terminates MNPS tracking.

After a normal restart that follows this controlled shutdown, VTAM no longer maintains persistent sessions. Therefore, when IMS opens the MNPS ACB, it does not need to recover any previous sessions.

Related concepts

XRF and VTAM generic resources (System Administration)

Emergency restart

IMS requires an emergency restart whenever it terminates without a controlled shutdown. When restarted, IMS restarts from the point of failure. You can initiate an emergency restart using the **/ERESTART** command.

With an emergency restart, IMS:

- Backs out DL/I in-flight units of work
- Applies committed but unwritten DEDB changes to the database
- Retains any in-doubt units of work for IMS subsystems connected to a CCTL or to Recoverable Resource Management Services (RRMS), the z/OS syncpoint manager.
- Resolves any unfinished External Subsystem Attach Facility (ESAF) units of work with external subsystems
- Releases locked messages in a shared-queues environment
- Uses checkpoint and X'22' log records to recover MODLBKS resources

Before starting IMS and entering the emergency restart command, you need to know the following:

- Whether system data sets must be reallocated and formatted during restart.
- Whether IMS has cleaned up its resources. You can check for message DFS627I or DFS627W to determine the previous ending status of the IMS resource termination manager. These messages tell you if IMS successfully completed resource cleanup processing. Successful resource cleanup guarantees IMS restart without requiring a z/OS restart.

After an IMS failure, you can use the ESAF In-Doubt Notification exit routine (DFSFIDN0) to determine in-doubt units of work during restart of the failed IMS. You can resolve the work outside of IMS.

During an emergency restart, IMS restores the system to the status it had at the last system checkpoint. Databases and areas reflect their content at the last sync point for each dependent region. The local message queues reflect their content at the time of the failure. And the system has just taken a new checkpoint. When you restart the dependent regions, IMS reschedules the previously processing programs. IMS also releases any transactions on the suspend queue.

If the backout of any database fails during an emergency restart, IMS identifies the database by issuing DFS981I and the database remains stopped after restart completes. After IMS is running again, you might able to complete the backout by entering a /START DB or UPDATE DB START(ACCESS) command; however, depending on the root cause of the backout failure, you might need to take additional steps to correct the problem.

After an IMS failure when XRF is used with VTAM MNPS, VTAM continues to maintain persistent sessions until the timer specified by the PSTIMER= keyword (specified in DFSDCxxx IMS.PROCLIB member) expires. If there is no XRF takeover, then VTAM might continue to maintain active persistent sessions when the MNPS ACB is opened after emergency restart. If this occurs, IMS issues the command **SETLOGON OPTCD=NPERSIST**, which tells VTAM to drop all persistent sessions being maintained. This is consistent with emergency restart processing when the XRF USERVAR= keyword is used, which also has no session recovery. IMS then issues the command **SETLOGON OPTCD=PERSIST** to start the normal session persistence for all new sessions on the MNPS ACB.

If an emergency restart fails, you do not have to cold start the entire IMS subsystem. The following table shows the various emergency restart cold start commands and how they affect an IMS system.

Table 19. Effects of emergency restart commands on an IMS subsystem

Emergency restart command	Effect on the IMS DB or DBCTL subsystem	Effect on the IMS TM or DCCTL subsystem
/ERESTART COLDBASE	Cold restart	Emergency restart
/ERESTART COLDCOMM	Emergency restart	Cold restart
/ERESTART COLDSYS	Cold restart	Cold restart

- The **/ERE COLDBASE** command performs a cold start of the DB portion of an IMS DB/DC subsystem.

  If you use this command, you are responsible for the recovery of the databases. IMS does not redo committed DEDB updates, and does not back out in-flight updates for DL/I databases. IMS identifies and stops databases that have in-doubt data or that need backout or recovery. You can backout in-flight DL/I data by running the Database Batch Backout utility, for which you should close (and, optionally, archive) the OLDS.
- The **/ERE COLDCOMM** command performs a cold start of the TM portion of an IMS DB/DC subsystem.

  This command initializes the message queues, recovers DEDBs, reloads MSDBs, and backs out in-flight changes to DL/I databases. At the same time, IMS maintains all existing in-doubt data.
- The **/ERE COLDSYS** command allows you to cold start both DB and TM.

  This command is essentially a combination of both **/ERE COLDBASE** and **/ERE COLDCOMM**, but it does not read the OLDS. Therefore, closing the OLDS is essential. The processing described for both the**/ERE COLDBASE** and **/ERE COLDCOMM** commands also pertains to **/ERE COLDSYS**.

Of the three forms of emergency restart command, you will likely use**/ERE COLDCOMM** most often.

If an emergency restart (**/ERE**) fails and a subsequent emergency restart (**/ERE COLDBASE** or **/ERE COLDCOMM**) also fails, you must issue the **/ERE COLDSYS** command to restart IMS. You must also close (and optionally, archive) the last OLDS that was used online before performing this **/ERE COLDSYS** restart because the log must be available in case database recovery is necessary.

Restriction: Do not add a Fast Path DBD to the active ACBLIB between an abnormal termination and an emergency restart. If you add a Fast Path DBD to the active ACBLIB after abnormal termination of IMS, it is not accessible after the emergency restart. If you make changes to a Fast Path DBD in the active ACBLIB after the last warm start or online change (**/MODIFY**), an emergency restart may fail with abend U0168.

If an IMS database encounters an I/O error and the Extended Specified Task Abnormal Exit (ESTAE) routine is not able to execute, then IMS does not create an EEQE and does not lock the error block. Consequently, an IMS emergency restart does not detect the bad block. In this situation, you might need to use the OVERRIDE keyword on the **/ERE** command or set the abnormal termination flag in the RECON data set for the failing subsystem.

Related concepts

"Emergency restarts and dynamic resource definition" on page 81
IMS systems that use dynamic resource definition automatically recover the resources and runtime descriptor definitions during emergency restart by processing the checkpoint records and X'22' log records during an emergency restart, unless the COLDSYS parameter is specified for the /ERESTART command.

Related reference

ESAF In-Doubt Notification exit routine (DFSFIDN0) (Exit Routines)

Defer of BMP backout programs to speed emergency restart

After a system-wide failure, IMS backs out the interrupted online programs as it restarts. This backout must be complete before regular work can continue. You can request postponement of BMPs by specifying the NOBMP keyword on the **/ERESTART** command.

This backout hastens restart if either of the following conditions is true:

- A BMP had run a long time since its last commit point and needs large quantities of data backed out
- The program has a low priority and could have its restart postponed with no trouble

If you postpone BMP backout, you must run the Database Batch Backout utility some time before you rerun the BMP, or before you put the affected database back online (in case other programs need it).

If you register the affected database with DBRC, DBRC prevents programs from using that database before a backout is performed.

IMS restart and global resource status in an IMSplex that uses RM

If you are operating in an IMSplex environment that uses the Resource Manager (RM) component of the Common Service Layer, you need to understand how restarting an IMS system affects both the global and local status information of the IMS resources that the restarting IMS system shares with the rest of the IMSplex.

The first IMS system to start in an IMSplex sets the PLEXPARM values that the IMS system knows about in RM. Subsequent IMS systems that start in the IMSplex obtain the PLEXPARM values by reading the PLEXPARM entry in RM. Subsequent IMS systems either use the PLEXPARM entry in RM if it has a value or update the PLEXPARM value in RM if they introduce a PLEXPARM variable.

When an IMS system restarts, if global status is maintained for a resource type, the type of restart determines how the global status is applied at the restarting IMS. For example:

- For cold starts, the global status of databases, areas, and transactions takes precedence.
- For warm starts, the restarting IMS system first applies the local status saved in the log records and then applies the global status stored in RM only if it changed while the IMS system was down.

- For emergency restarts, the restarting IMS system first applies the local status saved in the log records and then applies the global status stored in RM only if it changed while the IMS system was down.
- For COLDSYS emergency restarts, the global status of databases, partitions, areas, and transactions takes precedence.
- For COLDBASE emergency restarts, the global status of databases, partitions, and areas take precedence.
- For COLDCOMM emergency restarts, the global status of transactions takes precedence.

Any error reading the global status of resources in RM during IMS initialization results in a message DFS3500W that states that the maintenance in RM of the global resource status is disabled for the specified resource type.

Before attempting to restart IMS again, take one of the following actions:

1. Cancel IMS or shutdown IMS after restart completes. After the error condition reported in the DFS3308W message is fixed, restart IMS. IMS will be synchronized with RM status, and no additional action is required.
2. Allow the IMS restart to complete, then take the following actions:
   - Issue the appropriate **UPDATE IMS SET(PLEXPARM)** command to enable global status at this IMS.
   - Issue the appropriate **QRY TRAN|DB|AREA SHOW(GLOBAL)** commands to see the resources with global status.
   - Issue appropriate command with the SCOPE(ACTIVE) parameter to ensure that the IMS is synchronized with RM status.

Related concepts

"Starting IMS systems and global command status" on page 94
Global command status is the status of a shared IMS resource as set by online commands and as seen by all IMSplex members that have the resource defined.

How to perform a subsystem restart with the z/OS Automatic Restart Manager

You can use the z/OS Automatic Restart Manager (ARM) to restart a subsystem or a job after a failure of the subsystem or environment in which it is running. IMS supports the z/OS Automatic Restart Manager in the DB/DC, DBCTL, DCCTL, and XRF environments.

The following subsystems also use the z/OS Automatic Restart Manager:

- The IRLM subsystem
- The CQS subsystem
- The CSL Open Database Manager (ODBM)
- The CSL Operations Manager (OM)
- The CSL Resource Manager (RM)
- The CSL Structured Call Interface (SCI)

IMS batch environments and IMS utilities do not support the z/OS Automatic Restart Manager.

After a failure, the z/OS Automatic Restart Manager will restart only the IMS control region and the CQS subsystem (each independently of the other). Other regions (such as DLISAS and DBRC) are in turn restarted by the IMS control region, and IMS dependent regions are typically restarted using automation.

In an XRF environment, the active IMS subsystem is automatically de-registered from the z/OS Automatic Restart Manager while the alternate IMS subsystem is in its tracking phase. Thus, the active subsystem is not be restarted after a failure; instead, the XRF alternate takes over as expected.

The z/OS Automatic Restart Manager ignores AUTO=N in the IMS startup parameters.

If an IMS or CQS subsystem abends before it completes restart, it is de-registered from the z/OS Automatic Restart Manager so that the abend can be fixed before attempting restart again.

If an IMS or CQS subsystem that is registered with the z/OS Automatic Restart Manager is canceled by the z/OS operator (using the **CANCEL** command), that IMS subsystem is not restarted by the z/OS Automatic Restart Manager unless the operator includes the ARMRESTART keyword on the command.

Related concepts

The z/OS Automatic Restart Manager (ARM) (System Administration)

BMP restart after system failure

IMS does not automatically restart BMPs after a system failure. You must start them yourself after IMS has completed its restart processing. The way in which you restart a BMP depends on whether its last checkpoint records exist in the OLDS.

When the last checkpoint records are in the OLDS, you can specify one of four values during restart:

- The 8-byte checkpoint ID
- The 14-byte time stamp ID IIIIDDDHHMMSSTHMIJU+0000 from message DFS0540I, where:
  - IIII is the region ID
  - DDD is the day of the year
  - HHMMSSTHMIJU+0000 is the time in hours, minutes, seconds, and tenths, hundredths, thousandths, ten-thousandths, hundred-thousandths, and millionths of a second, plus the local time zone offset
- The value LAST

Specify this value as an EXEC parameter. When you specify LAST, you must not change the BMP's job name, PSB name, or program name; IMS uses these names to locate checkpoint information for the BMP.

Restarting a BMP using XRST

If CKPTID=LAST is coded, IMS can automatically locate the last checkpoint taken by the BMP under most conditions if all of the following requirements are met:

- The BMP is restarted with the same job name, the same PSB, and the same program name used when it ended abnormally.
- The BMP restarts on the same IMS system that it ended abnormally on, and this system was not cold-started after the BMP abend occurred.
- If the restart checkpoint is no longer available on an OLDS data set, the SLDSREAD Logger function must be available . In this case, the checkpoint is on an archived SLDS data set.

If these requirements are met, an IMSLOGR DD statement that points to the log data set is not required. Coding an IMSLOGR DD statement and using IMS automatic checkpoint location are mutually exclusive processes. If an IMSLOGR DD statement is specified, IMS does not use the automated process to locate the checkpoint, but only searches the IMSLOGR concatenation. If the required checkpoint is not found in the concatenation, the BMP ends abnormally with abend code U0102.

If these requirements are not met, an IMSLOGR DD statement that points to the correct log data set is required. Automatic checkpoint location is restricted to CKPTID=LAST. If you use a checkpoint value, you must code the appropriate IMSLOGR DD statement.

Important: If you do not specify CKPTID=LAST, the data updated between the specified checkpoint and the actual last checkpoint taken has been committed by IMS.

If the program issued XRST and CHKP calls, you can restart it from the last checkpoint by specifying the checkpoint ID in the JCL and supplying, as input, the SLDS created in the program's previous execution.

Restarting batch jobs

Before you restart a DL/I batch job, ensure all database changes have been backed out. IMS might have backed out the database changes dynamically if a pseudo-abend occurred and you specified **BKO**=Y in the JCL and the data set resides on DASD. Otherwise, run the Database Batch Backout utility.

About this task

To restart a batch job, specify the ID of its last checkpoint in the JCL and supply, as input, the SLDS created in the program's previous execution. The input is specified on the IMSLOGR DD statement.

The batch job that is performing the restart must have the same job name as the original. If you do not use the same job name for the restart, IMS cannot find the checkpoint and the job fails with a U0102 error.

Reconnecting CCTLs or ODBA application programs

IMS DB system shutdown or failure generally does not cause the termination of a CCTL or ODBA application program, but it does disconnect the CCTL or ODBA application program. The CCTL or ODBA application program can automatically reconnect after an IMS system termination, or a CCTL operator can manually reconnect it.

Related reference

CCTL exit routines (Exit Routines)

Chapter 3. Monitoring IMS

You can monitor IMS by using various IMS commands, utilities, tools, and user exits.

Related concepts

Tools for detailed monitoring (System Administration)

Monitoring the system

In order to gather problem determination and performance information, you need to monitor the status of the system on a regular schedule. For example, to determine if you should start an extra message region, you might monitor the status of the queues during peak load.

You can determine the current status of the system by issuing **/DISPLAY STATUS** or **QUERY** commands, specifying the appropriate keywords. You can monitor the status of the IRLM by using the z/OS command **MODIFY***irlmproc***,STATUS**.

You can use traces enabled by commands such as **/TRACE**, **UPDATE TRACE** (type-2 trace tables), **UPDATE TRAN**, and **UPDATE MSLINK** to help diagnose system operation problems. These commands turn on or off various IMS traces, which record use of IMS control blocks, message queue and I/O line buffers, save area sets, and MSC links.

IMS records trace information on the IMS log unless you request that the traces be recorded on an external trace data set. You can also use the **/TRACE** command to trace locking activities and to start and stop the IMS monitor.

You can use the IMS Monitor user exit (IMSMON) to access IMS Monitor data for your own performance monitoring tool or utility.

You can use the Operations Manager (OM) audit trail log to see the commands, the command responses, and the selected system messages that are routed through OM. When OM audit trail logging is enabled, the OM audit trail log is saved to log stream in the z/OS System Logger, as specified on the AUDITLOG= keyword of the IMSPLEX parameter in the OM initialization parameters PROCLIB member (CSLOIxxx). You can use TSO SPOC to view the OM audit trail log.

Related concepts

“User access problems” on page 152

The task of determining and responding to user access problems can be done by the IMS MTO, the network support group, or a user liaison group.

IMS Trace Facility (System Administration)

IMS Monitor (System Administration)

Collecting and interpreting IMS monitoring data (System Administration)

Related reference

IMS Monitor user exit (IMSMON) (Exit Routines)

Candidate subsystem messages for monitoring

Certain messages that are issued by IMS and its component subsystems can be critical to detecting problems early and identifying their root cause.

The following sections contain example lists of subsystem messages that are candidates for monitoring in the IMS environment.

Important: Use the lists only as an example when determining which messages to monitor in your own IMS environment. The lists contain only a small subset of all of the messages your installation might need to monitor. The lists also might contain messages that your installation does not need to monitor.

The lists are organized into the following sections:

- Messages that are issued in all IMS environments. These are typically messages that are issued by the IMS control region or system components.
- Messages that are issued by IMS DB components. These messages can be issued only in DB/TM and DBCTL environments.
- Messages that are issued by IMS TM components. These messages can be issued only in DB/TM and DCCTL environments.

Some sections contain separate lists for optional components that run in a given environment.

Messages to monitor in all IMS environments

The following messages are related to IMS system components and processing, so can be issued in all IMS environments.

- DFS0414I CONTINUING WITH ALTERNATE LOG
- DFS629I IMS ttt TCB <action> IMS|SYS sss|uuuuvvvvIMS ttt TCB <action> IMS|SYS sss|uuuuvvvv or IMS BATCH REGION ABEND - IMS|SYS uuuu or PSW AT ERROR = hhhhhhhh hhhhhhhh or MODID = cccccccc EPA = aaaaaaaa or IMS DBC REGION ABEND
- DFS0738X ERROR TERMINATING OLDS RC=xx dddddddd nnnnnnn
- DFS2867A EXTERNAL TRACE NOT USABLE, REPLY "Y" TO USE OLDS, "N" TO TRACE INCORE
- DFS3258A LAST ONLINE LOG DATA SET IS BEING USED - NEED ARCHIVE or SYSTEM WAITING FOR AN ONLINE LOG DATA SET - NEED ARCHIVE
- DFS3262E NO DATA SET AVAILABLE FOR LOG WRITE AHEAD
- DFS3306A CTL REGION WAITING FOR csltype
- DFS3785E DIAGNOSE AWE INITIALIZATION FAILED – reason_text
- DFS3786E DIAGNOSE AWE PROCESSING ERROR – reason_text
- DFS3787E DIAGNOSE SYSOUT PROCESSING ERROR – reason_text
- DFS555I RAN tttttttt ABEND (SYSID sss); REASON=reason; MSG IN PROCESS: xxxx (up to 78 bytes of data) time-stamp
- DSP0014I DYNAMIC ALLOCATION FAILED FOR RECONn RETURN CODE=xx REASON CODE=xxxx
- DSP0027I ddname DATA SET IS FULL
- DSP0256I NO SPARE RECON DATA SET AVAILABLE
- DSP1135A SCI REGISTRATION FAILED, IMSPLEX NAME=nnnnn, RC=xxxxxxxx, RSN=yyyyyyyy, JOB=jjjjjjj
- DSP1141I RECON LOSS NOTIFICATION RECEIVED, JOB=jjjjjjj
- DSP1145I RECON LOSS NOTIFICATION NOT SENT, JOB=jjjjjjj
- DSP1147I DBRC REGION WAITING FOR SCI, IMSPLEX NAME=ppppp
- DSP2002E DBRC INITIALIZATION ERROR IN modulename servicename RC=rc detail

IMSRSC repository messages:

- DFS4409A REPOSITORY CHANGE LIST IS NOT ACCESSIBLE - IMS RESOURCES MAY BE OUT OF SYNC
- DFS4411E REPOSITORY CHANGE LIST PROCESSING FAILED RC=rc RSN=rsn
- DFS4413E REPOSITORY CHANGE LIST PROCESSING FAILED FOR RSCNAME=rscname RSCTYPE=rsctype CC=cc

Messages to monitor in DB/TM and DBCTL environments

The following messages are related to IMS DB components and processing, so can be issued only in DB/TM and DBCTL environments.

- DFS047A UNABLE TO OBTAIN AUTHORIZATION FOR DATA BASE xxxxxxxx. REASON CODE = zz. PSB= psbname.

- DFS0565I cccc COMMAND NOT PROCESSED DB=xxxxxxxx IN USE BY PSB=psbname, REG=region-number
- DFS691I WAITING FOR CTL xxxx - yyyyyyyy
- DFS0730I UNABLE TO OPEN OR CLOSE DATASET WITH DDNAME ddname FOR REASON x, yy, z DATABASE dbdname programid
- DFS0811A UNABLE TO OBTAIN GGG GIGABYTES OF 64-BIT STORAGE FOR THE ACB POOL. REASON=xxxx
- DFS0844I modulename dbname DATASET FULL, DDNAME=ddname
- DFS0845I dbname DATASET LIMIT REACHED, DDNAME=ddname
- DFS981I DBD=dbdname WITHIN PSB=psbname STOPPED DUE TO (BACKOUT FAILURE|ERE NOBMP START|COLDBASE START|REMOTE TAKEOVER)
- DFS2011I IRLM FAILURE - IMS QUIESCING
- DFS2012I GLOBALLY SHARED DATA BASE|AREA - dbdname|areaname STOPPED or SHARING DATA BASE(S) STOPPED BECAUSE DATA SHARING DISCONTINUED
- DFS2503W DYNAMIC ALLOCATION|DEALLOCATION| CREATION| DELETION FAILED FOR DATA SET NAME xxxxxxxx.xxxxxxxx.xxxxxxxx DATABASE NAME dbdname REASON CODE yyyyy
- DFS3867A NOTIFY REQUEST TO SET AN ADS TO AVAILABLE STATUS FAILED FOR AREA=xxxxxxxx DD=yyyyyyyy
- DFS4164W FDR FOR (imsid) TIMEOUT DETECTED DURING LOG AND XCF SURVEILLANCE

Messages to monitor in DB/TM and DCCTL environments

The following messages are related to IMS TM components and processing, so can be issued only in DB/TM and DCCTL environments.

- DFS271 UNABLE TO LOAD ERROR MESSAGE OUTPUT DESCRIPTION
- DFS2013 NUMBER OF RECORDS IN QBLKS DATA SET HAS EXCEEDED UPPER THRESHOLD
- DFS2014 NUMBER OF RECORDS IN SMSGQ DATA SET HAS EXCEEDED UPPER THRESHOLD
- DFS2015 NUMBER OF RECORDS IN LMSGQ DATA SET HAS EXCEEDED UPPER THRESHOLD
- DFS2088I APPC/OTMA SMQ ENABLEMENT INACTIVE. REASON = xxx.
- DFS3492W APPC/IMS TIMEOUT LIMIT REACHED FOR LU luname, TP-ID tp-id, variable text
- DFS554A TRAN tttttttt ABEND (SYSID sss); REASON=reason; MSG IN PROCESS: xxxx (up to 78 bytes of data) time-stamp

OTMA messages:

- DFS1988W OTMA input messages from member yyyyyyyy have reached xx% of the maximum active input message limit zzzz
- DFS1989E OTMA input messages from member yyyyyyyy have reached the maximum active input message limit zzzz
- DFS2088I APPC/OTMA SMQ ENABLEMENT INACTIVE. REASON = xxx.
- DFS3428W THE TOTAL OTMA INPUT MESSAGES(TIB) HAVE REACHED xx% OF THE GLOBAL LIMIT zzzz
- DFS3429E THE TOTAL OTMA INPUT MESSAGES(TIB) HAVE REACHED THE GLOBAL LIMIT zzzz
- DFS3492W APPC/IMS TIMEOUT LIMIT REACHED FOR LU luname, TP-ID tp-id, variable text
- DFS3494E OTMA HAS TIMED OUT FOR TMEMBER/TPIPE xxxx/yyyy variable text
- DFS3495W OTMA HAS BEEN WAITING FOR AN ACK FROM TMEMBER/TPIPE xxxx/yyyy FOR OVER zzzz SECONDS.
- DFS4382W THE TOTAL OTMA TPIPE COUNT nnnnn HAS REACHED 80% OF mmmmm for MEMBER name

- DFS4383E THE TOTAL OTMA TPIPE COUNT nnnnn HAS REACHED 100% OF mmmmm for MEMBER name
- DFS4385W THE GLOBAL OTMA TPIPE COUNNT nnnnn HAS REACHED 100% OF kkkkk

Related concepts

"Processing of messages, commands, and command responses" on page 257
The automated operator interface (AOI) process system messages, command, and command responses. AOI lets an application program, using DL/I calls, issue most IMS operator commands and receive command responses.

IMS messages overview (Messages and Codes)

DFS messages (Messages and Codes)

Non-DFS messages (Messages and Codes)

Monitoring IMS Connect connections

IMS manages TCP/IP connections through IMS Connect, which serves as the TCP/IP gateway to IMS. The IMS Connect function can also be viewed as a TCP/IP socket server.

TCP/IP connections to IMS Connect can be classified into the following types:

- IMS DB client connections, where the client is accessing IMS DB.
- IMS TM client connections, where the client is accessing IMS TM.
- IMS-to-IMS TCP/IP connections, where one IMS Connect instance connects to another IMS Connect instance for communications between two geographically remote IMS systems.

In addition to the TCP/IP connections that can be made to IMS Connect, you must also consider the connections between IMS Connect and the IMS server. IMS Connect runs in a separate address space from IMS and connects with certain components of IMS by using different communications methods.

For IMS DB client support, IMS Connect uses the Structured Call Interface (SCI) component of the IMS Common Service Layer (CSL) to connect to the IMS Open Database Manager (ODBM) component of CSL. ODBM communicates directly with IMS DB through the IMS Open Database Access (ODBA) interface.

IMS Connect also uses SCI for direct communications with IMS when IMS-to-IMS TCP/IP connections are used to support Multiple Systems Coupling (MSC) TCP/IP links.

For IMS TM client support, IMS Connect uses the z/OS cross-system coupling facility to connect to the Open Transaction Manager (OTMA) component of IMS.

To monitor and verify the status of each type of the connection managed by IMS Connect, you might need to use different commands or facilities.

The following subsections describe only some of the commands that you can use to check the status of IMS Connect connections.

Related concepts

"MSC TCP/IP link operations" on page 203
Operating procedures for MSC TCP/IP links are generally the same as the operating procedures for MSC VTAM links; however, a few differences do exist.

Checking port TCP/IP connection status

You can check the status of all of the TCP/IP connections on a port.

About this task

To see the status of all of the TCP/IP connections on a port:

Procedure

Issue any one of the following IMS Connect commands:

- In the IMS type-2 command format, **QUERY IMSCON** TYPE(PORT)
- In the WTOR command format, **VIEWPORT**
- In the z/OS MODIFY command format,**QUERY PORT**

Related reference
QUERY IMSCON commands (Commands)
IMS Connect WTOR commands (Commands)
IMS Connect z/OS commands (Commands)

Checking client TCP/IP connection status

You can check the status of the TCP/IP connections for one or more IMS Connect clients.

About this task

To check the status of TCP/IP connections for specific clients:

Procedure

Issue the command **QUERY IMSCON** TYPE(CLIENT) in the IMS type-2 command format.

No equivalent WTOR or z/OS MODIFY command exists for **QUERY IMSCON** TYPE(CLIENT); however, the WTOR **VIEWPORT** command and the z/OS MODIFY **QUERY PORT** command provide some client-related connection information.

Related reference
QUERY IMSCON commands (Commands)
IMS Connect WTOR commands (Commands)
IMS Connect z/OS commands (Commands)

Checking remote IMS Connect TCP/IP connection status

You can check the status of TCP/IP connections with a remote IMS Connect instance.

About this task

To check the status of a TCP/IP connection to another instance of IMS Connect:

Procedure

Issue any of the following IMS Connect commands:

- In the IMS type-2 command format, **QUERY IMSCON** TYPE(RMTIMSCON) or **QUERY IMSCON** TYPE(SENDCLNT)
- In the WTOR command format, **VIEWRMT**
- In the z/OS MODIFY command format, **QUERY RMTIMSCON**

Related reference
QUERY IMSCON commands (Commands)
IMS Connect WTOR commands (Commands)
IMS Connect z/OS commands (Commands)

Checking IMSplex member SCI connection status

You can check the status of connections between IMS Connect and SCI within an IMSplex.

About this task

IMS Connect uses SCI to communicate a number of different IMSplex members for a number of different purposes. IMS Connect provides different commands to check the status of an IMSplex and the status of the IMSplex members.

Procedure

- For IMSplex status, issue any of the following IMS Connect commands:
  - In the IMS type-2 command format, **QUERY IMSCON** TYPE(IMSPLEX)
  - In the WTOR command format, **VIEWIP**
  - In the z/OS MODIFY command format, **QUERY IMSPLEX**
- For IMS connection status for MSC TCP/IP link support, issue any of the following IMS Connect commands:
  - In the IMS type-2 command format, **QUERY IMSCON** TYPE(MSC)
  - In the WTOR command format, **VIEWMSC**
  - In the z/OS MODIFY command format, **QUERY MSC**
- For ODBM connection status for IMS DB access, issue any of the following IMS Connect commands:
  - In the IMS type-2 command format, **QUERY IMSCON** TYPE(ODBM)
  - In the WTOR command format, **VIEWIA**
  - In the z/OS MODIFY command format, **QUERY ALIAS**

Related reference

QUERY IMSCON commands (Commands)
IMS Connect WTOR commands (Commands)
IMS Connect z/OS commands (Commands)

Checking XCF data store connection status for IMS TM clients

You can issue variations of the IMS and IMS Connect commands to determine the status of XCF data store connections for IMS TM clients.

About this task

In the following examples, the XCF connection between IMS Connect and OTMA is defined by the following DATASTORE configuration statement:

```
DATASTORE=(ID=DSNAME,MEMBER=ICONNAME,TMEMBER=IMSNAME,GROUP=GRPNAME...)
```

Procedure

- From IMS, use the following commands to check the status of IMS Connect:
  - **/DIS OTMA**

    When IMS Connect is ready for use, the output of the **/DIS OTMA** command appears as shown in the following example:

```
GROUP/MEMBER  XCF-STATUS  USER-STATUS    SECURITY TIB   INPT SMEM  DRUEXIT  T/O  ACEEAGE
GRPNAME
-IMSNAME      ACTIVE      SERVER         FULL*
-ICONNAME     ACTIVE      ACCEPT TRAFFIC
```

```
* CHECK, FULL, NONE or PROFILE depending on the OTMA Security
  setting (for example, enter /SEC OTMA NONE on the MVS system
  console to turn off RACF security for IMS OTMA clients). FULL
  is the default setting for OTMA security at IMS startup.
```

- **/DISPLAY TMEMBER** *tmembername* TPIPE

  For example, the command input /DISPLAY TMEMBER CLIENT1 TPIPE ALL might produce the following output:

```
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*
```

- From IMS Connect, you can use the following WTOR-format commands to display status:

  VIEWHWS
  Displays information about data stores, ports, and IMSplex.

  Tip: If you use the SUMMARY option for the **VIEWHWS** command, you can display IMS Connect information without the detailed output for each socket.

  VIEWDS
  Displays information about data stores.

  Alternatively, you can issue the following IMS type-2-format commands through the OM API to display the status:

  QUERY IMSCON TYPE(CONFIG)
  Displays the status and activities of IMS Connect. Unlike the **VIEWHWS** command, individual resources such as port and data store are not displayed.

  QUERY IMSCON TYPE(PORT)
  Displays the status and activities of one or more ports defined to IMS Connect.

  If you specify the SHOW(CLIENT) keyword, an additional line is displayed for each client associated with the specified port.

  QUERY IMSCON TYPE(DATASTORE)
  Displays the status and activities of one or more data stores defined to IMS Connect.

- Enter the following IMS commands if you fail to receive a response to a request message sent from an IMS TM client (or to check the readiness of the host data store):

  /DIS A REG
  You can use this command to verify that the dependent region where your host application runs is properly configured and ready to accept messages:

```
REGID JOBNAME   TYPE  TRAN/STEP PROGRAM  STATUS   CLASS
    1    Job1   TP                       WAITING  1,  2,  3,  4
```

  /DIS TRAN TranName
  You can use this command to verify the class and status of a transaction and whether or not a transaction is currently queued for processing:

```
TRAN     CLS ENQCT   QCT   LCT  PLCT CP NP LP SEGSZ SEGNO PARLM    RC
TRANNAME   2     1     1 65535 65535  8  8  8     0     0  NONE     0
```

  QCT is the number of transactions that are currently queued. ENQCT includes transactions that have been dequeued (processed), as well as those that are currently on the queue.

- If you receive an IMS Connect error message on either the z/OS system console or in the response message at the client when you issue IMS or IMS Connect commands, the format of IMS Connect message numbers is HWS*xnnnn*, where the *x* represents an alphabetic character and *nnnn* represents a 4-digit number.

- IMS Connect requires that all active clients have unique client names. IMS TM Resource Adapter creates a unique CLIENTID, which identifies each request that an application makes to execute an IMS transaction. If you are using TCP/IP clients that access IMS TM other than IMS TM Resource Adapter, you must ensure that those clients each use a unique client name. This client name value is displayed for a client in the "CLIENT=" or "ORIGIN=" fields.

  When IMS Connect receives an initial request from IMS TM Resource Adapter with a blank Client ID, IMS Connect generates a unique Client ID to be used for that TCP/IP persistent socket. IMS Connect returns this Client ID to IMS TM Resource Adapter on any reply message.

Results

Important: There are OTMA restrictions on IMS commands.

Related reference

OTMA restrictions and requirements (Communications and Connections)
/DISPLAY OTMA command (Commands)
/DISPLAY TMEMBER command (Commands)
QUERY IMSCON commands (Commands)
IMS Connect WTOR commands (Commands)
IMS Connect z/OS commands (Commands)

IMS system log utilities

The system log data sets are a basic source for statistics about the processing performed by the online system. Individual log record types contain data that can be analyzed in many ways. For example, you can select and format all activity pertaining to a specified user ID, or about IMS pools.

IMS provides several utilities to assist with extracting log records from the system log. These utilities also assist with reducing and merging data that is spread across several log data sets.

In addition to the following utilities, IMS Performance Analyzer for z/OS provides a comprehensive suite of reports to help you manage the performance and resource utilization of your IMS systems.

Related concepts

IMS Performance Analyzer for z/OS overview

File Select and Formatting Print utility (DFSERA10)

You use this utility to examine message segments or database change activity in detail. This utility prints the contents of log records contained in the OLDS, SLDS, or the CQS log stream. Each log record is presented as one or more segments of 32 bytes. The printed output gives each segment in both character and hexadecimal formats.

You can specify selection criteria for a subset of the records rather than printing all records. You can also specify a starting record number and the number of records to process. You can use an exit routine to customize the selection and formatting of the log records.

Although you can use the File Select and Formatting Print Program to copy entire input logs, you can more conveniently use the Log Archive utility (DFSUARC0). You use one or more SLDSs as input and specify a user data set as output. Also, you need to specify DBRC=N0 in the EXEC statement to prevent DBRC from making entries in the RECON data set about your backup log. Making backup copies of the system log data sets can be useful to obtain an alternative input source for statistics and other monitoring activities occurring in parallel with production use of the system log.

Related reference

File Select and Formatting Print utility (DFSERA10) (System Utilities)

Log Transaction Analysis utility (DFSILTA0)

In an IMS DB/DC or DCCTL environment, you can use this utility to collect information about individual transactions, based on records on the system log. Many events are tabulated in the Log Analysis report produced by this utility, including total response time, time on the input queue, processing time, and time on the output queue.

You can select a start time for the report tabulation; analysis begins at the first checkpoint after the start time. To control how much transaction activity is tabulated, you can specify an interval (in minutes) of elapsed time from the start time before the utility ends the tabulation, or you can relate the activity reported to a number of IMS checkpoints.

You can retitle a Log Analysis report or change the sequence in which the detailed transaction lines are printed. You can sort by transaction code or by any of the fields in the report. You can also suppress printing so that the output is stored on a DASD data set.

Using this utility, you can create an output data set, in system log format, that is a copy of all or part of the input system logs. By having a copy of the system log, you can monitor system activity without impacting the use of the OLDS for recovery.

Related reference

Log Transaction Analysis utility (DFSILTA0) (System Utilities)

Statistical Analysis utility (DFSISTS0)

In an IMS DB/DC or DCCTL environment, you can use this utility to produce several summary reports. You can use these reports to obtain actual transaction loads and response times for the system. The statistics produced are dependent on the input system log data sets.

The following set of reports is produced:

- Telecommunication line and terminal (distributed traffic over 24-hour day)
- Transaction (distributed activity over 24-hour day)
- Transaction response
- Messages queued but not sent (listing by destination and by transaction code)
- Program-to-program messages (listing by destination and by transaction code)
- Application accounting
- IMS accounting

Related reference

Statistical Analysis utility (DFSISTS0) (System Utilities)

Gathering performance-related data

IMS provides the DB Monitor and the IMS Monitor that gather and format IMS performance-related data and record this data on a statistics log.

About this task

The DB Monitor is available to IMS batch systems; it can monitor the activity between application programs and databases. The IMS Monitor is available to IMS online systems. In addition to performing all of the functions of the DB Monitor, the IMS Monitor can track and record information about activities occurring in the IMS control region and data communication activities.

IMS uses the statistics produced by each Monitor to generate reports. The report programs run offline and print reports that summarize and categorize IMS activities.

The DB Monitor records performance data during execution of an IMS DB batch system. The DB Monitor can be active during the entire execution of an IMS batch job, or you can stop and restart it from the system console.

Related concepts

DB Monitor reports (System Administration)

IMS Monitor reports (System Administration)

Activating and controlling the DB Monitor

To activate the DB Monitor, specify MON=Y in the PROC statement of the batch job. When you submit the job, IMS uses parameter substitution to update the PARM field of the EXEC statement with a Y in the appropriate position.

About this task

To stop the DB Monitor, the system console operator can use the **MODIFY***jobname***,STOP** command. Message DFS2215A displays on the system console when the Monitor is inactive.

To reactivate the DB Monitor, the console operator can use the **MODIFY***jobname***,START** command.

Message DFS2216A displays on the console when the Monitor is active again.

Logging the data from the DB Monitor

IMS records the data produced by the DB Monitor on either the batch log or a separate DB Monitor log defined by the //IMSMON DD statement. You use the //IMSMON DD statement in the batch procedure to control where data is logged.

About this task

- To store the Monitor records on the batch log, either include a //IMSMON DD DUMMY statement or omit the //IMSMON DD statement entirely.
- To store the Monitor records on a separate DB Monitor log, include a valid //IMSMON DD statement.

If, for any reason, IMS cannot open the DB Monitor log data set specified on the //IMSMON DD statement, IMS displays message DFS2217I on the system console. Batch execution continues, but the Monitor is inactive.

If the DB Monitor log device encounters I/O errors, IMS displays message DFS2219I on the system console. Batch execution continues, but the Monitor is inactive.

If you want to stop the Monitor and force an end-of-volume for the DB Monitor log, use the **MODIFY***jobname***,STOPEOV** command. When you use the STOPEOV keyword, the batch region does not continue executing until the z/OS mount request for a new data set is satisfied.

Note: If you enter an incorrect job name on the **MODIFY** command, z/OS issues an error message. If you make some other error while entering the **MODIFY** command, IMS issues message DFS2218I, followed by either message DFS2215A or message DFS2216A.

Data recording and the IMS Monitor

The IMS Monitor records performance-related data during execution of the IMS online subsystem. When a significant event occurs within IMS, IMS passes relevant data to the IMS Monitor when activated. The IMS Monitor formats a record describing the event, including the time stamps, and logs the event.

Activating and controlling the IMS Monitor

If you create a DFSDCMON member in the IMS.SDFSRESL data set, you do not need a DD statement in the IMS, DBC, or DCC procedures for the IMS Monitor because IMS dynamically allocates and deallocates the IMS Monitor data set. You start and stop the IMS Monitor with the **/TRACE** command.

About this task

Otherwise, to activate the IMS Monitor, you must include a DD statement (using IMSMON as the data set name) in the IMS, DBC, or DCC procedures to specify the IMS Monitor log data set. When you include this DD statement, the IMS Monitor becomes available. If the IMS Monitor is available but inactive, processor usage is unaffected.

Monitoring using the /TRACE command

In addition to starting and stopping the IMS Monitor, you can specify the types of events to be monitored using the **/TRACE** command. You can use the **/TRACE** command to monitor various activities.

Some of the activities that you can monitor with the **/TRACE** command include:

- Telecommunication line and logical link activity
- Scheduling and termination events
- Activity between application programs and message queues
- Activity between application programs and databases (full function and Fast Path)

You can also use the **/TRACE** command to limit the monitoring to:

- Particular databases, partitions, or areas
- Particular dependent regions
- A particular interval of time

Log data from the IMS Monitor

Due to how your IMS Monitor log data set is stored, you should process the log after you stop the IMS Monitor and before restarting it again.

If the IMS Monitor log data set is on tape, IMS issues a tape mount request each time you start the IMS Monitor, and IMS rewinds the tape each time you stop the IMS Monitor. If the IMS Monitor log data set is on DASD, IMS uses the same data set each time you start the IMS Monitor.

Recommendation: Start the IMS Monitor and allow it to run for a period of time, and then stop it to write a "snapshot" of current activity on the IMS Monitor log. You must stop the IMS Monitor before you take a shutdown checkpoint in order for the report program to produce usable output.

I/O errors on the IMS Monitor log data set

If a permanent I/O error occurs on the IMS Monitor log data set, IMS stops the IMS Monitor and issues message DFS2202. In this situation, you cannot restart the IMS Monitor until you restart IMS because IMS does not close the IMS Monitor log data set until you shut down IMS.

If the problem that caused the error has not been corrected when you restart IMS, you should specify a different volume or unit for the new execution.

Monitoring IRLM

You can use the **MODIFY***irlmproc***,STATUS** command to obtain information about IRLM activity.

About this task

This command provides:

- The IMS IDs of IMS subsystems using this IRLM
- The number of locks that are held and waiting for each subsystem on this IRLM
- Identification of this IRLM: its subsystem name and IRLM number

Tracing IRLM activity

The IMS Monitor does not collect IRLM trace activity. IRLM uses the z/OS component trace (CTRACE) facility. You can use the **TRACE CT** command to run various types of sublevel traces.

About this task

DBM
Trace interactions with the identified DBMS.

EXP
Trace any exception condition.

INT
Trace member and group events other than normal locking activity.

SLM
Trace interactions with the z/OS locking component.

XCF
Trace all interactions with the z/OS cross-system coupling facility services.

XIT
Trace only asynchronous interactions with the z/OS locking component.

Chapter 4. Shutting down IMS

This approach is a common sequence for shutting down the entire online system.

Procedure

1. Stop communications based on your IMS environment.

Option	Description
Stop data communications	For an IMS DB/DC or DCCTL environment
Disconnect from the CCTL	For an IMS DBCTL environment

2. Stop dependent regions.
3. Stop MSC links.
4. Stop OTMA.
5. Stop the control region.
6. For an IMS DB/DC or DBCTL environment, stop the IRLM.
7. For a shared-queues environment, if CQS has not shut down, shut it down.
8. For an IMSplex system using CSL, shut down the CSL manager address spaces (OM, RM, repository server and SCI).

Results

The command used to shut down the control region also forces termination of data communications and the dependent regions if they have not already been terminated in an orderly way, and can also tell the CQS subsystem to shut down.

Stopping Transaction Manager

You can use the **/STOP DC** command to prevent new users from logging on to VTAM terminals. This command does not terminate existing sessions.

About this task

You can use the **/STOP LINE** and **IDLE LINE** commands to disable devices such as printers, spools, punches, disks, readers, and tapes. You can use the **/PURGE TRAN** or **UPDATE TRAN START(SCHD) STOP(Q)** command to prevent existing VTAM terminal users from entering new transactions.

To terminate a VTAM session, you can use one of the following commands, or you can wait for the end user to terminate the session.

- **/STOP NODE**
- **/CLSDST NODE**
- VTAM **VARY**
- **/IDLE NODE**

  You can use this command if you already entered a **/STOP DC** or **/CHECKPOINT** command or you can wait for the end user to terminate the session.

Recommendation: Use the **/CLSDST NODE** command to terminate a VTAM session instead of the **/STOP NODE** command. The **/STOP** command requires that you subsequently issue a **/START** command before you or a user can initiate a session. You can initiate a session using the **/OPNDST** command.

To shut down VTAM, use the **HALT NET** command (abbreviated as **Z NET**). If you issue a **Z NET** command with no keywords, you terminate VTAM, and IMS permits no new logons, but existing sessions

continue. The existing VTAM sessions must be ended by the terminal user, the VTAM operator, or the MTO. If you issue the **Z NET,QUICK** command, VTAM does not allow new sessions to be established or any additional input or output operations to take place, and IMS terminates existing sessions.

Stopping APPC

You use the **/STOP APPC** command to prevent new users or programs from allocating APPC conversations with the IMS APPC BASE LU.

About this task

The **/STOP APPC CANCEL** command prevents new users or programs from allocating APPC conversations with any IMS APPC LU. In both cases, the conversation is rejected with a `TP_Not_Available_No_Retry` return code by APPC/MVS.

Stopping OTMA

You use the **/STOP TMEMBER** command to suspend input from one or more specified OTMA client transaction members (tmembers).

About this task

The connection remains active, transactions already being processed are unaffected, conversational transactions already in progress can complete normally, and all of the OTMA protocol commands can still be processed by OTMA.

If you want to shut down OTMA completely, use the **/STOP OTMA** command.

Stopping dependent regions

You can use the **/STOP REGION** command to terminate dependent regions. You also terminate dependent regions when you shut down the control region using a **/CHECKPOINT** command.

Shutting down the IMS control region

Use the **/CHECKPOINT FREEZE|DUMPQ|PURGE** command to shut down the IMS control region. You should provide guidelines for the MTO on selecting the right kind of shutdown, either specifying the FREEZE or PURGE keyword.

Using the FREEZE keyword causes the quickest shutdown, and the PURGE keyword causes the slowest shutdown.

- **/CHECKPOINT FREEZE|DUMPQ|PURGE** immediately terminates sessions for all logical units as follows:

  FREEZE
  Immediately after current input/output message

  DUMPQ
  After blocks have been written to the log

  PURGE
  After all queues are empty
- **/CHECKPOINT [FREEZE|DUMPQ|PURGE] QUIESCE** allows all network nodes to complete normal processing before IMS shuts down.

If dynamic resource definition (DRD) is enabled for your IMS system, you must ensure that changes to your runtime resource and descriptor definitions are saved to the resource definition data set or an IMSRSC repository. If AUTOEXPORT is enabled in the DFSDFxxx PROCLIB member, the definitions will be exported automatically to the repository, the RDDS, or both, depending on the AUTOEXPORT specification and your DRD configuration, during the IMS shutdown checkpoint processing. If IMSRSC repository is enabled and AUTOEXPORT is not enabled, use the **QUERY** command with the SHOW(EXPORTNEEDED)

filter to identify any unexported resources, and export them. If IMS shuts down before dynamically defined resources can be exported, you can use the DFSURCL0 utility to create a non-system RDDS data set that contains only the resource definitions that have not been exported to the repository.

The following table illustrates what IMS does in response to each type of checkpoint command.

Table 20. System actions during checkpoint

System action	FREEZE	DUMPQ[1]	PURGE
Stop terminal input	X	X	X
Process transactions		X[2]	X
Free message regions	X	X	X
Purge real storage message queue to disk	X	X	
Log checkpoint data	X	X	X
Notify the CCTL	X		X
Wait for the CCTLs to disconnect from their resource managers	X		X
Write MSDBs to checkpoint data set	X	X	X
Write CHKPT ID to master terminal	X	X	X
Purge database buffer pool	X	X	X
Terminate message regions	X	X	X
Send queued output	X	X	X
Wait for BMP's sync point	X	X	
Wait for BMP's completion			X
Send shutdown message to terminals		X	X
Dump message queues to log		X	
Close databases	X	X	X
Close queues	X	X	X
Close log	X	X	X
Terminate IMS control region	X	X	X

Notes:

1. The **/CHECKPOINT DUMPQ** command is not valid for an IMS DBCTL environment.
2. Applies to Fast Path messages only.

Recommendations:

- Have the MTO broadcast a message to all users about 10 minutes before entering the IMS shutdown command. This lets end users, except those involved in conversations, terminate their work in an orderly way.
- If you use the IMS monitor, stop it with the **/TRACE SET OFF MON** command before using the **/CHECKPOINT** command.
- Allow active VTAM terminals to complete processing before the IMS shutdown begins.

After the shutdown process has begun, you can use the **/DISPLAY SHUTDOWN STATUS** command to list the following information:

- Communication lines and terminals that still contain active messages
- BMPs and CCTL threads that are still active

To speed up the shutdown process, you can use the **/IDLE** command to stop I/O operations on specified lines.

In the IMS DBCTL environment, IMS notifies the CCTL that IMS is shutting down. In response, the CCTL must disconnect from the IMS DBCTL subsystem. The CCTL can control how long its threads process before it disconnects, for example, it could let all threads reach a sync point first.

If IMS fails to shut down, and if logging resources are available, you must force IMS to terminate.

Exceptions:

- If no logging resources are available, the OLDS must be archived before you can force shutdown.
- IMS does not shut down if the CQS subsystem is unavailable while you are running in a shared-queues environment.

Related concepts

"Forced termination of IMS" on page 130
You normally use the **/CHECKPOINT** command to shut down IMS. However, in certain error situations such as a control region loop, you might need to force termination of IMS. In this case, use the z/OS **MODIFY** command, and be sure to request a dump of the IMS control region.

Related reference

Create RDDS from Log Records utility (DFSURCL0) (System Utilities)

Shutting down IMS by using the /CHECKPOINT commands

The way that IMS shuts down depends on which form of the **/CHECKPOINT** command you use. You can choose to specify the FREEZE, DUMPQ, or PURGE keyword, and each keyword shuts down IMS differently.

The forms of the **/CHECKPOINT** command are:

- **/CHECKPOINT FREEZE**
- **/CHECKPOINT DUMPQ**
- **/CHECKPOINT PURGE**

Subsections:

Shutting down IMS by using /CHECKPOINT FREEZE

The fastest way to terminate IMS in an orderly way is to use the **/CHECKPOINT FREEZE** command. Using the **/CHECKPOINT FREEZE** command, you allow IMS to complete input and output messages that are in transit. Then IMS:

- Stops each communication line
- Terminates VTAM sessions
- Terminates message regions as soon as the messages currently being processed have completed
- Returns status code XD to batch message regions when they next issue a DL/I checkpoint call

In a DBCTL environment, IMS notifies the CCTL of the **/CHECKPOINT FREEZE** command so the CCTL can disconnect from its connected resource managers. The CCTL controls how long its CCTL threads can continue processing before the disconnect occurs. For example, the CCTL might let all its threads reach a sync point.

During shutdown, IMS also closes message queue data sets and databases, and writes checkpoint data to the OLDS, just as for a simple checkpoint. IMS issues message DFS994, which contains the last checkpoint ID, and closes the OLDS. IMS updates the checkpoint ID table to reflect the shutdown, and writes it to the restart data set (RDS). Then IMS closes the RDS and terminates the IMS control region.

During normal shutdown, IMS tells CQS to shut down. If you do not want CQS to shut down, specify the NOCQSSHUT keyword on the **/CHECKPOINT** command. If you are shutting down a single IMS subsystem in a sysplex in order to activate a system definition, change a parameter specification, or for reasons that are not related to the CQS subsystem, you do not need to shut down the CQS address space. By leaving the CQS address space active after IMS has shut down, CQS can continue to participate in structure checkpoint, overflow, and rebuild events.

Recommendations:

- Ensure that the CQS address space is active during structure checkpoint processing. During a structure checkpoint, z/OS deletes log records that are no longer needed for structure recovery, thus reclaiming log space. If CQS is not active during the structure checkpoint, it is possible that the CQS restart log records will be deleted. In this case, the next restart for the CQS address space must be a cold start. You must respond COLD to WTOR message CQS0032A. If you choose to leave CQS up, you can reduce the number of times that CQS needs to cold start due to log record deletion.

  If CQS is active when IMS is started, IMS reconnects to the existing CQS and does not need to wait for CQS initialization and restart to complete.
- Although using the **/CHECKPOINT FREEZE** command is the fastest way to accomplish an orderly shutdown, you should not use it regularly for normal system termination. When you use the **/CHECKPOINT FREEZE** command, end users might not receive responses for extended periods of time (until you restart IMS). IMS also rejects Fast Path input messages during shutdown. If during restart, IMS must load the message queues from the log to the message queue data sets (a BUILDQ), you might need to access previously archived OLDSs (to isolate the last DUMPQ or SNAPQ checkpoint).

Shutting down IMS by using /CHECKPOINT DUMPQ

The **/CHECKPOINT DUMPQ** command requests an immediate shutdown. Using the **/CHECKPOINT DUMPQ** command, you allow IMS DB/DC or IMS DCCTL to complete input and output messages that are in transit. Then IMS:

- Stops each communication line
- Terminates VTAM sessions
- Terminates message regions as soon as the messages currently being processed have completed
- Allows Fast Path message-driven regions to process all received messages before terminating
- Returns status code XD to batch message regions when they next issue a DL/I checkpoint call

In a non-shared-queues environment, IMS writes the contents of the message queues to the system log along with checkpoint data. In a shared-queues environment, IMS does not dump the shared queues during shutdown.

The cost of executing a **/CHECKPOINT DUMPQ** command over a **/CHECKPOINT FREEZE** command is the time it takes to dump the queues; this time varies depending on the system design, transaction volumes, and activity at a specific time of day.

Shutting down IMS by using /CHECKPOINT PURGE

Using the **/CHECKPOINT PURGE** command is the most time-consuming method of terminating IMS. As the result of the **/CHECKPOINT PURGE** command, IMS stops each input communication line as soon as it receives any messages that are in transit. IMS then:

- Processes all messages in the input queue, if possible (if the transaction and program are not stopped)
- Transmits all output, if possible (if the line and terminal are not stopped)
- Stops output communication lines after it has terminated all active regions and has sent all possible output messages
- Terminates VTAM sessions

IMS writes any unprocessed input messages and any untransmitted output messages to the system log along with checkpoint data.

For an IMS DBCTL environment, the **/CHECKPOINT PURGE** command operates exactly like the **/CHECKPOINT FREEZE** command, except that the IMS DBCTL system waits for BMPs to complete their processing.

Forcing an acceleration of the shutdown process

All three variations of full shutdown (**/CHECKPOINT FREEZE | DUMPQ | PURGE**) allow transactions that had already been started to complete, and allow BMP programs to run to their next checkpoint. These commands might also allow CCTL threads that had already been started to complete. In some cases, you might have to wait a long time before IMS finally shuts down. For example, a conversational transaction or CCTL might be waiting for an end user to enter a response; or a long BMP program might not take any checkpoints.

If necessary, you can force shutdown (although some work might be lost). You can use various forms of the **/DISPLAY** or **QUERY** command to discover what work is still running. Then, you can use other commands to force shutdown (for example, the **/EXIT**, **/IDLE LINE**, or **/STOP REGION** commands).

If all else fails, you can bring down IMS by issuing a z/OS **MODIFY IMS,DUMP** command or **MODIFY IMS,STOP** command.

Attention: Only use the z/OS **MODIFY IMS** commands as a last resort, because you must immediately enter an emergency restart (**/ERESTART**) command and shut down again before you can run batch jobs.

IMS shutdown considerations

If automatic export to the IMSRSC repository during IMS shutdown fails, IMS is not allowed to shut down. A DFS4391E message is issued to indicate that IMS cannot shut down because automatic export failed. The error must be resolved and shutdown reissued so that automatic export can complete and IMS can shut down. Or, you can issue the **UPDATE IMS** command to set AUTOEXPORT(N) and retry the shutdown to allow IMS to shut down. If you need to cold start IMS when automatic export has not been done, see the procedures for retrieving the resources to be exported from the IMS log by using the Create RDDS from Log Records utility (DFSURCL0) and the EXPORTNEEDED parameter.

When an IMS shutdown command is issued in an IMS that has autoexport to repository enabled, the DFS4370I message is issued before the autoexport to the IMSRSC repository is started by IMS. The DFS4370I message indicates that IMS shutdown is checking for resource and descriptor definitions that have been created or updated and should be autoexported to the repository. The message indicates that IMS shutdown processing has not yet started.

Related reference

/CHECKPOINT command (Commands)

Shutting down an IMS system that uses dynamic resource definition

If dynamic resource definition (DRD) is enabled and you configured your system to automatically import resource and descriptor definitions from a resource definition data set (RDDS) or an IMSRSC repository

during a cold start, ensure that any online changes to the runtime resource and descriptor definitions are reflected in a system RDDS or repository.

Any changes made to online resources and descriptors that are not saved to a system RDDS or a repository are lost when you perform a cold start or when you use the **/ERESTART** command with the **COLDSYS** parameter to perform an emergency restart.

If AUTOEXPORT is enabled in the DFSDFxxx PROCLIB member, resources and descriptor definitions are exported automatically to the repository, the RDDS, or both, depending on the AUTOEXPORT specification and your DRD configuration, before the IMS shutdown checkpoint.

If AUTOEXPORT is not enabled, export resource and descriptor definitions to a repository or an RDDS by issuing the **EXPORT DEFN** command before shutting down IMS. You can check for and identify which resource and descriptor definitions have not yet been exported to the IMSRSC repository by issuing the **QUERY** *rsc_type* **SHOW(EXPORTNEEDED)** command, where *rsc_type* can be DB, DBDESC, PGM, PGMDESC, RTC, RTCDESC, TRAN, or TRANDESC.

If IMS is shut down with one or more resource definitions not exported to the IMSRSC repository, a DFS4419I message is issued during IMS shutdown to indicate that the IMS log and the IMSRSC repository are not in synchronization. If IMS must cold start after DFS4419I is issued, use the DFSURCL0 utility with the EXPORTNEEDED parameter after IMS is cold started.

Related concepts

Overview of the IMSRSC repository (System Definition)

Exporting MODBLKS resource and descriptor definitions to an RDDS (System Definition)

Exporting MODBLKS resource and descriptor definitions to an IMSRSC repository (System Definition)

Related reference

/ERESTART command (Commands)

/CHECKPOINT command (Commands)

Create RDDS from Log Records utility (DFSURCL0) (System Utilities)

Related information

DFS4419I (Messages and Codes)

Session termination

Terminating a session releases the terminal from its current logical connection to IMS (the VTAM application program). Terminating a session also makes the terminal available for sessions with other VTAM application programs, or you can terminate communications altogether.

Sessions can be terminated by the IMS MTO, the VTAM network operator, or the user at the terminal. There are two types of session termination:

Orderly termination
The terminal is allowed to complete normal processing before the session is terminated.

Immediate termination
Forces the terminal to terminate the session unconditionally.

Because these two session termination methods have different consequences, you must develop specific procedures for different situations.

Orderly termination

To initiate an orderly termination of the network, use the IMS **/CHECKPOINT [FREEZE | PURGE | DUMPQ] QUIESCE** command. When you specify the QUIESCE keyword, IMS sends a VTAM shutdown command to all terminals and waits until they all complete normal processing.

After all terminals indicate that shutdown is complete, IMS issues VTAM CLSDST macro instruction to cause VTAM to send the **UNBIND** command to all terminals. This command releases the terminals from their sessions with IMS, and prohibits additional data transmission.

During an orderly termination, the IMS MTO can terminate the network unconditionally, rather than wait for the orderly termination to complete, by initiating an immediate termination.

Immediate termination

To initiate an immediate termination of the network, use the IMS **/CHECKPOINT FREEZE | PURGE | DUMPQ** command. When you enter this command, without the QUIESCE keyword, IMS issues the VTAM CLSDST macro instruction to cause VTAM to send the **UNBIND** command to all terminals. This command releases the terminals from their sessions with IMS, and prohibits additional data transmission.

You can use the **/CLSDST** or **/STOP** commands to terminate stations or portions of a network selectively. These commands cause IMS to issue the VTAM CLSDST macro instruction for the specified stations. The **/STOP** command also prevents additional sessions from being established until you issue a **/START** command for the terminal.

The VTAM network operator can use the **VARY** command to terminate a session immediately. It is sometimes necessary for the VTAM network operator to use this command to terminate sessions that have an error preventing an I/O operation from completing.

The terminal user can use VTAM session control commands to terminate a session with IMS. For normal processing, when the user decides to terminate a session with IMS, the terminal should send the VTAM request-shutdown command. IMS completes any input or output currently in progress for that terminal and then issues the VTAM CLSDST macro instruction.

If the terminal detects an error condition from which it cannot recover, it can send the VTAM terminate-session command. VTAM releases the terminal from the session and notifies IMS.

Shutting down an IMS network

You can shut down the IMS network while shutting down IMS or without shutting down IMS. Use the **/CHECKPOINT** command to terminate the network and shut down IMS. Use the **/STOP DC** command to terminate the network only.

About this task

You can also use the VTAM **HALT NET** command to shut down the VTAM network.

If your network includes APPC, be aware of the following conditions:

- Allocated LU 6.2 conversations might prolong the shutdown process.
- Network shutdown does not prevent the application programs from establishing LU 6.2 conversations.
- LU 6.2-originated transactions, not scheduled but on the IMS queue, terminate with message DFS1970.

Terminating an ISC session from CICS

You can only use CICS control operator commands to terminate ISC sessions.

The CICS operator can release a session by using the **CEMT SET TERMINAL(***termid***) RELEASED | OUTSERVICE** command, where *termid* is the four-character session name (defined on the **DEFINE SESSIONS** command in the CSD utility), or the TRMIDNT in the DFHTCT TYPE=TERMINAL program. The following list describes the effects of the keywords for the **CEMT SET TERMINAL** command:

RELEASED
The session terminates when active transactions complete (unless you also specify the PURGE or FORCEPURGE keywords) and the session is left in a VTAM between-brackets state.

When you specify RELEASED, CICS initiates an orderly termination between IMS and CICS. Although, from the CICS point of view, the session may appear to be in warm-start state, it is actually in cold-start state as a result of the VTAM SBI/BIS (Stop Bracket Initiation / Bracket Initiation Stopped) command flow.

OUTSERVICE
The session terminates when active transactions complete (unless you also specify the PURGE or FORCEPURGE keywords), the session is left in a VTAM between-brackets state, and no additional transactions can use the terminal. For a VTAM terminal, using this keyword also causes it to be released and the operator to be signed off, either immediately or when the current transaction has completed.

When you specify OUTSERVICE, CICS requests resynchronization when the session is re-initiated, that is, the session is left in a warm-start state. To initiate a session in cold-start mode, the CICS operator must use the **CEMT SET TERMINAL (termid) COLDACQ** command.

You can specify both RELEASED and OUTSERVICE together on the same command.

Restriction: The OUTSERVICE keyword is not applicable for LU 6.2 devices.

Recommendation: Although a CICS application program can issue an EXEC CICS DISCONNECT call to initiate orderly session termination, this approach is not recommended in a normal application. However, you can write an "operator control" application to issue this call.

Any messages about the session's termination are sent to transient data destination CSMT.

Related tasks

"Connecting ISC sessions from CICS to IMS" on page 92
You can connect CICS® ISC sessions to IMS in two distinct ways. You can initiate the session explicitly or the CICS operator can initiate the session through a command.

Stopping the IRLM

You can stop the IRLM from the system console through a version of the **MODIFY** or **STOP** commands.

About this task

MODIFY*irlmproc***,ABEND,NODUMP**
STOP*irlmproc*

Recommendation: Terminate the IRLM subsystem only after all IMS subsystems connected to the IRLM have completed their processing.

Shutting down CQS

During normal shutdown in a shared-queues environment, IMS tells Common Queue Server (CQS) to shut down, but if it does not, you can use the z/OS **STOP** command to shut down the CQS subsystem. To shut down CQS, IMS can issue the CQSSHUT request, the CQSDISC request with the CQSSHUT=YES parameter command.

About this task

The IMS subsystem does not have access to the shared queues as long as CQS is down, and messages that are locked by that IMS subsystem are unavailable to all other IMS subsystems in the sysplex.

Normally, when a client disconnects from CQS by using the CQSDISC request and specifying the CQSSHUT=YES parameter, CQS shuts down after no clients are connected to it. In some cases, however, the CQS address space remains active, even when no clients are connected to it. This can happen under any of the following conditions:

- No client is connected to CQS when CQS is started.
- A client that was connected to CQS terminates abnormally, without issuing a CQSDISC request to disconnect from CQS, or issues a CQSDISC request with the CQSSHUT=NO parameter specified.

Procedure

You can shut down a CQS address space that has no clients connected to it by issuing the z/OS STOP command and specifying the job name of the CQS address space.

```
P cqsjobname
```

cqsjobname is the job name of the CQS address space that you want to stop. If no clients are connected to a CQS, that CQS shuts down. If clients are connected to the CQS, the STOP command is rejected, and message CQS0300I is issued.

Related concepts

CQS clients and handling special events (System Programming APIs)

Shutting down an IMSplex

If you are running IMS as an IMSplex with CSL, and you are shutting down the entire IMSplex, you should stop the CSL manager address spaces (ODBM, OM, RM, and SCI) after you shut down the IMS control region.

About this task

If you are planning to restart the IMS control region, you do not need to stop the CSL address spaces. The new IMS control region will connect to the existing CSL address spaces when it initializes.

The following briefly summarizes shutting down the IMSplex:

Procedure

1. Shut down the IMSplex components that participate in that IMSplex by issuing a **/CHE FREEZE** or similar command to the individual IMS components.
2. Shut down CSL by using one of the following methods:
   - A z/OS **MODIFY** command
   - A z/OS **STOP** command for each component
   - A CSLZSHUT request to an SCI in the IMSplex
3. Shut down CQS by issuing a z/OS **MODIFY** command.

Results

If automatic export to the IMSRSC repository during IMS shutdown fails, IMS is not allowed to shut down. A DFS4391E message is issued to indicate that IMS cannot shut down because automatic export failed. The error must be resolved and shutdown reissued so that automatic export can complete and IMS can shut down. Or, you can issue the **UPDATE IMS** command to set AUTOEXPORT(N) and retry the shutdown to allow IMS to shut down. If you need to cold start IMS when automatic export has not been done, see the procedures for retrieving the resources to be exported from the IMS log by using the Create RDDS from Log Records utility (DFSURCL0) and the EXPORTNEEDED parameter.

Shutting down the CSL

A Common Service Layer (CSL) is comprised of multiple address spaces. As such, you can shut down either an entire CSL or individual CSL manager address spaces.

About this task

You can shut down:

- A single CSL component, such as an instance of ODBM, OM, RM, or SCI
- A CSL, including all of its components, on a single z/OS image

- A CSL, including all of its components, that spans an IMSplex across multiple z/OS images

The entire CSL should be shut down only if no IMSplex components are connected to any CSL managers. You should first issue an IMS **/CHE FREEZE** or similar command to terminate all the IMSplex components that might be connected to CSL prior to shutting down CSL. If a CQS address space has more than one client, it might not shut down, and you might need to shut it down after all the clients have terminated.

If you shut down an instance of ODBM, OM, or RM and there are other instances of ODBM, OM, or RM active in the IMSplex, IMSplex members can still participate in IMSplex activities; however, an IMSplex member can communicate only with the SCI with which it is registered. If that SCI is shut down, any IMSplex members on the same z/OS image as that SCI cannot communicate with other IMSplex members until that SCI is restarted.

Attention: If you are running Automatic RECON Loss Notification (ARLN), you must keep SCI active (even after IMS is shut down) until all jobs that need access to the RECON data sets have completed.

If you are shutting down only some of the CSL managers to, for example, apply maintenance, you can shut down the individual CSL managers using the **STOP** command. You do not have to shut down the other IMSplex components if you are only shutting down a CSL manager to apply maintenance.

To shut down the entire CSL on one z/OS image, a program can issue a CSLZSHUT request to SCI that specifies the entire CSL to shut down. Or, you can issue a **MODIFY** command for one CSL. The **MODIFY** command shuts down the CSL on the z/OS image that is associated with the SCI that receives the command. The CSL managers quiesce work and then terminate. This command is used to terminate the CSL on a single z/OS image in an orderly manner.

If you shut down the whole IMSplex, a CSLZSHUT request can be issued to SCI that specifies that the entire CSL is to be shut down. Or you can issue the **MODIFY** command for the entire CSL. The **MODIFY** command shuts down the CSLs on all z/OS images in a single IMSplex that is associated with the SCI.

Recommendation: Use the **CSL SHUTDOWN** command with the z/OS **MODIFY** command interface to shut down the CSL, rather than stopping individual components.

You can shut down the CSL by either:

Procedure

- Issuing the z/OS **STOP** command to individual CSL manager address spaces.
- Issuing the **CSL SHUTDOWN** command using the z/OS MODIFY command interface.
- Using the CSLZSHUT request in a CSL application programming interface (API).

Related reference

CSLZSHUT: shutdown request (System Programming APIs)

Shutting down the CSL using z/OS commands

You can shut down the CSL as one unit by issuing the **CSL SHUTDOWN** command to any SCI in the IMSplex with the z/OS MODIFY command interface. You can stop individual modular units in the IMSplex by issuing the z/OS **STOP** command to the address space you want to stop.

About this task

To shut down a CSL on one z/OS image, issue the z/OS **MODIFY** command as follows:

```
F scijobname,SHUTDOWN CSLLCL
```

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

To shut down an entire IMSplex, issue the **z/OS** MODIFY command as follows:

```
F scijobname,SHUTDOWN CSLPLEX
```

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

Note: To shut down the CSL managers using the **SHUTDOWN CSLPLEX** command, a local SCI is required. If you issue the **SHUTDOWN CSLPLEX** command on a system without an active SCI, the CSL managers will not shut down.

In each of these examples, `scijobname` is the name of the SCI in the CSL. After it receives the command, SCI notifies other CSL managers (ODBMs, OMs, and RMs) to stop, and then SCI stops. If clients are currently connected to any CSL manager and were not first stopped with a **/CHE FREEZE** or other command, message CSL0300I is issued, work is quiesced, and then the CSL manager stops.

Shutting down the CSL ODBM

You can shut down ODBM by using one of several methods.

About this task

The options to shut down ODBM include:

- The system operator can use a z/OS **STOP** command:

  ```
  P odbmjobname
  ```

  `odbmjobname` is the job name of the ODBM address space to stop.
- Issue the **CSL SHUTDOWN** command, which can shut down either a CSL on one z/OS image or an entire IMSplex.
- Issue the CSLZSHUT request. CSLZSHUT is a programming interface that an assembler program can issue to shut down one or more CSL address spaces.

If no clients are connected to ODBM, ODBM shuts down. If clients are connected to ODBM, message CSL0300I is issued, and ODBM quiesces in-flight work. After all work is quiesced, the ODBM address space terminates.

Before shutting down an ODBM, consider the reasons for shutting down and how shutting down ODBM can impact other IMSplex members.

Related reference

CSLZSHUT: shutdown request (System Programming APIs)

Shutting down the CSL OM

Before shutting down an OM, consider the reasons for shutting down and how shutting down OM can impact other IMSplex members.

About this task

Recommendation: Although you can shut down OM by itself, you should shut down OM by shutting down the CSL as one unit.

To shut down the OM, do one of the following:

- The system operator can use a z/OS **STOP** command:

  ```
  P omjobname
  ```

  `omjobname` is the job name of the OM address space to stop.
- Issue the **CSL SHUTDOWN** command.

- Issue the CSLZSHUT request.

If no clients are connected to OM, the OM shuts down. If clients are connected to OM, message CSL0300I is issued, and OM quiesces in-flight work. After all work is quiesced, the OM address space terminates.

Related reference

CSLZSHUT: shutdown request (System Programming APIs)

Shutting down the CSL RM

Before shutting down a Resource Manager (RM), consider the reasons for shutting down and how shutting down RM can impact other IMSplex members. There are multiple ways to shut down RM.

About this task

Recommendation: Shut down RM by shutting down the CSL as one unit.

To shut down the RM, do one of the following:

Procedure

- The system operator can use a z/OS STOP command: `P rmjobname`
  `rmjobname` is the job name of the RM address space to stop.
- Issue the **`CSL SHUTDOWN`** command.
- Issue the CSLZSHUT request.

Results

If no clients are connected to RM and no IMSplex-wide process steps are in progress (or no IMSplex-wide processes are in progress for an RM with no resource structure defined), RM shuts down. If clients are connected to RM, message CSL0300I is issued, and RM quiesces in-flight work. RM waits for process steps to timeout and returns the process step responses to RM clients.

For an IMSplex defined with no resource structure, RM terminates any IMSplex-wide process that is in progress and issues message CSL2210I for each terminated process. To avoid this premature termination of the process, complete all IMSplex-wide processes before RM shutdown is attempted.

If the IMSRSC repository is enabled, RM termination disconnects from any repository RM is managing and deregisters from the Repository Server (RS) address space.

Related concepts

Overview of the IMSRSC repository (System Definition)

CSL RM, IMS, and Repository Server termination (System Administration)

Related reference

CSL SHUTDOWN command (Commands)

CSLZSHUT: shutdown request (System Programming APIs)

Related information

CSL0300I (Messages and Codes)

CSL2210I (Messages and Codes)

Shutting down the CSL SCI

Before shutting down the Structured Call Interface (SCI), consider the reasons for shutting down and how shutting down the SCI can impact other IMSplex members. There are multiple ways of shutting down SCI.

About this task

Recommendation: Shut down SCI by shutting down the CSL as one unit.

To shut down the SCI, do one of the following:

- The system operator can use a z/OS **STOP** command:

```
P scijobname
```

  `scijobname` is the job name of the SCI address space to stop.
- Issue the **CSL SHUTDOWN** command.
- Issue the CSLZSHUT request.

If no clients are connected to SCI, the SCI shuts down. However, if there are registered members on the local z/OS image, message CSL0300I is issued, and SCI does not process any new requests or messages from local members. After all in-flight requests have completed or timed out, the SCI address space terminates.

Related reference

CSLZSHUT: shutdown request (System Programming APIs)

Forced termination of IMS

You normally use the **/CHECKPOINT** command to shut down IMS. However, in certain error situations such as a control region loop, you might need to force termination of IMS. In this case, use the z/OS **MODIFY** command, and be sure to request a dump of the IMS control region.

You can also use the z/OS **CANCEL** command, specifying the DUMP keyword, to terminate IMS. If you need to enter this command more than once, give each **CANCEL** time to complete before entering it again.

Attention: Using the z/OS **CANCEL** command more than once might stop IMS from completing termination tasks in the proper order and from releasing z/OS system resources. If IMS does not complete termination, you might need to IPL your system before restarting IMS.

Related concepts

“Shutting down the IMS control region” on page 118
Use the **/CHECKPOINT FREEZE|DUMPQ|PURGE** command to shut down the IMS control region. You should provide guidelines for the MTO on selecting the right kind of shutdown, either specifying the FREEZE or PURGE keyword.

“Control region failures” on page 133
Software errors that cause an abend or a loop, or failures that are related to system data sets (including the log data sets) can cause control region failures. After you determine and correct the cause of the failure, you can restart the system by using the **/ERESTART** command.

Related reference

z/OS: MODIFY command

Related information

z/OS: CANCEL command

Offline dump formatter

Offline dump formatting can reduce the duration of IMS outages by increasing the time required for an online IMS subsystem to terminate. You should be able to restart online subsystems faster as long as z/OS remains operational.

Any DL/I batch job can also use offline dump formatting after calling for a SYSMDUMP.

IMS can produce a machine-readable dump for later offline formatting. This can be an SDUMP, SYSMDUMP, standalone dump, a dump produced using the DUMP command, or any other machine-readable dump, such as a z/OS SVC dump.

Producing a dump using the z/OS MODIFY command

Use the z/OS **MODIFY** command to create a machine-readable dump rather than a z/OS or JES **CANCEL** command.

About this task

The **CANCEL** command often produces undesirable online IMS dump formatting for canceled address spaces in addition to the requested dump for offline formatting.

Producing a dump using the z/OS DUMP command

The z/OS operator can request a console dump at any time, whether or not IMS is terminating, by entering the **DUMP COMM=description** command.

About this task

z/OS responds with the following message:

```
xx IEE094D  SPECIFY OPERAND(S) FOR DUMP COMMAND
```

To which the z/OS operator must then reply with the following command:

```
xx,JOBNAME=(imsname,dbrcname,dliname,irlmname),
   SDATA=(PSA,NUC,SQA,RGN,CSA,TRT)
```

Unless you dump all specified areas, the IMS Offline Dump Formatter might not be able to format all desired IMS subsets of the dump.

Producing a dump using the standalone dump (SADMP)

The z/OS operator can take a standalone dump of the entire system if z/OS fails. The operator receives a message when the SADMP completes.

About this task

This includes:

- Suspending operation of the system.
- Setting the address of the device containing your high-speed SADMP program, which you have generated to dump real and virtual areas.

  To dump virtual areas, code PROMPT on the AMDSADMP macro.
- Performing a STORE STATUS.
- Initiating the IPL of the standalone dump program. This program prompts you for:

  ```
  AMD001A TAPE=____
  AMD011A TITLE=______..._____
  ```
- Answering the prompts for dumping virtual storage areas:

  ```
  AMD059D ENTER DUMP OPTIONS, 'LIST', OR 'END'.
  > dump CSA,ASID('IMSjobname','DL/Iname','DBRCname','IRLMname')
  AMD059D ENTER DUMP OPTIONS, 'LIST', OR 'END'.
  > end
  AMD010I PROCESSING ASID=  ASCB= JOBNAME=*MASTER*
  ```
- Waiting until the standalone dump program completes. The following messages signal completion:

  ```
  AMD005I  REAL DUMP DONE
  AMD025I  VIRTUAL DUMPING COMPLETE FOR CSA
  AMD010I  PROCESSING ASID=0001
  ```

```
⋮
AMD023I  VIRTUAL DUMP COMPLETE - 00
```

Recommendation: Do not press the EXTERNAL INTERRUPT key before the dump completes because this could prevent the IMS Offline Dump Formatter from producing a formatted dump later.

Keeping dump data sets available

z/OS console and IMS master terminal operators must keep the system dump data sets available to avoid losing dumps of the IMS subsystem. z/OS operators should transfer and process IMS-produced SDUMPs using the IEBGENER program according to your installation procedures.

About this task

If you produce IMS dumps using SYSMDUMP DD statements, the MTO should transfer and process the dump data sets according to your installation procedures.

The z/OS operators should be aware of the following IMS messages that relate to the offline dump process:

```
DFS3906I  DFSDUMP FAILED BECAUSE ALL SYSTEM DUMP DATA SETS ARE FULL
DFS3906A  REPLY "S" TO SKIP, OR "U" TO RETRY AFTER CLEARING
          A DUMP DATASET.
```

In this case, if a dump is required, reply U after clearing one or more dump data sets. If a dump is not required, reply S.

```
DFS3907I  DFSSUMP FAILED BECAUSE A DUMP IS IN PROGRESS
DFS3907A  REPLY "S" TO SKIP, OR "U" TO RETRY AFTER THE CURRENT
          DUMP COMPLETES.
```

In this case, if the dump is required, reply U after you receive the following message:

```
IEA911E  COMPLETE/PARTIAL DUMP ON SYS1.DUMPxx.
```

If a dump is not required, reply S.

```
IEA793A  NO DUMP data sets AVAILABLE FOR DUMP=nnn BY JOB (imsproc).
USE THE DUMPPDS COMMAND OR REPLAY D TO DELELTE THE DUMP
```

In this case, the dump is written to the dump data set after clearing one or more dump data sets. If a dump is not required, reply D.

Database resource adapter storage

You can use the database resource adapter (DRA) to access an IMS DB subsystem. In this instance, the DRA code actually runs in the CCTL region or z/OS application region, not in any of the IMS DB regions.

A shutdown of the IMS DB subsystem with a dump does not produce any information about the DRA.

Chapter 5. IMS failure recovery

When a failure occurs, you must perform the correct recovery and restart procedures to ensure system integrity. You can use the IMS control region Extended Specified Task Abnormal Exit (ESTAE) routines to clean up resources after an IMS abend.

The reason for the abnormal termination and the results of cleanup processing have a great effect on the subsequent restart process. The processing sequence of these routines is as follows:

- The storage management task ESTAE routine writes recovery data to the restart data set (RDS) and then closes the RDS.
- The IMS control task ESTAE routine terminates all dependent regions and issues message DFS629I.
- All ESTAE routines attempt to purge the OLDS log buffer and to write a termination log record.
- The ESTAE routines tell DBRC to sign off abnormally.
- The ESTAE routines tell IRLM to quit, and either retain or release locks, based on whether DBRC indicates that databases have been updated.
- The ESTAE routines disconnect XRF alternate subsystems from the z/OS availability manager to perform I/O prevention.

In addition, the IMS resource cleanup module cleans up resources whenever IMS terminates. It releases the common storage area (CSA), closes DBDSs, deletes the subsystem interface control block, closes the VTAM Access-method Control Block (ACB), and issues message DFS627I.

If you use the DL/I address space (you specified LSO=S in the IMS procedure), IMS issues message DFS603I to indicate that cleanup is successful.

z/OS system failures

If IMS terminates because of a z/OS failure or a hardware or power failure, you must IPL z/OS and then restart IMS using an **/ERESTART** command.

Database I/O errors, occurring as a result of, or at the same time as, a hardware failure, can leave damaged data on DASD. If IMS terminates abnormally after this kind of an I/O error before IMS can record the I/O error on the IMS log and in DBRC, IMS does not know that there is damaged data. And if IMS does not know that there is damaged data, your data integrity is compromised.

If you experience or suspect I/O errors that lead to these kinds of failures so that it is necessary to use the **/ERE OVERRIDE** command, check and, if necessary, recover all databases allocated for dynamic backout or DEDB Redo during emergency restart.

If APPC/MVS fails, restart APPC/MVS and issue the IMS **/START APPC** command to establish the interface between IMS and APPC/MVS.

Control region failures

Software errors that cause an abend or a loop, or failures that are related to system data sets (including the log data sets) can cause control region failures. After you determine and correct the cause of the failure, you can restart the system by using the **/ERESTART** command.

If the failure results in a loop in the control region, you might need to force termination of the region with the z/OS **MODIFY** command.

When the IMS control region fails, APPC/MVS terminates LU 6.2 conversations with the DEALLOCATE abend.

When the IMS control region fails, the CQS subsystem to which it is connected does not shut down. If you need to shut down the CQS subsystem, use the z/OS **STOP** or **CANCEL** command.

Related concepts

"Forced termination of IMS" on page 130

You normally use the **/CHECKPOINT** command to shut down IMS. However, in certain error situations such as a control region loop, you might need to force termination of IMS. In this case, use the z/OS **MODIFY** command, and be sure to request a dump of the IMS control region.

Emergency restart failures

Emergency restart backs out incomplete changes from programs that were active at the time of failure. However, damaged logs or message queues might prevent emergency restart from completing. If this occurs, you might be able to perform an emergency restart from a previous checkpoint or by specifying FORMAT ALL on the **/ERESTART** command.

If you are running an IMS DB/DC system, and an emergency restart fails, you do not have to cold start the entire system. You can use the commands **/ERE COLDCOMM** and **/ERE COLDBASE**. These commands cold start part of the system while restarting the other part: **/ERE COLDCOMM** cold starts the DC part and restarts the rest, while **/ERE COLDBASE** cold starts the DB part and restarts the rest.

If you are running an IMS DBCTL or DCCTL system, and an emergency restart fails, you must cold start the entire system using the **/ERE COLDSYS** command.

Re-establishing database integrity

If no emergency restart succeeds, an appropriate response depends on your understanding the reason for the failure. Although it might not work exactly as written for your installation, use the following procedure to reestablish database integrity.

Procedure

1. Run the Log Recovery utility (DFSULTR0) to close the OLDS (CLS mode). If the utility needs additional logs, concatenate them in your JCL, oldest first, and rerun the utility.
2. Be sure that your log is correct and that your databases are consistent with the log.
3. Produce an SLDS using the Log Archive utility (DFSUARC0).
4. Forward recover all updated DEDBs to make them consistent with the log using the Database Recovery utility (DFSURDB0).

   Recommendation: If the recovered area utilized the shared VSO option, ensure that there are no failed-persistent XES connections to the CF structure used by the area. If necessary, use the z/OS **SETXCF** command to delete the connections prior to cold start of the systems involved.
5. For full-function databases, run the Batch Backout utility (DFSBBO00) to back out programs (PSBs) that were active at the time of failure. Determine these from the output of the Log Recovery utility.
6. Update the RECON data set, if necessary, to reflect the current status of the system.
7. If you use MSDBs, rebuild the MSDBINIT data set from the last used MSDBCP1 and MSDBCP2 data sets and the last SLDS.

What to do next

At this point, database integrity should be reestablished. If you are using DBRC or data sharing, additional actions might be required before you perform a cold start.

System data set failures

If an IMS system data set fails because of an unrecoverable I/O error, you must scratch, reallocate, and reformat the data set before you restart the system using the **/ERESTART** command.

Message queue data set failures

Two types of problems can occur with message queue data sets: they can run out of space, or an I/O error can occur. In either case, you need to scratch and reallocate the problem data set, and increase its size if necessary.

Note: This topic does not apply to a DBCTL environment or to a shared-queues environment.

IMS automatically reconstructs the contents of the message queues when you restart IMS and specify the BUILDQ and FORMAT keywords on the restart command.

Restriction: IMS can only reconstruct the message queue data sets from the initial cold start or from a SNAPQ or DUMPQ checkpoint.

To recover message queues during IMS restart, use the **/NRE BUILDQ** or **/ERE BUILDQ** command. If you also want to reinitialize the message queue data sets (that is, reformat them with null records), use the FORMAT keyword on the restart command.

To use the **/NRE BUILDQ** command, you must have previously shut down IMS using a **/CHECKPOINT DUMPQ | PURGE** command. You can use the**/ERE BUILDQ** command to recover messages from a prior SNAPQ checkpoint (**/CHECKPOINT SNAPQ**).

If an emergency restart fails, you must initiate an emergency cold start using the **/ERE COLDSYS** command. An emergency cold start does not perform any database recovery, so you must close and archive the last OLDS.

If an IMS restart fails, you can use the IBM IMS Queue Control Facility for z/OS (QCF) to recover the message queues if you use one of the following cold start commands after the failed restart: **/ERE CHECKPOINT 0**, **/ERE COLDCOMM**, or **/ERE COLDSYS**. With QCF, you can select messages from the OLDS or an SLDS and requeue them to the message queues after IMS restarts (cold start). QCF provides recovery modes that analyze and select the messages to be requeued.

Related reference

IMS Queue Control Facility overview

Other system data set failures

You should make periodic backup copies of IMS system data sets so that if an error occurs, you can recover your data or recreate it from a backed up copy.

If an error occurs on an IMS data set, such as IMS.ACBLIBx or IMS.FORMATx, you need to recover it. If you have made periodic backup copies of the data set, you can use the latest copy. If you have not changed the content of the data set since making the copy, you can use it as is. If you have changed the content of the data set since making the copy, for example, you have added ACB generations since copying the ACBLIB data set, you need to redo the changes before the backup copy can be considered up-to-date.

If you have not made backup copies of the data set, you need to recreate it. For example, if you lose the active format library (IMS.FORMATx) and you have no backup copy of it, you need to rerun Message Format Services and redo all your format definitions.

If you have inactive data sets from an online change, you can recover those data sets involved with online change (MODBLKS, ACBLIB, and FORMAT). In this case, you must reapply any changes made to the system data sets.

IMS does not offer any specific utilities or commands to recover system data sets.

RECON data set recovery

If an I/O error occurs on a RECON data set, DBRC tries to use an available spare data set. DBRC then copies the healthy RECON data set to the spare data set, and then activates the spare data set.

When other subsystems using the failed RECON data set complete their processing, you can delete and redefine the error data set so it can be used as a spare. If, however, you want to analyze the RECON error, you should allocate new space for the RECON data set, rather than deleting and redefining it.

If DBRC cannot locate a spare data set, all currently executing jobs continue to process using the RECON data set in single mode. DBRC allows new jobs to start with only one RECON data set if you specify the STARTNEW keyword on the **INIT.RECON** or **CHANGE.RECON** command (or the DSPAPI FUNC=COMMAND COMMAND=INIT.RECON or DSPAPI FUNC=COMMAND COMMAND=CHANGE.RECON API requests).

Attention: Do not allow jobs to start with only one RECON data set because it jeopardizes the integrity of the system.

Restoring RECON data sets if both are unusable

It is unlikely that both RECON data sets would be unusable. However, if this situation occurs, you must back up, delete, and redefine your RECON data sets, make an image copy of all applicable DBDSs, and close any open, out-of-date OLDSs.

About this task

To restore and resynchronize the RECON data sets with the databases:

Procedure

1. Stop all jobs that require access to the RECON data set.
2. Optional: If you can access both RECON data sets, use the VSAM access method services **REPRO** command to back them up. This step is optional, but recommended.
3. Use the VSAM access method services utility to delete and redefine your RECON data sets.
4. Issue the access method services **REPRO** command to restore one of the RECON data sets.
5. Issue the **REPRO** command to restore the other RECON data set from the first.
6. Issue the DBRC **LIST.RECON** command or the DSPAPI FUNC=QUERY API request to list one of the RECON data sets. Based on the list, determine which DBDSs IMS updated since you made the backup in Step 2. If you cannot determine which DBDSs have been updated, assume that all have been updated.
7. Issue the DBRC **CHANGE.IC** command (or the DSPAPI FUNC=COMMAND COMMAND=CHANGE.IC API request) with the INVALID keyword to mark all image copy records in error for all applicable DBDSs in Step 6.
8. Make an image copy of all applicable DBDSs from Step 6.
9. Issue the DBRC **BACKUP.RECON** command (or DSPAPI FUNC=COMMAND COMMAND=BACKUP.RECON API request) to make a backup copy of the RECON data sets.
10. Close any open, out-of-date OLDSs using the **NOTIFY.PRILOG** command (or DSPAPI FUNC=COMMAND COMMAND=NOTIFY.PRILOG API request) to clean up the new RECON data set.

Results

The RECON data sets are now restored and resynchronized with the databases.

Note: If DBRC does not manage a large number of databases, it might be easier to use the following procedure:

1. Stop all jobs that require access to the RECON data set.
2. Define new RECON data sets.
3. Initialize these RECON data sets.

4. Register the environment (always keep a backup copy of the most recently initialized, but not yet used, RECON data set available).
5. Take image copies of all databases.

Log errors

You can experience three types of errors during logging. A write error on the OLDS, a read error on the OLDS or SLDS, and a read or write error on the WADS can occur.

OLDS write error
When an OLDS has a write error, IMS stops the OLDS (or a pair of OLDSs) in error. No specific MTO interaction is required. When an OLDS is no longer needed for dynamic backout, IMS dynamically deallocates a stopped OLDS.

If you are using single logging, IMS switches to the next OLDS. If the error causes only two OLDSs to remain available, IMS terminates. If an error occurs and no other OLDS is available, IMS abends with code U0616.

If you are using dual logging and an error occurs (either to one or both volumes in the pair), IMS switches to the next OLDS pair and continues. What happens next depends on what is specified on the DEGRADE control statement of the LOGGER section of the DFSDF*xxx* member.

DEGRADE=YES
Specifies that IMS will continue to use all good pairs of OLDSs until only two pairs remain. At this point, IMS switches (degrades) to single-logging mode, using whatever good data sets are left from each pair of OLDSs. When only two good OLDSs remain, IMS terminates (as it does in normal single-logging mode). DEGRADE=YES is the default.

DEGRADE=NO
Specifies that IMS will terminate when only two good pair of OLDSs remain.

Recommendation: Use the **/START OLDS** command to allocate new pairs of OLDSs.

OLDS or SLDS read error
When read errors occur on the OLDS or the SLDS during emergency restart, IMS abends. When read errors occur on the OLDS during dynamic backout, backout fails.

WADS error
When a WADS has an I/O error, IMS switches to another WADS if one is available. If none is available, processing continues without a WADS. IMS maintains log-write-ahead protocols by truncating log buffers. As with OLDS errors, no specific MTO interaction is required. You should scratch and reallocate any WADS with a write error after you shut down IMS.

Log error recovery

Problems can occur with any of the log data sets (OLDS, SLDS, WADS, restart data set (RDS), and RLDS). Your actions depend on the log, the type of error, whether single or dual logging was in effect when the error occurred, and what processing was being performed when the error occurred.

Log Recovery utility (DFSULTR0)

The Log Recovery utility produces a usable log data set from an OLDS or SLDS that contains read errors or was not properly closed.

The Log Recovery utility has four modes of operation:

- CLS mode, to close an OLDS
- DUP mode, to create:

  Create an interim log containing error records
  Create a closed batch SLDS containing an end-of-file mark
- REP mode, to:

  - Read the interim log
  - Replace the error records with data that you specify
  - Create a new log
- PSB mode, to produce a report of active PSBs from the input log

When IMS opens the input logs, DBRC validates them, ensuring that the data set name and volume serial numbers agree with those in the RECON data set. For log recovery, DBRC requires that the entire log data set be specified as input to the IMS Log Recovery utility, not just selected volumes. For a closed SLDS, DBRC requires only the last volume as input to the Log Recovery utility.

When IMS opens the output logs, it calls DBRC to do the following:

- Create an IPRIOLDS record (and ISECOLDS record if you are using dual logging) using the time stamp of the input log and the data set name from the job file control block of the output log. Applies to DUP mode.
- Create new OLDSs from interim OLDSs. Applies to REP mode.

DBRC issues the **GENJCL.CLOSE** command to generate a job to run the Log Recovery utility. When you issue this command, specify the subsystem ID of the IMS subsystem that created the OLDS to be closed. You can also specify the OLDS to be closed. If you do not specify an OLDS, DBRC closes the most recent, open OLDS. You can issue the **GENJCL.CLOSE** command using the Recovery Control utility or as an IMS command.

When IMS reads the write-ahead data sets (WADS), the Log Recovery utility might issue the DFS3253W warning message to explain the MVS messages that VSAM issues in the job log.

Related reference

Log Recovery utility (DFSULTR0) (System Utilities)

Related information

DFS3253W (Messages and Codes)

OLDS recovery

IMS automatically closes the OLDS during normal shutdown or during emergency restart. You must close an OLDS before you can archive it or use it as input to any utility.

You must close the OLDS using the Log Recovery utility in either of the following circumstances:

- When an emergency restart fails and, rather than performing another emergency restart, you perform a cold start
- When IMS does not close the OLDS because IMS detected a write error (single logging only)

The Log Recovery utility recovers the OLDS from the following types of errors:

- An I/O error while reading the input log data set
- An error in the log record
- A sequence error in the log record, the log block, or the OLDS write time stamp

To recover the OLDS, the Log Recovery utility does the following:

- In CLS mode, the utility closes an input OLDS from information in the WADS if IMS cannot close it because of a system failure. Otherwise, the utility closes the OLDS from the OLDS used immediately after the OLDS in error. The utility uses the immediately prior OLDS, if any, to establish a base point for close processing, using the last block sequence number.

  If dual logging is in effect, IMS must close both OLDSs. If IMS successfully closes the logs, you do not need to use the Log Recovery utility.
- In DUP mode, the utility reads an OLDS and duplicates all readable records on a log.
- In REP mode, the utility reads the interim log created in DUP mode, copies good log blocks, and replaces error blocks with good ones based on information that you specify in control statements. The output log data set is a usable OLDS.

If the OLDS being recovered has not been closed (DBRC shows a stop time of zero), you must use the output from REP mode as input to CLS mode.

SLDS recovery

The Log Recovery utility closes a system log data set (SLDS) created by an IMS batch job. You must close an SLDS before you can use it as input to any utilities or to an IMS restart.

The Log Recovery utility recovers the SLDS from the following types of errors:

- An I/O error while reading the input log data set
- An error in the log record
- A log record sequence error

To recover an SLDS, the Log Recovery utility reads the SLDS based on the mode that it is in:

- In DUP mode, the utility reads an SLDS and duplicates all readable log records onto an interim log. When the utility encounters the first error or end of file (EOF), it stops copying and closes the SLDS. When you specify a nonzero error count, DUP mode writes error blocks and error ID records on the interim log until either the error count reaches the number you specify or EOF is reached. The utility uses the interim log containing these error blocks and error ID records as input to REP mode.

  You can close an SLDS by using DUP mode and specifying the error count as zero. REP mode is not required.
- In REP mode, the utility reads the interim log created by DUP mode, copies good log blocks, and replaces error blocks with good ones based on information that you specify in control statements. The output log data set is a usable SLDS.

WADS or RDS log recovery

You cannot recover a write-ahead data set (WADS) or a restart data set (RDS). If a problem occurs with either type of data set, you must scratch, reallocate, and format it during an IMS restart.

Dependent region failures

Two types of dependent region failures can occur: an application program failure or a region controller failure. In each case, if an LU 6.2 conversation established by the application program is running in the failing dependent region, APPC/MVS terminates the conversation with the DEALLOCATE abend.

- Application program failures, in which the application program abends but the region is still active.
- Region controller failures, in which the region fails. This kind of failure can occur, for example, when the program loops.

Application program failures

When an IMS application program abends, IMS issues message DFS554I (and in some cases message DFS555I) and then backs out all database changes made by the failing program to the last checkpoint, or to last point at which the program was scheduled. You must use the **/START PROGRAM** and **/START TRANSACTION** commands to restart the program and transaction after correcting the cause of the failure.

If an error occurs during database backout and restartable backout does not correct the problem, follow the procedures for database recovery before reactivating the program and transaction.

Region controller failures

One symptom of a dependent region failure is an increase in the number of transactions on the message queue. If the region appears to be looping, stop the region with the **/STOP REGION ABDUMP** command. If the region does not stop, enter the **/STOP REGION CANCEL** command.

You can monitor the status of your regions with the **/DISPLAY ACTIVE**, **/DISPLAY Q TRANSACTION**, and **/DISPLAY Q BALGRP** commands.

z/OS issues messages IEF450I and IEF404I to indicate an abend in a BMP, MPP, or IFP region. If these messages are not followed by a DFS554I message, issue the **/STOP REGION ABDUMP** command so that IMS recognizes that the region has terminated and backs out any database updates.

After you determine and correct the problem that caused the region to loop or that caused the abend, use the **/START** command to start the transaction and application program again.

Restart of BMP regions

If the application program uses the IMS restart call (XRST), you can restart the program from its latest checkpoint. You can identify this checkpoint from the DFS681I, DFS395A, or DFS682I messages.

DFS681I
This message lists the checkpoint ID and the PSB name, and is issued when the program takes a checkpoint.

DFS395A
The Database Batch Backout utility issues this message when it cannot restart.

DFS682I
This message lists the checkpoint ID and the program name, and is issued after emergency restart or after dynamic backout following an abend.

Specify the checkpoint ID value or LAST in the PARM field of the JCL used for restarting the BMP.

To restart a BMP after a cold start of IMS, use the IMS Log Analysis utility to scan for X'37' log records that contain BMP restart information for the desired BMP (job name, PSB name, program name). The last X'37' log record printed contains the information needed to restart the BMP region.

Restart of Fast Path regions

If a Fast Path message-driven application program or DEDB online utility program abends, you can restart the Fast Path region immediately. You can restart the region with the **/START REGION** command.

You should provide the MTO with documentation of the names of the procedures that are started using the **/START REGION** command because the procedure name must be given with this command. You should keep this documentation current because these regions require system resources, such as virtual storage and database buffers.

Database failures

IMS issues message DFS0451I or DFS0451A when there is an error in an IMS database. Every time IMS encounters a DL/I I/O error, IMS creates an extended error queue element (EEQE) that identifies the block or VSAM control interval in error.

When IMS closes a database, it automatically retries read and write errors on DL/I databases. If successful, forward recovery of the database is not required. Otherwise, forward recovery is eventually required. It might be possible to defer recovery to a more convenient time. Deferring recovery does not inhibit scheduling access or updating.

Using DEDB multiple area data sets also allows application programs to continue when I/O errors exist. For DEDB I/O errors, IMS issues messages DFS2571, DFS2572, DFS3712, and DFS3713. If a DEDB area is not available, the application receives an FH status code.

IMS maintains I/O information and buffer images across restarts. IMS does this by recording the EEQEs in DBRC, notifying all systems that are using the IRLM, and, during initialization and checkpoint, logging EEQEs and virtual buffers to the OLDS.

Related concepts
Database failures (Database Administration)

Database recovery

The primary methods for recovering a database after a failure are *forward recovery* and *backward recovery* or backout. Two methods of database recovery are possible for each type of recovery.

Forward recovery involves reconstructing the database from a copy of the database made prior to the database failure. Forward recovery uses the information you have been keeping, such as image copies, logs, and so forth, and reapplies it to the database copy. It is based on the notion that if you knew what the data was like at one time and you know what you have done to it since then, you can process the data to return the database to the state it was in just before it was lost.

IMS supplies two methods for the forward recovery of databases:

- The Database Recovery utility (DFSURDB0).
- If you have the Database Recovery Facility or Online Recovery Service products installed, you can use the /RECOVER command.

Backout allows you to remove incorrect or unwanted changes from existing information or work without rebuilding from a prior copy of the database.

There are two types of database backout:

- Dynamic backout, in which IMS automatically backs out changes to a database, usually after an application program error.
- Database batch backout, in which the Batch Backout utility (DFSBBO00) removes database changes made by IMS batch jobs and online programs.

Related concepts

Recovery of databases (Database Administration)

Database backout (Database Administration)

Related reference

Database Recovery utility (DFSURDB0) (Database Utilities)

Recovering from a network failure when the remote terminal stops responding

If IMS stops responding to a remote terminal, you might be able to recover the session if the IMS master terminal is still active. The MTO receives a message that indicates why the terminals are not operational if they have not been stopped by the **/CLSDST**, **/STOP**, or **/PSTOP** commands. Correct the error and issue a **/START** or **/RSTART** command to restore operation for the terminals.

About this task

You can use the various forms of the **/DISPLAY** command to list the status for communication lines, terminals, nodes, and ETO dynamic users.

- Use the **/DISPLAY LINE** command for communication lines
- Use the **/DISPLAY LINE PTERM** command for terminals
- Use the **/DISPLAY NODE** command for nodes
- Use the **/DISPLAY USER** command for ETO dynamic users

For a VTAM terminal session, you might only need to use the **/OPNDST NODE** command to restart it.

If you have a hung node, you can free it with a **/CLSDST FORCE** command, if the following conditions exist:

- A VTAM display indicates that no session exists.
- The output from a **/DISPLAY NODE** command indicates that a CID exists, the node is connected, and the node is not idle.

If a session does exist, terminate it with the VTAM command **VARY INACT,FORCE**, but use caution, because this command terminates all parallel sessions.

If a static node or ETO dynamic user is hung in Fast Path input response mode, as indicated by a "RESP-INP-FP" status in the output of the **/DISPLAY NODE** or **/DISPLAY USER** commands, you can reset the node or user with the appropriate **/STOP** and **/START** command sequence. For static nodes, issue **/STOP NODE** and **/START NODE**. For ETO dynamic users, issue**/STOP USER** and **/START USER**.

If a static node or ETO dynamic user is hung in full-function input response mode, as indicated by a "RESP-INP" status in the output of the **/DISPLAY NODE** or **/DISPLAY USER** commands, you can reset the node or user with the **/DEQUEUE** command to discard response-mode output so that the **/RSTART** command can reset terminal response mode.

If all terminals are operational and appear to function, then it is likely that an application program has terminated abnormally, or is in a loop, or has a resource conflict that prevents its scheduling. To find the cause of the problem, use one of the following commands: **/DISPLAY ACTIVE**, **/DISPLAY PROGRAM**, **/DISPLAY TRAN**, or **QUERY TRAN**.

If you use **/DISPLAY ACTIVE** and continually see a program listed as active in a MPP or BMP region, that program is probably in a loop. For a program that is continually active, check that its PROCLIM value (specified in the PSB) is correct. Use the **/STOP REGION ABDUMP** command to terminate the region, or use the **/STOP** command to stop programs and transactions. You should be careful when using the **/STOP REGION CANCEL** command because in certain instances it can cause an abend of the IMS control region.

CPI Communications failures

For a CPI Communications driven LU 6.2 conversation, IMS TM is not involved and places no restrictions on your choice of either committing or backing out updates.

Session failure

In the event of a session failure, a CPI Communications driven application program can provide full integrity by issuing commit (SRRCMIT) or a backout (SRRBACK) calls. If an LU 6.2 session fails during an LU 6.2 conversation, you can end the conversation or continue processing.

If the session fails before the transaction completes phase one of the two-phase commit sync-point protocol, IMS takes action as follows:

- For both standard and modified standard DL/I application programs, if the original conversation is asynchronous, IMS commits all database changes, sends all output messages asynchronously, and deallocates any established conversations.
- For standard DL/I application programs:
  - If the conversation is synchronous and a user destination exit routine is provided, the routine determines whether to abort or commit. If it commits, IMS sends all output messages asynchronously and commits all database changes. If it aborts, or if no user destination exit routine is provided, IMS aborts the transaction and backs out all changes.
- For modified standard DL/I application programs:
  - If the user destination exit routine commits, IMS commits database changes and inserts all messages to the IMS message queue asynchronously.
  - If the original conversation is synchronous and the user destination exit routine aborts, IMS aborts the transaction, backs out all database changes, and discards all messages inserted to the IMS message queue (unless the insert used an express PCB).

If the session fails after the transaction starts phase two of the two-phase commit, IMS sync-point processing continues for both standard and modified DL/I application programs, despite a session failure.

System failure

When the IMS system fails during a CPI Communications driven LU 6.2 conversation, IMS determines whether to resolve in-doubts for IMS-protected resources. Examples of IMS-protected resources are IMS DB databases, Db2 for z/OS databases, and IMS TM messages.

For standard and modified standard DL/I application programs, if the IMS system fails before the transaction completes phase one of the two-phase commit sync-point protocol, IMS backs out changed data during IMS restart. This backout includes all discardable transactions that were processing at the time of the system failure. IMS requeues non-discardable transactions so they can be processed.

If the transaction completes phase one of the two-phase commit, IMS resolves in-doubts during IMS restart. If only IMS resources are affected, IMS commits changes. If Db2 for z/OS resources are affected, IMS tells Db2 for z/OS to commit or abort the changed data, as appropriate.

Recovery processing for CPI Communications driven application programs

If your application programs require recovery assistance, they should use the implicit API provided by IMS rather than the CPI API. Implicit support allows application programs that do not normally use LU 6.2 protocols to use LU 6.2 devices.

The implicit API support is the original IMS DL/I API (using *xxx*TDLI calls, where *xxx* represents the programming language used).

IMS provides no recovery processing for application programs that use the explicit CPI application programming interface (API). IMS discards all messages for and from CPI Communications driven application programs at IMS restart, regardless of their state at the time of failure. Application program designers should use the SAA resource recovery resynchronization functions for explicit CPI application programs.

Related reference

DL/I calls for IMS and CICS (Application Programming)

MSC VTAM message resynchronization and recovery

For IMS systems linked by VTAM, IMS maintains the integrity of messages and control blocks during system and session failures. IMS recovers sessions automatically if both half-sessions save resynchronization information.

Because a logical link is broken after an abend caused by a system failure, user action is required for recovery. Performing a normal cold start destroys session resynchronization information, so you should use the /ERESTART command to restart IMS.

During emergency restart, IMS restores the message queues. After the link is reestablished, IMS resynchronizes messages between half-sessions.

In contrast, the message queues are lost during a cold start because IMS cannot restart the link at the point at which the system failed. Messages that were lost must be resubmitted.

After you reestablish a VTAM link after a session fails, you must reassign as primary the half-session originally assigned as primary (the one from which you issue the /RSTART command or the type-2 `UPDATE MSLINK(`*`linkname`*`) START(COMM)` command).

A z/OS system can run more than one IMS subsystem at a time. One IMS subsystem communicates with the other subsystem using real-storage-to-real-storage communication. z/OS provides the real-storage-to-real-storage connection primarily for system protection and testing purposes.

If a physical link between systems fails, the MTO should reestablish communication through an alternate physical link:

1. Use the /MSASSIGN LINK command or the type-2 `UPDATE MSLINK(`*`linkname`*`) SET(MSPLINK(`*`msplinkname`*`))` to reassign the logical link to an alternate physical link. This command must be entered on both systems.

2. Use the /RSTART LINK command or the type-2 UPDATE MSLINK NAME(*linkname*) START(COMM) command to start the logical link.

IMSRSC repository recovery

You can restore the IMSRSC repository with backup IMSRSC repository or by scratching the IMSRSC repository data sets and repopulating.

Backing up the IMSRSC repository

Consider backing up the IMSRSC repository data sets, so that the IMSRSC repository can be restored.

By backing up the IMSRSC repository data sets, IMSRSC repository can be restored in the following conditions:

- IMSRSC repository error
- Falling back from a new IMS release that changed the IMSRSC repository format
- Backing off maintenance that changed the IMSRSC repository format

The IMSRSC repository consists of a minimum of 4 data sets:

- Primary repository index data set (RID)
- Primary repository member data set (RMD)
- Secondary repository index data set (RID)
- Secondary repository member data set (RMD)

Use IDCAMS to back up either the primary RID and RMD pair, or the secondary RID and RMD pair. Consider using a naming convention for the IMSRSC repository backup data sets to help tie them together to indicate that they belong to one IMSRSC repository.

Scratching and repopulating the IMSRSC repository data sets

If an IMSRSC repository error occurs with no backup IMSRSC repository, you need to scratch and repopulate your IMSRSC repository data sets.

In different conditions, perform the following steps to scratch only the IMSRSC repository data sets, not the IMSRSC repository catalog data sets, and repopulate the IMSRSC data sets from IMSs or RDDSs.

Table 21. Scratching and repopulating the IMSRSC repository data sets

Condition	IMS systems are not cloned	IMS systems are cloned
IMS is active	Route the following **EXPORT** command to the IMS to populate the repository with the IMS's resource definitions: `EXPORT DEFN TARGET(REPO) NAME(*) TYPE(ALL)` A separate **EXPORT** command must be issued for each IMS.	Route the following **EXPORT** command to one of the IMSs: `EXPORT DEFN TARGET(REPO) NAME(*) TYPE(ALL) SET(IMSID(ims1,ims2,...))` The IMSID list must contain the IMSID of each IMS in the IMSplex, which uses the IMSRSC repository, including the IMSIDs of any IMSs that are not active.

Table 21. Scratching and repopulating the IMSRSC repository data sets (continued)		
Condition	**IMS systems are not cloned**	**IMS systems are cloned**
IMS is not active	Use the DFSURCL0 utility to create an RDDS from the IMS's log records and then use the CSLURP10 utility to populate the repository with the definitions from the RDDS. The CSLURP10 IMSID() input parameter specifies the IMSID of the IMS system, for which the resource definitions are being written to the repository.	Use the DFSURCL0 utility to create an RDDS from one of the IMS's log records. Then use the CSLURP10 utility to populate the repository for all of the IMSs in the IMSplex, which use the IMSRSC repository, by listing all of the IMSIDs on the CSLURP10 IMSID() input parameter. You can also see CSLURLFL procedure (System Definition) about how to load the IMSRSC repository from IMS log records in one job.

CQS failures and structure failures

The CQS subsystem automatically recovers from most types of failures. Because it is registered with the z/OS Automatic Restart Manager, if the CQS subsystem fails, z/OS restarts it. If one of the structures in the coupling facility fails, CQS initiates a structure rebuild (using data from the CQS log) to recover the structure.

If you lose connection with the last remaining coupling facility, IMS stops sharing the message queues and stops to work because there are no queues available for input or output. After the connection is restored, CQS reconnects to the structures and rebuilds them if necessary, and IMS resumes work.

In the case of a resource structure, a coupling facility list structure, CQS does not support structure recovery, structure checkpointing, or overflow processing. To provide optimum performance, CQS does not log changes to resource structures. To aid in recovering resources, you can define resource structures in the CFRM policy to duplex them automatically.

In the case of a nonrecoverable queue structure (RECOVERABLE=NO), CQS does not support structure recovery. When a nonrecoverable queue structure fails, CQS begins recovery processing, and cold starts the newly-allocated structure (initializes it to empty). All active CQSs that were connected to the failed structure then abend with an ABENDU0373. You should restart the CQSs, at which time, the CQSs performs resync processing with any connected clients. You can manually restart CQS, use the z/OS ARM feature (ARMRST=Y), or use your own automation. When CQS restarts after the failure of a nonrecoverable queue structure, all data in the previous (failed) structure is lost.

CQS log recovery

CQS uses a z/OS log stream for its recovery log. You cannot recover if the last offload data set in the z/OS log stream is full and CQS fails. Be sure to schedule CQS system checkpoints frequently enough to prevent your offload data sets from filling. You can define groups of offload data sets for the system log, and z/OS uses each group in turn so that you do not run out of data sets. CQS can recover your system by using these data sets.

CCTL failures

CCTL failures commonly include CCTL region failures, in which a program is not stopped, and when a program is stopped

CCTL region failure

If a CCTL fails, it normally disconnects from the IMS DBCTL system and has no effect on the IMS DBCTL system itself. The CCTL threads are terminated immediately or after they complete their current DBCTL request. These thread terminations appear as DFS554I messages. The programs are not stopped.

What action IMS takes for database changes made (units of recovery), depend on the sync-point state of the thread. If a CCTL fails, a U113 abend of the DBCTL control region can occur.

Related reference

CCTL exit routines (Exit Routines)

CCTL thread failure

A CCTL thread failure can occur if the CCTL thread makes a database resource adapter (DRA) request, and IMS ends abnormally during that request.

When this happens, IMS issues a DFS554I message, and the CCTL indicates the failure to its operator. The program is stopped; you can start it again using the **/START** command after the problem is resolved. If a CCTL thread fails within the CCTL itself, IMS takes no action.

A thread can also fail because of an abend while the thread is within IMS or the DRA. For example, when the IMS DBCTL system fails, all CCTL threads ends abnormally in the DRA with a U002 abend. What happens to the UOR associated with the CCTL thread is determined by its sync-point state. If the failure occurs while UOR is in-flight, IMS backs the UOR out; if the UOR is in-doubt (in the middle of the sync-point process), the UOR remains in-doubt.

Related concepts

Unit of recovery (Application Programming)

CCTL thread looping

Use CCTL commands to determine if the CCTL is looping or if it has issued a DBCTL request and is waiting for a response. Refer to your CCTL documentation if the CCTL is looping. Otherwise, issue the /DISPLAY command to find the region with the recovery token that corresponds to the CCTL task that is waiting for a response from DBCTL. Then use the /STOP REGION command to force that thread to end.

DBCTL failures

A termination of a IMS DBCTL system does not cause connected subsystems to terminate; the subsystems are simply left without DBCTL services. If any of the DBCTL address spaces (DBC, DBRC, or DLISAS) fails, all of its address spaces are terminated.

The state of the DRA is determined by how the CCTL responds to notification of the IMS DBCTL system failure.

Normally, you terminate the IMS DBCTL system using a **/CHECKPOINT FREEZE** command, but you can also use the z/OS **MODIFY** command to force termination, particularly if an IMS shutdown or failure results in a loop. If you use the **MODIFY** command and want a dump of the address space, use the DUMP keyword; otherwise use the STOP keyword. The DBCTL control region terminates with a U0020 abend, and IMS issues the following messages: DFS628I and DFS629I. These messages tell you that an abend is scheduled and what the job name is.

After determining the cause of the failure and correcting it, restart the IMS DBCTL system using the **/ERESTART** command. The CCTL cannot reconnect to the IMS DBCTL system until the **/ERE** command completes. The CCTL has several options on how to respond to a DBCTL failure.

When a CCTL connects to an IMS DBCTL system after a restart, they resynchronize in-doubt UORs automatically. Resynchronization means that the CCTL requests an action (commit, abort, or forget) and the IMS DBCTL system processes that request. If the commit or abort process fails, IMS takes the same actions as in the DB/DC environment. If the commit process fails for systems with DEDBs, IMS sends message DFS2282I to the DBCTL operator. For all failures, IMS removes the EEQEs because the UOR is no longer in-doubt.

If a CCTL that was connected to IMS fails, there could be in-doubt UORs. Normally, the CCTL can restart, reconnect to IMS, and resolve any in-doubt UORs. If the CCTL is cold started, the in-doubt UORs must be manually resolved with a **/CHA CCTL** command to commit or abort the UORs. This process will remove the EEQEs that are associated with the in-doubt UORs.

Do not attempt to in-doubt EEQEs directly by DBRC commands. Use the **/CHA CCTL** command to resolve the in-doubt UORs and clean up the EEQEs.

If a CCTL requests resynchronization of an indoubt UOR, and the DBCTL subsystem has no knowledge of that UOR, IMS returns code 218 to the CCTL, and sends message DFS2283I to the DBCTL operator. The DBCTL operator can use the /DISPLAY command to display the in-doubt UORs known to the IMS DBCTL subsystem.

Related concepts

Recovery in an IMS DBCTL environment (System Administration)

Related reference

CCTL exit routines (Exit Routines)

IRLM failures

When an IRLM fails, IMS subsystems using the IRLM cannot continue normal operation. IMS terminates active programs using the IRLM with abend 3303 and puts their messages on a suspend queue.

For wait-for-input programs using the IRLM, IMS either terminates them with abend 3303 on their next database call or gives them a QC status code. IMS inhibits subsequent program scheduling until IMS reconnects with the IRLM.

Before IMS can back out the IRLM failure, every dependent region with an intent to use a database must terminate its threads. IMS cannot terminate or reconnect to the IRLM until all dependent regions disconnect from the IRLM. Therefore, you must abnormally terminate any region that has not terminated (such as those programs that are waiting in an for a timer expiration or a for a reply to a Write-to-Operator with Reply (WTOR) message.

Restart the IRLM using the z/OS **START** command. Then, reconnect IMS subsystems to the restarted IRLM using the z/OS **MODIFY***jobname***RECONNECT** command. IMS automatically dequeues messages on the suspend queue.

Recovery with data sharing

In a data-sharing environment, you must extend IMS recovery procedures by protecting surviving subsystems from database records that contain incomplete changes left by a failing subsystem. IMS recovers a shared database by restoring the database from the latest image copy.

You can use the merge function provided with the IMS Database Change Accumulation utility (DFSUCUM0) to merge the log data sets from the subsystems that updated the database.

Apply the records from the utility (in the output data sets) for a forward recovery. If only one subsystem in a data-sharing environment updated the database, IMS does not need to use the sequence numbers and the Database Change Accumulation utility, because it has only one set of log data sets to manage.

IMS data sharing also introduces other challenges. Two IMS Internal Resource Lock Managers (IRLMs) might lose communication with each other because:

- An z/OS cross-system coupling facility (XCF) might fail.
- An IRLM might fail, leaving its IMS subsystems running.

- A z/OS system might fail, bringing down the IRLM, VTAM, DBRC, and IMS subsystems running under it.

In a data-sharing environment, IMS must ensure data integrity after failure of one or more components of data sharing.

The IRLM and DBRC work together with IMS to protect shared databases before and after a failure leaves incomplete changes in these databases. This protection remains after a failure until failed subsystems complete backout of incomplete changes.

Related reference

Database Change Accumulation utility (DFSUCUM0) (Database Utilities)

DBRC and protecting data

DBRC provides database-level (level 1) data sharing and database-level protection after a failure occurs. DBRC prevents conflicts in database allocation that could compromise database integrity. DBRC uses the RECON data set to record information about each database authorization it grants to IMS subsystems, and to indicate whether a subsystem is updating a database.

DBRC removes this indication when any of the following occur:

- The subsystem terminates normally.
- Dynamic backout completes after a failure.

If an IMS subsystem fails during an update, the RECON data set still indicates this incomplete update. DBRC does not grant additional database authorizations unless the IRLM retains locks that protect the incomplete changes.

IRLM and protecting data

The internal resource lock manager (IRLM) can protect your data if an IMS subsystem fails.

If the IRLM is present and remains running, it retains locks on incomplete changes at the database record level after an IMS subsystem fails. The IRLM removes these locks when dynamic backout completes.

If the IRLM fails while its IMS subsystems continue to run, the subsystems stop all work in progress, dynamically back out all incomplete database changes, and call DBRC to relinquish their authorizations to shared databases. After you restart the IRLM, the subsystems resume processing.

If two or more IRLMs are active during block-level data sharing, and one IRLM fails with or without the z/OS system that it runs under, the surviving IRLMs and their IMS subsystems reauthorize with DBRC the use of all shared databases.

In this case, because the surviving IRLMs know about the block-level locks (that protect the incomplete changes made by the failed subsystem), they permit their IMS subsystems to continue block-level sharing of the affected databases.

Fast Database Recovery (FDBR) regions

In a sysplex data-sharing environment, if one IMS subsystem fails while it holds locks on data in shared databases or while waiting for work on an external subsystem, the other sharing IMS subsystems must wait for the failed subsystem to restart and release its locks. To reduce the amount of time that the sharing subsystems must wait, you can use IMS Fast Database Recovery (FDBR) regions.

FDBR regions monitor an IMS subsystem and can automatically recover database resources (shared databases and areas) if the monitored subsystem fails. In addition, in-doubt work on an attached external subsystem can be identified by calling the External Subsystem Attach Facility (ESAF) In-Doubt Notification exit routine (DFSFIDN0), which passes an informational message that you can use to automate in-doubt work handling during recovery.

An FDBR region tracks a single subsystem, so you must set up a separate FDBR region for each subsystem that you want FDBR to track.

An FDBR region does not have to run in the same z/OS system as the IMS subsystem it tracks, but both the FDBR region and the IMS subsystem must be in the same z/OS cross-system coupling facility (XCF) group. The FDBR region also must have access to the following IMS data sets:

- ACBLIBx
- MODBLKSx
- MODSTATx
- OLDS
- Restart data set (RDS)
- RECONx
- SDFSRESLx
- WADS

FDBR is similar to XRF in that it operates in distinct phases:

- Surveillance
- Recovery process
- Post-recovery

Restrictions:

- You cannot use an FDBR region with the following types of subsystems: DCCTL, Extended Recovery Facility (XRF), or DBCTL standby.
- While using a tracking system such as FDBR, you must not use a buffer manager that is different from that of the active system.

You do not have to register a CSL with an FDBR region. If you do register a CSL with an FDBR region, it gives you the following benefits:

- a more complete picture of the IMSplex when looking at it from the **QUERY IMSPLEX** command
- a monitor program can watch the status of FDBR using the notifications that SCI sends when an IMSplex member joins and leaves the IMSplex

FDBR surveillance

During the surveillance phase, the FDBR region monitors database activity by reading the IMS log. It monitors all activity for shared databases and areas, and can monitor activity for nonshared areas.

Restriction: An FDBR region does not track MSDB activity.

If the IMS subsystem that the FDBR region is tracking fails, the FDBR region automatically begins recovery for all tracked databases and areas. If the IRLM for the tracked IMS subsystem fails, or if there is no log activity from the tracked IMS subsystem for a specified length of time, the FDBR region issues a message telling the operator to begin FDBR recovery.

FDBR recovery process

If the tracked IMS subsystem fails, the FDBR region automatically begins recovery. During automatic recovery, the FDBR region uses the MVS Availability Manager to determine when I/O is complete for the databases and areas it will recover.

During recovery, the FDBR region does the following for all tracked databases and areas:

- Allocates and opens database data sets
- Backs out updates for full-function databases
- Performs REDO processing for DEDB areas

After recovery is complete, the FDBR region tells IRLM to release retained locks for the recovered databases and areas. The FDBR region then issues a message that recovery is complete.

If the tracked IMS subsystem's z/OS cross-system coupling facility (XCF) fails, the FDBR region stops tracking and issues a message. If the FDBR region's XCF fails, the region abnormally terminates.

To begin recovery manually, issue the z/OS **MODIFY***fdbrproc*`, RECOVER` command. For manual recovery, the FDBR does not wait for I/O to complete because it assumes the I/O is already complete.

Online change and FDBR

FDBR tracks online changes through the x'70' log records.

To avoid problems if FDBR is shut down or restarted after an online change is performed, or to avoid problems under other circumstances after online change, take one or more IMS system checkpoints after the online change is complete and before FDBR is shut down or restarted. Doing so ensures that FDBR restarts from a checkpoint that was taken after the last online change.

For example, before you restart FDBR after an online change, issue the /CHECKPOINT command on the active system to take a simple checkpoint and verify from the DFS3804I message that the latest restart checkpoint time stamp is after the online change.

FDBR post-recovery

When recovery is complete, all shared databases and areas are available to other sharing IMS subsystems in the sysplex. You can restart the failed IMS subsystem at any time.

During IMS restart, IMS does the following to complete recovery:

- Writes updates for shared VSO areas to DASD
- Recovers nonshared DEDB areas that were not tracked by the FDBR region
- Recovers MSDBs

After the IMS subsystem is running, you can reestablish tracking by the FDBR region by issuing the z/OS **START***fdbrproc* command.

Commands for using FDBR

Use the commands for Fast Database Recovery (FDBR) to start and terminate tracking, obtain status, and initiate recovery. For each command, `fdrproc` represents the FDBR procedure name.

About this task

Starting tracking

Normally, you start FDBR with a JCL job, submitted during system start, after the tracked IMS is started.

About this task

You can also issue the z/OS **START** command to start FDBR tracking of an IMS subsystem, as follows:

Procedure

```
S fdrproc
```

Obtaining status

You use the z/OS **MODIFY** command to obtain Fast database recovery (FDBR) status.

About this task

```
F FDR1,STATUS
```

This command returns the following information:

```
DFS000I:  PHASE:   TRACKING  LOG-TIME:  17:48:41  FDR1
DFS000I:  ACT-ID:  SYS3      GROUPNAME: FDRSYS3   FDR1
DFS000I:  TIMEOUT: 060 SEC   AREA01:    NORECOV   FDR1
DFS000I:  SVSOOPEN: SERIAL   FPBUFF:    LOCAL     FDR1
```

The returned information includes:

PHASE
This field indicates the phase of the FDBR region. You can display the following phases:

INIT (initialization phase)
TRACKING (tracking phase)
RECOVERY (recovery phase)

ACT-ID
The IMS ID that FDBR is tracking. If the FDBR region is in initialization phase, N/A is shown.

GROUPNAME
z/OS cross-system coupling facility group name used for XCF monitoring. If the FDBR region is in initialization phase, N/A is shown.

TIMEOUT
XCF timeout value. If the FDBR region is in initialization phase, N/A is shown.

LOG-TIME
Time associated with the log record currently being processed by FDBR region. The length of time the FDBR lags behind the IMS in reading the log is the difference between the current time, as shown by the time stamp and the log time. If the FDBR region is in initialization phase, N/A is shown.

AREA01=
This field shows the share level 0|1 DEDB area recovery option as follows:

NORECOV
NORECOV option is specified in DFSFDR*xx* IMS.PROCLIB member.

RECOV
RECOV option is specified in DFSFDR*xx* IMS.PROCLIB member.

Initiating recovery

Use the z/OS **MODIFY** command to initiate FDBR.

About this task

You can use this command if FDBR determines a time out status during z/OS cross-system coupling facility (XCF) or log surveillance. The command is not accepted if FDBR is being initialized or is already in a recovery process.

Procedure

The following command starts FDBR immediately: `F fdrproc,RECOVER`

Terminating tracking

Use the z/OS **MODIFY** command to terminate FDBR.

About this task

The following command stops FDBR tracking activity:

```
F fdrproc,TERM
```

The command is not accepted if FDBR is being initialized or is in a recovery process.

Stopping the FDBR region

You can use z/OS **MODIFY** commands to stop the FDBR region functions.

Procedure

1. The following command ends FDBR functions without producing a dump:
   a) `F fdrproc,STOP`
   FDBR ends with a return code of 0200.
2. The following command ends FDBR functions and produces a dump:
   a) `F fdrproc,DUMP`
   FDBR ends with return code of 0020.

User access problems

The task of determining and responding to user access problems can be done by the IMS MTO, the network support group, or a user liaison group.

If your installation has a user liaison group that helps end users with problem determination, you need to explain what information that group should obtain from the end user before they contact the IMS MTO. This information includes a description of the problem's symptoms (such as poor response time or no response at all), details of any system or application program error messages received, terminal status lights, and so forth. Be sure to give the user liaison group guidelines on whether to call the network support group, application support group, or IMS MTO.

After they have gathered all information from the end user, they should develop a plan for determining the cause of the problem and, if possible, correcting it. Typically, user problems can be classified as:

- Poor system response
- No response, or terminal keyboard locked
- Unexpected output (system or application program error messages)

If the general system response is poor, the MTO should follow the procedures described in "Monitoring the system" on page 105.

If the end user is getting no response at all or has a locked keyboard, the problem might be in the network or in the status of the application program resources required by the transaction. For example, an OLDS might need to be archived to free additional logging resources. Use one of the following commands to determine if any of the required resources are unavailable:

- **/DISPLAY STATUS**
- **/DISPLAY OLDS**
- **QUERY DB SHOW(STATUS)**

If the end user is receiving application program error messages or invalid output, the MTO should follow your procedures for contacting the appropriate application support group. You should define the documentation that MTO need to provide to the support groups, including a dump of the message region (if one was produced), or exact details of the original transaction the user entered. For IMS or VTAM system messages, the MTO should follow the appropriate error recovery procedures.

Related concepts

"Monitoring the system" on page 105

In order to gather problem determination and performance information, you need to monitor the status of the system on a regular schedule. For example, to determine if you should start an extra message region, you might monitor the status of the queues during peak load.

Recovery points

If you use either full-function or Fast Path DEDB databases, you can create recovery points in the database without having to take IMS offline. By creating recovery points you will improve the recovery of the database when an outage occurs.

When you quiesce the database you establish a point of consistency, which is when there are no updates pending for the database and the information stored on a direct access storage device (DASD) accurately reflects the current information stored in the database. When this point of consistency is reached, you create an image copy that you can use to provide a quick method of recovering the database to this point.

Important: This process must occur on every IMS in an IMSplex that is actively using the database.

You can create this point of consistency without making the database unavailable to applications. When the quiesce command is invoked, all updates that are in progress are committed and stored on DASD. However, the next DL/I call that attempts to access the database results in a wait until the quiesce on the database is released. Once the quiesce is released, these applications can access the database again.

There are two ways to quiesce the database using the **UPDATE DB** command with the START(QUIESCE) keyword:

START(QUIESCE)
This command causes the database to become quiesced and then releases the quiesce, thereby creating the new recovery point for the database. This is the default form of the quiesce command.

START(QUIESCE) OPTION(HOLD)
This command causes the database to become quiesced and keeps the database quiesced until an **UPDATE DB STOP(QUIESCE)** command is issued that releases the quiesce. While the database is quiesced, you can create image copies of the database data sets.

Creating recovery points

You use the IMS single point of control (SPOC) OM API to enter the **UPDATE DB** command with the START(QUIESCE) keyword to establish a recovery point. You can also enter an **UPDATE DB** START(QUIESCE) OPTION(HOLD) command to have a recovery point established, and keep the database quiesced.

About this task

The **UPDATE** START(QUIESCE) command stops all updates in progress and saves them to a direct access storage device (DASD). When this command is issued, all open ALLOC records on any sharing systems are closed. When the command completes, you have a recovery point, and applications can continue to make updates to the database.

To create recovery points for high-activity databases:

Procedure

1. Enter one of the following commands from IMS SPOC:

Option	Description
UPDATE AREA NAME*(area_name)* START(QUIESCE)	To quiesce a Fast Path area and establish a new recovery point.
UPDATE DB NAME*(db_name)* START(QUIESCE)	To quiesce a full-function database and establish a new recovery point.

Option	Description
UPDATE DB NAME*(HALDBmaster_name)* START(QUIESCE)	To quiesce a full-function HALDB. **Note:** When the quiesce function is invoked on a full-function HALDB, it affects all of the partitions of the HALDB. The command does not set on the quiesce in progress flag on in the RECON data sets for the HALDB master. Instead it sets the quiesce in progress flag on in each HALDB partition.
UPDATE DB NAME*(DEDB_name)* START(QUIESCE)	To quiesce a Fast Path DEDB and establish a new recovery point.
UPDATE DATAGRP NAME(dbgrp_name) START(QUIESCE)	To quiesce a database group and establish a new recovery point.

IMS SPOC returns the following output:

```
DBName   MbrName    CC
DBXYZ    IM02        0
DBXYZ    IM01        0
DBXYZ    IM03        0
```

2. Verify that the command has processed successfully. Enter the following command from IMS SPOC: **QRY DB NAME***(DBXYZ)* SHOW(STATUS)

   IMS SPOC returns the following output:

```
DBName    MbrName    CC TYPE     LclStat
DBXYZ     IM03        0 DLI      ALLOCS,OPEN
DBXYZ     IM01        0 DLI      ALLOCS,OPEN
DBXYZ     IM02        0 DLI      ALLOCS,OPEN
```

3. Create a concurrent image copy of the database. Verify that the image copy completes successfully.

Creating recovery points and keeping the database quiesced

The **UPDATE DB START(QUIESCE)** OPTION(HOLD) command causes the database to become quiesced and keeps the database quiesced until an **UPDATE DB STOP(QUIESCE)** command is issued that releases the quiesce. While the database is quiesced, you can create image copies of the database data sets.

About this task

To create recovery points for low-activity databases:

Procedure

1. Enter one of the following commands from IMS SPOC:

Option	Description
UPDATE AREA NAME(area_name) START(QUIESCE) OPTION(HOLD)	To quiesce a Fast Path area and establish a new recovery point and keep the area quiesced.
UPDATE DB NAME*(db_name)* START(QUIESCE) OPTION(HOLD)	To quiesce a full-function database and establish a new recovery point and keep the database quiesced.
UPDATE DB NAME*(HALDBmaster_name)* START(QUIESCE) OPTION(HOLD)	To quiesce a HALDB and establish a new recovery point and keep the HALDB quiesced. **Note:** When the quiesce function is invoked on a full-function HALDB, it affects all of the partitions of the

Option	Description
	HALDB. The command does not set the quiesce in progress flag on in the RECON data sets for the HALDB master. Instead it sets the quiesce in progress flag on in each HALDB partition.
UPDATE DB NAME*(DEDB_name)* START(QUIESCE) OPTION(HOLD)	To quiesce a Fast Path DEDB and establish a new recovery point and keep the database quiesced.
UPDATE DATAGRP NAME*(dbgrp_name)* START(QUIESCE) OPTION(HOLD)	To quiesce a database group and establish a new recovery point and keep the database group quiesced.

The following IMS SPOC output is returned:

```
DBName   MbrName    CC
DBXYZ    IM02        0
DBXYZ    IM01        0
DBXYZ    IM03        0
```

2. Check that the command executed successfully. Enter the following command from IMS SPOC: **QRY DB NAME***(DBXYZ)* SHOW(STATUS)

   IMS returns the following results:

```
IMS SPOC output:
DBName    MbrName    CC TYPE     LclStat
DBXYZ     IM03        0 DLI      ALLOCS,OPEN,QUIESCED
DBXYZ     IM01        0 DLI      ALLOCS,OPEN,QUIESCED
DBXYZ     IM02        0 DLI      ALLOCS,OPEN,QUIESCED
```

3. Create an image copy of the database.

   Verify that the image copy completes successfully.

   The database should be successfully backed up.

4. Enter one of the following commands to release the quiesce on the database, which is now successfully backed up.

Option	Description
UPDATE AREA NAME*(area_name)* STOP(QUIESCE)	To release the quiesce on a Fast Path area that was previously quiesced.
UPDATE DB NAME*(db_name)* STOP(QUIESCE)	To release the quiesce on a database that was previously quiesced.
UPDATE DB NAME*(HALDBmaster_name)* STOP(QUIESCE)	To release the quiesce on a HALDB that was previously quiesced.
UPDATE DB NAME*(DEDB_name)* STOP(QUIESCE)	To release the quiesce on a Fast Path DEDB that was previously quiesced.
UPDATE DATAGRP NAME*(dbgrp_name)* STOP(QUIESCE)	To release the quiesce on a database group that was previously quiesced.

IMS SPOC returns the following output:

```
DBName   MbrName    CC
DBXYZ    IM03        0
DBXYZ    IM02        0
DBXYZ    IM01        0
```

5. Enter the following command to ensure that the quiesced database has been released successfully: **QRY DB NAME***(DBXYZ)* SHOW(STATUS)

IMS SPOC returns the following output:

```
DBName    MbrName    CC TYPE     LclStat
DBXYZ     IM03        0 DLI      ALLOCS,OPEN
DBXYZ     IM01        0 DLI      ALLOCS,OPEN
DBXYZ     IM02        0 DLI      ALLOCS,OPEN
```

Executing recovery-related functions

While IMS is running, an IMS system programmer or operator might need to execute functions relating to the recoverability of the system. You can initiate recovery through the type-1 and type-2 IMS commands.

About this task

You must ensure that resource definitions have been exported to a resource definition data set (RDDS) if you use DRD and perform a cold start from an RDDS. Otherwise, any resource changes made dynamically are lost across a cold start.

Issuing DBRC commands

You can issue batch and online (/RMxxxxxx) commands and DBRC API requests to use DBRC functions. Authorization checking is supported for DBRC commands that are submitted as batch or online commands.

About this task

The **/RMxxxxxx** commands perform the following DBRC functions:

- Record recovery information in the RECON data set
- Generate JCL for various IMS utilities and generate user-defined output
- List general information in the RECON data set
- Gather specific information from the RECON data set

The IMS security administrator can use a security product like RACF (Resource Access Control Facility), a security exit routine, or both to control authorization of DBRC commands. Commands issued through the z/OS console are always authorized.

Recommendation: Allow operators or automation programs to issue the **/RMLIST** command (or DSPAPI FUNC=QUERY API request) and the **/RMGENJCL** command (or DSPAPI FUNC=COMMAND COMMAND=GENJCL API request). Restrict the use of **/RMCHANGE**, **/RMDELETE**, and **/RMNOTIFY** commands (or equivalent API requests), because they update the RECON data sets.

Attention: When the **/RMLIST** command is issued from the OM API (such as TSO SPOC), the amount of output that is generated can be large. Because the RECON data set is reserved during command processing, if the generated output is large, do not issue the command during peak times. To prevent the command from being issued with the DBRC='RECON' option inadvertently, consider protecting the command with a security product.

Related concepts

DBRC and data sharing support (System Administration)

Supporting data sharing (System Administration)

DBRC API (System Programming APIs)

Dumping the message queues

You use the **/CHECKPOINT SNAPQ** command to save the message queues in a nonshared-queues environment. This command dumps the message queues to the log without terminating the online system.

About this task

Recommendation: Schedule the **/CHECKPOINT SNAPQ** regularly because it shortens the time required for emergency restart if a problem occurs on the message queue data sets. Consider the following intervals:

- Whenever the OLDS is switched
- Once each hour
- Once each shift
- Twice each day (midnight and noon)
- Once each day

For a shared-queues environment, use the **/CQCHKPT SHAREDQ** command to dump the shared queues.

Recovering the message queues

In a non-shared-queues environment, you can recover the message queues during an IMS restart if the previous shutdown included the DUMPQ or the SNAPQ keyword.

About this task

Specify the BUILDQ keyword on the **/NRESTART** or **/ERESTART** command to restore the messages to the message queue data sets from the IMS log. Specify the FORMAT keyword on the **/NRE** or **/ERE** command if you also want to reinitialize the message queue data sets.

In order to use the **/NRE BUILDQ** command, the system must be shut down using a **/CHECKPOINT DUMPQ | PURGE** command. To use the **/ERE BUILDQ** command, you need only a prior **/CHECKPOINT SNAPQ** command.

Restriction: If a **/NRE BUILDQ** or **/ERE BUILDQ** command fails and you cold start IMS or cold start DC, messages are lost and are not processed.

You can use the IBM IMS Queue Control Facility for z/OS to select messages from the OLDS (or SLDS) and reinsert them into the IMS message queues after an IMS cold start.

For a shared-queues environment, CQS automatically rebuilds the message queues if the coupling facility fails. You can also use the **SETXCF START,REBUILD** command to rebuild the queues manually.

Related reference

IMS Queue Control Facility overview

IBM IMS Queue Control Facility for z/OS interface (Diagnosis)

Archiving the OLDS

You should archive the OLDS to an SLDS at regular intervals. If you are not using automatic archiving, the MTO or automation program should issue the DBRC **GENJCL** command (or DSPAPI FUNC=COMMAND

COMMAND=GENJCL API request) at regular intervals to generate the JCL for the Log Archive utility, and should execute the utility.

Making databases recoverable or nonrecoverable

You can change recoverable full-function DBs to nonrecoverable (after deleting recovery-related records from the RECON data set) by using the **CHANGE.DB NONRECOV** command.

About this task

You can change to recoverable again by using the **CHANGE.DB RECOVABL** command.

Use the **LIST.DB** command or the DBRC QUERY TYPE DB API request to display whether a database is recoverable.

Running recovery-related utilities

Depending on your recovery strategy, the MTO might be responsible for running various recovery-related utilities at regular intervals.

These could include:

- Database Image Copy utility
- Database Image Copy 2 utility
- Online Database Image Copy utility
- Database Change Accumulation utility

The MTO should also run these utilities when a database changes from nonrecoverable to recoverable.

Related reference

Database Image Copy utility (DFSUDMP0) (Database Utilities)
Database Image Copy 2 utility (DFSUDMT0) (Database Utilities)
Online Database Image Copy utility (DFSUICP0) (Database Utilities)
Database Change Accumulation utility (DFSUCUM0) (Database Utilities)

Chapter 6. Developing operating procedures

After you have made basic design decisions about operations and recovery, you are ready to implement these decisions. Primarily, implementation takes the form of developing procedures for operating IMS.

About this task

The importance of developing good procedures cannot be over emphasized. The installations with the most success are those with well-written, well-maintained, well-tested, and well-understood procedures.

Establishing procedures for operating IMS is not a simple task. Because each installation has unique requirements and resources, no single approach to operations is possible. These topics provide some basic guidelines for establishing procedures.

Operations personnel

It is important that you identify the operational requirements for IMS, and identify the people responsible for performing the various operations tasks. After assigning responsibilities, you are ready to set up the procedures necessary to coordinate the operations task.

Many people are involved, directly or indirectly, in the operation of a successful IMS system. To help you understand who needs operating procedures and what it is they are responsible for operating, these topics look briefly at the various functions performed by these people. This topic also gives you an understanding of the context in which MTOs operate the system, and users or remote terminal operators use it. A fundamental understanding of the MTO is important because the subsequent topics describe the procedures you develop for the MTO and user.

An IMS DBCTL environment includes no IMS terminals, so the DBCTL operator must use a z/OS console. Throughout these topics, this DBCTL operator is called the IMS MTO.

The following groups of people represent functions often defined at an installation:

- **z/OS and JES system operators**. This group is responsible for operating z/OS and JES.

  If the IMS MTO does not have access to a z/OS console, you need to establish procedures so the MTO can communicate with the z/OS operator in case of a failure. This is particularly important when the MTO is physically distant from the z/OS console.
- **Network support group**. This group is responsible for operation of the network, including: VTAM, physical telecommunications lines, and modems.

  You must establish procedures for communication between the MTO and network support personnel for initializing the online system and isolating and correcting errors related to the IMS network.
- **DASD and tape pool operators**. This group is responsible for managing tape and disk media.

  The MTO must communicate with DASD and tape operators when mounting archive tapes, and when creating image copies, change accumulations, and so forth. Communication is also required when the MTO is analyzing and correcting I/O errors.
- **User liaison group**. This group is the primary interface for the end user.

  The user liaison group needs procedures for answering user questions relating to both IMS and individual application programs. The user liaison group also needs procedures for problem determination and correction, and for communicating with the MTO when unable to solve a problem. The user liaison group might also need to communicate with application support groups when a user problem relates to a specific application program rather than to the IMS subsystem.
- **Application support groups**. This group develops and maintains IMS application programs.

  You need to identify representatives for the support of each application running on the IMS subsystem. You should give the user liaison group and the IMS MTO procedures for contacting this group in case of application problems.

- **IMS system administrator, system programmer, or recovery specialist**. This person (or group) is responsible for installing IMS and designing the system configuration.

  There are certain error conditions the MTO cannot correct. For such errors, the MTO needs instructions on how to contact the IMS system administrator, system programmer, or recovery specialist. This person (or group) will help resolve the more complex recovery problems.

The following figure provides an overview of the relationships between these functional groups.

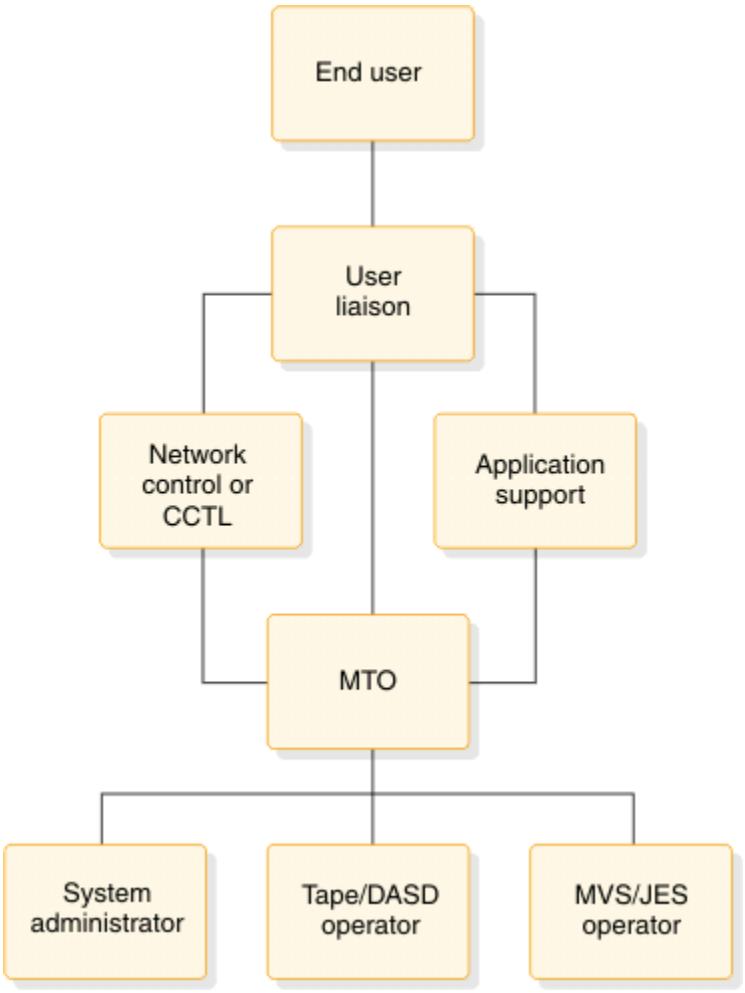


Figure 6. Relationships among operations groups

Depending on the size of your installation, some of these functions might be performed by the same person or group. In large installations, the same function might be performed by many different people. The IMS MTO, for example, might have access to a z/OS system console. This access would allow the MTO to enter z/OS START commands.

In this book, operations procedures are classified into two types:

- Those for operating IMS, which are generally performed primarily by the MTO and secondarily by the system administrator or recovery specialist
- Those for use by an end user

Developing procedures for an end user is not applicable in an IMS DBCTL environment, which has no DBCTL terminal end users. References to end users for a DBCTL environment actually refer to the CCTL end users.

Related concepts

"Developing user procedures" on page 213

End users (remote terminal operators) are the people who use the terminals connected to IMS. End-user procedures should focus on the needs of end users, not on IMS operation.

Establishing operating procedure documents

You need to establish operating procedure documents as part of your overall strategy for controlling an IMS system. This will help you to ensure that operators will handle various situations in a consistent manner.

Establishing operations interactions

The online IMS environment should be transparent to general business procedures. It is probably only part of the data processing services. Your procedures, to some degree, reflect your installation's organization.

Definitions of support groups

Support groups provide specific functions for different groups of users, different environments, and even for different operations.

An *IMS operations group* must implement procedures and use tools, often specified by system programmers, that originate from requirements for database and data communication administration. The requirements for scheduling jobs, handling tapes and direct access devices, and producing printed output are developed in conjunction with general system operations procedures.

The *system operations group* provides z/OS support, and also often supports the hardware. The *software support group* provides support for the operating system and the language compilers. The *applications support group* provides support for the application programs, and is represented by user liaison and development maintenance groups.

The *user liaison group* is of special importance. Most of the problems that end users experience are application-program dependent. The user liaison group, acting as first point of contact, can answer such questions, but operations cannot. The user liaison group can also be the channel for problem resolution. They can initiate problem tracking, and their interpretation of error events can help all end users. The user liaison group analyzes problems to determine if they are application-program related, procedural, hardware, or system software. They can contact the master terminal control group regarding network or operations problems, or interact with application-program development support for programming errors. Very often, they are key personnel in any decisions affecting recovery, because they must interpret the actions taken on behalf of end users.

Operations contacts and responsibilities

Every organization needs to have layers of support and advice for control responsibilities for day-to-day operations. You need to plan for casual support and expert support.

Some suggestions for assigning control responsibilities are:

- Have a user liaison group answer day-to-day application-program questions, track problems, and monitor service.

  Make the user liaison group the primary contact for the end user and for IMS operations. Encourage the user liaison group to be part of recovery decisions.
- Have a database-administration contact for recovery, design change evaluation, and database monitoring and analysis.
- Shield the MTO from most end-user inquiries so that the MTO can concentrate on availability, system status, and production statistics. Use other supervisory terminals to distribute the workload. Take advantage of automated operator programs to initiate MTO commands and reduce operator tedium.
- Establish system operations contacts for VTAM operation and hardware problems.
- Have a security contact for access problems. Other duties could be password and security maintenance, as well as violation monitoring.

- Have a monitoring and performance analysis coordinator who draws upon specialists for detailed studies.

These suggested contacts might only be suitable for prime shift or peak production hours. You should plan a contact list for other times, such as production cycle overruns and off-shift periods.

Plan for procedure requirements

The established operating procedures reflect many decisions and responsibilities. Development of the procedures takes place during the majority of the application implementation stages.

The content of the procedures draws upon several other administration tasks:

- The online IMS design as reflected in system definition
- The variety of application operation requirements
- The database administration procedures

The following table summarizes the tasks involved in developing a procedural document, relates them to subtasks, and briefly indicates the other groups of personnel involved. The tasks toward the bottom of the figure (starting with "Testing") are concerned with refinement of the procedures.

Table 22. Elements for establishing operation procedures

Development task	Operations procedure subtasks	Related support group
Initial structure	Plan terminal operator information Verify IMS message use Plan for automated operator Plan organization and contacts	User liaison (System support) Development
Application requirements	Plan scheduling Plan database availability Assess operator intervention Plan output handling	User liaison and DBA DBA User liaison and DBA Operations
Network control	Plan startup terminals Coordinate operator responsibilities Plan phased connections Plan alternate connections	User liaison and DBA Operations Operations Operations
System definition	Assess restrictions for MTO	Operations
Security	Design procedures Assess operator requirements	User liaison Security control
Administration	Establish system log control Plan error recording Plan network control desk	Operations Operations Operations
Problem procedures	Plan resource recovery	DBA and Development
Monitoring procedures	Plan IMS Monitor use Plan statistics gathering	Performance support User liaison
Recovery procedures	Decide on operator control Decide on problem analysis	Development Support groups and DBA
Testing	Respond to feedback	Development

Table 22. Elements for establishing operation procedures (continued)

Development task	Operations procedure subtasks	Related support group
Master terminal operator training	Respond to feedback	User education
Production control	Audit activities Respond to feedback	Operations User liaison
Modifications to applications	Respond to application-program changes Make tuning changes	Development Performance support

Planning the content of procedures

Your procedures for the master terminal operator (MTO) will vary depending upon their knowledge of IMS operations and the application-program logic

The operating procedure includes four types of instructions:

- The sequence of events for normal operation
- The information to be recorded
- The responsibilities and degree of autonomous action for operators
- The steps to be taken for error incidents or recovery action

Establish the pattern for normal processing by accumulating and combining the different requirements of both end users and databases.

The information the operators should record primarily concerns recovery actions; however, they should also record events that occur during the production cycle (in a form that can be audited). Many installations use a run book or system operation log for recording events during normal operations.

Whom you choose to have operational control strongly influences your procedure documentation. The instructions can vary a great deal, depending on their knowledge of IMS operation and the application-program logic. Based on your needs, you can train your MTO in one or more of the following areas:

- General operations
- Control of an application on a dedicated system
- Service for multiple applications

The arrangements you make to handle error or recovery events are also important in your control of the online IMS system. A knowledgeable operator can restart IMS and fall back to a valid checkpoint. An operator who suspects database damage can involve personnel from database administration or can schedule the required backout or change accumulation jobs.

A suitable document to guide the actions of MTOs should have several sections:

- An introduction

  This section gives an overview of IMS operation (both generally and specifically for your installation), how the network is controlled, and how the master terminal is used. The intended audience is new or substitute operators.
- Operating procedures

  This section shows the normal progression of events, but also provides supplementary instructions for error conditions. You can organize this section as a series of flowcharts.
- Commands reference section

  This section shows syntax and recommendations for using IMS commands and other commands as appropriate. Include only those command keywords and parameters that you consider necessary for

situations the MTO should handle without assistance from other groups. You might also want to include customized explanations of selected IMS messages to guide the MTO's actions.

- Configuration tables

  This section contains descriptions of the IMS and system configuration. These descriptions help the operators to visualize the resources under their control. For example, you can include a chart of terminal connections related to application-program use, or representations of the databases that are online (together with the logical dependencies). You could also display larger copies in the operations area.

- Supplementary procedures

  This section contains procedures not covered in earlier sections. With any large system, you can make alternative configurations available. This section can also contain recovery procedures indexed by symptom.

- A chronological record of events

  This section lists the various events an MTO is likely to encounter during normal operations. This section could also list the types of events an MTO should record in an operations log or run book. Because an online IMS operation is likely to span several operator shifts, some type of record of the MTO's actions is desirable.

- A problem-reporting procedure

  This section describes how to report problems or unusual events. An incident report is a useful tool to help you control the integrity of the system and service for the end user. Any of the following personnel or groups could fill in this report form: the end user or user liaison group, the IMS MTO or network administrator, the database administrator, or the system programmer.

  Recommendation: Regardless of who fills in the form, you should require that it be completed, even though the person competing the form contacts other operating personnel. This form can become a controlling document in the case of a recovery action, but can also be used for minor complaints. To foster cooperation, you should return a copy of the report, showing actions taken, to the originator. You should also periodically review these reports to discover recurring problems and make valuable procedural changes.

Related concepts

“Procedures for user terminal operators” on page 213
The application-development cycle should include the development of end-user procedures, along with education about the transactions and terminal hardware.

Related tasks

“Planning availability of IMS service” on page 170
In order to provide a framework for the operator control of the online IMS subsystem, you need to obtain overall requirements. You can then proceed gradually to fill in the details of specific requirements of various user groups.

How to set the level of operator control

IMS is designed so processing is controlled by message traffic, which places control of the operation primarily in the hands of the master terminal operator. From an administration point of view, control of operations is far more than a structured use of IMS commands.

For a complex system such as IMS, you need to carefully match operational control to operator capability. You must decide whether an operator or an application administrator should have primary control.

The MTO can have a wide range of responsibilities. At one end of the spectrum, you have an operator using a standard set of instructions, designed using a cook-book approach: "if such-and-such happens, perform procedure X". At the other end of the spectrum, you have a system programmer who understands the flexibility of IMS control as well as the application program and database requirements. The first approach requires good symptom-based instructions; the second requires clear guidelines and periodic auditing.

If an operator handles your operations, the two areas for which they need instructions, detailed to a level appropriate for their knowledge and responsibilities, are:

- Normal operations

  You need to document instructions for making specific resources available for committed end-user service. The instructions must be especially clear about when and how to shut down and restart IMS. If alternative network connections are part of the system design, the instructions should have a set of additional procedures for them.

  Operators who understand the end-user priorities should be allowed to change the scheduling algorithm by making reassignment of message priorities or even of message region classes.
- Error and recovery operations (triggered by symptoms or user feedback)

  These events range from temporary transaction processing problems to problems that have serious database implications. Your recovery procedures should include carefully documented and tested actions for your operators to take.

  Recommendation: Design your procedures so the operator initiates recovery actions in cooperation with your user-liaison group and your system-support personnel. Structure your operations to take advantage of as much consultation as possible. Insist that your operators record all recovery actions in a run log, including justification for decisions made.

  An important part of the operations strategy is the recovery procedures used to restart the online IMS subsystem after a failure of the operating system or the IMS control region.

Record-keeping procedures for the MTO

The master terminal operator (MTO) needs to keep various types of information that document system activity. Consider designing standard forms or online panels for recording this information.

The MTO needs to record the following general types of information:

- IMS control region activity
- Utility executions
- Application program activity
- Problems or unusual events
- Startup, shutdown, and system log activity

Some of this information can be recorded automatically. DBRC records all OLDS and archive activity, and can also record the execution of batch jobs and recovery-related utilities.

By default, the master secondary terminal provides a hardcopy log of IMS system activity, including checkpoint information and IMS messages. The logging of IMS system messages to the secondary master terminal can be disabled by using the **/SMCOPY MSG OFF** command.

You can also have the secondary master terminal provide a hardcopy logging option for certain IMS commands and command responses by specifying the COPYLOG keyword in the COMM system definition macro or by issuing the **/SMCOPY** command with the appropriate keyword.

The information logged to the secondary master terminal can be helpful when tracking a series of events affecting the operation of IMS, or when verifying that an operations task was performed. Information useful for analyzing an IMS error situation is often in the z/OS hardcopy log.

You can select whether IMS logs commands issued from the master terminal or commands entered from other terminals, or both. In either case, IMS only logs certain commands. When you issue a command from a terminal other than the master terminal, IMS logs output and identifies its node, or line and physical terminal number.

The **/SMCOPY** command starts and stops the hardcopy logging facility for commands, system messages, or both.

IMS logs the following commands to the secondary master terminal:

/ACTIVATE
/ALLOCATE
/ASSIGN
/CHECKPOINT
/CLSDST
/COMPT
/DBDUMP
/DBRECOVERY
/DELETE
/DEQUEUE
/DISPLAY
/IDLE
/MODIFY
/MONITOR
/MSASSIGN
/OPNDST
/PSTOP
/PURGE
/QUIESCE
/RCLSDST
/RCOMPT
/RMCHANGE
/RMDELETE
/RMGENJCL
/RMINIT
/RMLIST
/RMNOTIFY
/RSTART
/START
/STOP
/SWITCH
/TRACE
/UNLOCK
SYSTEM

Recording IMS control region activity

You should design a form on which the master terminal operator (MTO) can document executions of the control region. Details to be recorded on the form include: the time when the system is started or shut down, the type of restart, and the reason for shutdown. You should make provisions on the form for recording error situations, such as DASD I/O errors, and actions taken to recover from such errors.

The operator can file the output of the DBRC **LIST.LOG** command (or `DSPAPI FUNC=QUERY` API request) with this form to provide a record of log archive activity.

Utility recording forms for IMS utilities

How much manual record keeping is necessary for IMS utilities depends on your use of DBRC. If you have registered all your databases with DBRC, DBRC records the execution of the recovery-related utilities in the RECON data set. By listing the RECON data set (**LIST.RECON** command or **DSPAPI FUNC=QUERY** API request) at regular intervals, you can easily maintain records of utility executions.

If you have not registered all your databases with DBRC, you should design forms for recording execution of the following utilities:

- Database Image Copy
- Database Image Copy 2
- Online Database Image Copy
- Database Change Accumulation
- Database Recovery
- Those utilities involved in database reorganizations

You also need to record the execution of batch database reorganizations. A database cannot be backed out to a point earlier than its latest batch reorganization. Nor can a forward recovery be done from an image copy preceding the latest batch reorganization. Thus, a knowledge of recent reorganizations is important when recovering a database.

The following graphics are sample forms for different utilities. These forms assume that changes for all databases are accumulated together. The first graphic shows the input log data sets, and the output change accumulation data sets.

Run		Data set name	Volume serial number	Purge		Input data sets*	
Date	Time			Date	Time	Name	Volume serial number
91005	1520503	IMS.CUM	CUM03	91004	1200000	IMS.CUM	CUM02 SLDS01 RLDS01
91006	1510031	IMS.CUM	CUM04	91005			

*Includes old change accumulation data set, and SLDS or RLDS

Figure 7. Sample form: running of the Database Change Accumulation utility

Image copy record for database _____ data set_____

Creation		Data set name	Volume serial number
Date	Time		
91004 91005	1100100 1100051	DB1.IC1 DB1.IC2	DB1IC1 DB1IC2

Figure 8. Sample form: running of the Image Copy utility

These forms are only samples. Use them as templates to design forms that suit your installation's procedures.

Application program recording forms

You might want to design forms to record the execution of BMPs and batch application programs. The latter is particularly important if you are not using DBRC to record logs produced by batch programs. Other information you might want to record includes checkpoint IDs and any error messages produced by the programs.

Forms for recording problems or unusual events

You might want to design an incident report form for recording any system problems or unusual events. Incident reports help you control the integrity of the system and services for end-users. This form might be filled in by the user liaison group, the MTO, or the recovery specialist.

Recommendation: Regardless of who fills in the form, you should require that it be completed, even though the person competing the form contacts other operating personnel. This form can become a controlling document in the case of a recovery action, but can also be used for minor complaints. To foster cooperation, you should return a copy of the report, showing actions taken, to the originator. You should also periodically review these reports to discover recurring problems and make valuable procedural changes.

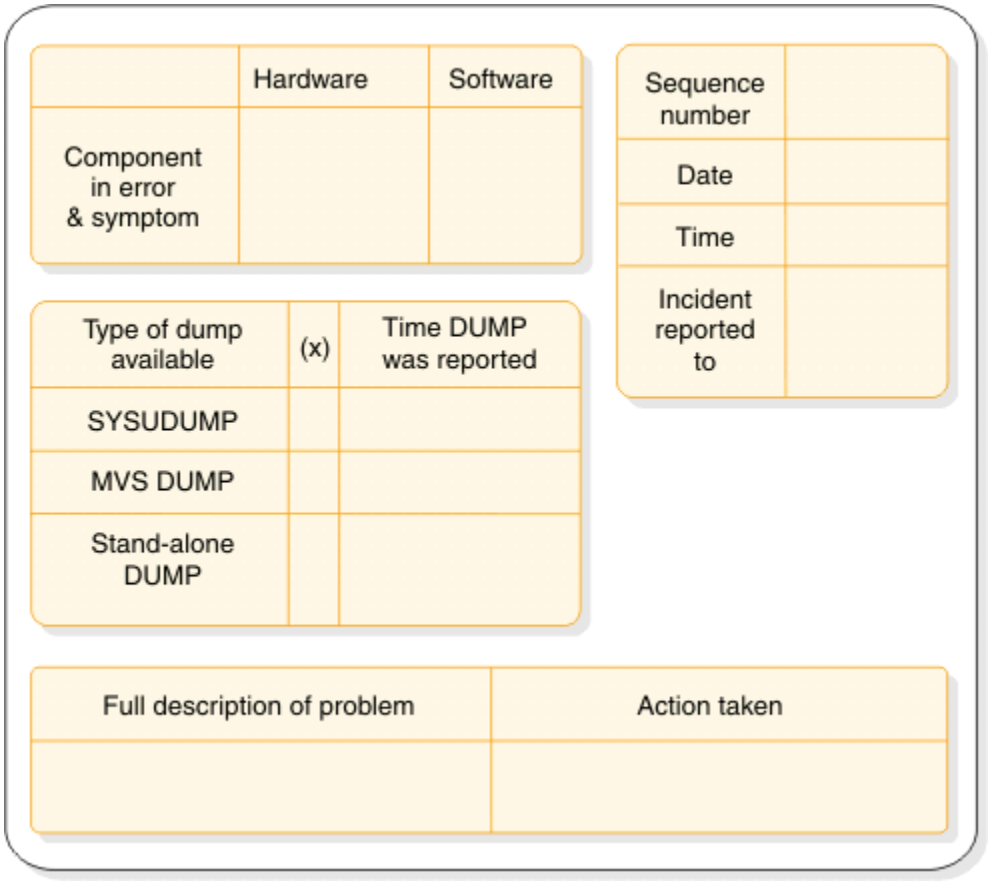


Figure 9. Sample form: incident report

Forms for recording startup, shutdown, and system log activity

You might want to design an online run sheet for recording startup, shutdown, and their relationship to system log activity.

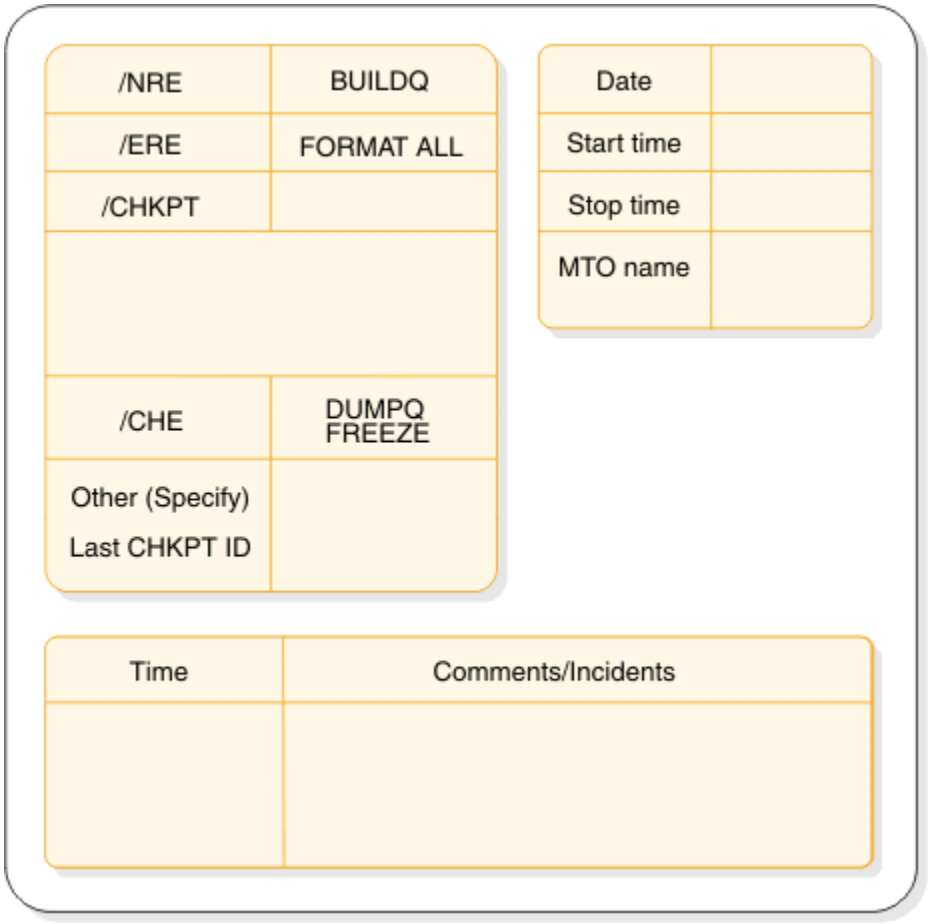


Figure 10. Sample form: online run sheet

The MTO fills in the online run sheet for each execution of the online system. When starting IMS, the MTO records (circles) the parameters used in the start command, along with the checkpoint ID used. When shutting down IMS normally, the MTO records (circles) the parameters used. When shutting down IMS abnormally, the MTO records the reason for the failure and fills in an incident report form. If the MTO shuts down IMS using a **/CHECKPOINT DUMPQ** command, that person records the closing checkpoint ID from the DFS994I message.

The MTO should use the bottom half of the form (comments/incidents) to record any unusual incidents. The MTO should note such things as: hardware errors on telecommunication lines, details about reassignment of LTERM names, and abending transactions. Finally, the MTO should add a cross reference to the incident report, where necessary.

DBRC records information about use of OLDSs in the RECON data set. After IMS archives an OLDS for an execution of the online system, the MTO can issue a `LIST.LOG` command to get information about the use of the OLDS. The MTO should file the hardcopy listing of this information with the online run sheet. This information provides an important means of communication between one shift of MTOs and the next. This information can also be used to assist in reconstruction of the RECON data set, if it becomes damaged.

The MTO should also file the complete hardcopy listing from the master terminal printer with the online run sheet.

OM audit trail

The IMS Operations Manager (OM) uses the z/OS system logger to provide an audit trail. IMS operator commands issued through OM, as well as their command response, are saved to the system logger. In addition, some unsolicited messages from OM, SCI, and IMS are also logged in the audit trail.

All input is logged before OM calls the OM Input user exit. Input is logged upon return from the OM Input user exit only if the exit modified the input. All output is logged before OM calls the OM Output user exit. Output is logged upon return from the OM output user exit only if the exit routine modified the output.

System logger

The system logger is a z/OS component that allows an application to log data from a sysplex environment. Data can be logged from one z/OS system or from multiple systems across the sysplex.

The z/OS system logger is a set of services that allows an application to write, browse, and delete log data. You can merge the log data from applications in a sysplex into a log stream, which is simply a collection of data in log blocks residing in the coupling facility; as the coupling facility fills, older data might be off-loaded to archival log stream data sets on DASD. Log blocks in the coupling facility can be backed up either in storage in each system or on DASD in staging data sets. When the log blocks in the coupling facility reach an installation-defined threshold value, they are off-loaded to DASD log data sets. Therefore, at any point in time, the log stream consists of records on the DASD log data sets (where data is hardened for longer term access) and the log blocks that currently reside in the coupling facility.

A system logger configuration includes the system logger address space in each system, the LOGR couple data set, a log stream structure in a coupling facility, DASD log data sets for off-loaded data from the coupling facility log stream, and optionally staging data sets for a backup copy of the log blocks resident in the log stream structure.

The system logger address space provides the application with services and connections to the coupling facility. An installation must:

1. Plan and predefine the coupling facility structures
2. Format and define a policy for the LOGR couple data set.
3. Define the DASD log data sets.
4. Optionally create staging data sets.

Related concepts

z/OS: Using system logger services

IBM Redbooks: System Logger

Planning availability of IMS service

In order to provide a framework for the operator control of the online IMS subsystem, you need to obtain overall requirements. You can then proceed gradually to fill in the details of specific requirements of various user groups.

About this task

Related concepts

“Planning the content of procedures” on page 163

Your procedures for the master terminal operator (MTO) will vary depending upon their knowledge of IMS operations and the application-program logic

Checklist for defining the production cycle

When defining the production cycle, obtain the overall production cycle requirement and the support requirements for databases and batch-program scheduling. Collecting the initial availability requirements ensures that you can plan your follow-up activities accordingly.

Take the following suggested steps to define the production cycle:

1. Obtain the overall production cycle requirement.

   For example, suppose the execution of an application program requires the online subsystem to be available from 0400 to 2000 hours, with no weekend requirements. Very often these requirements are made to fit the data processing organization's constraints. In this example, the operations group must be agreeable to the early-morning scheduling (which is probably required for transaction input from locations in different time zones).

2. Obtain support requirements for databases and batch-program scheduling.

   For example, a daily batch-report program might cause certain databases to be unavailable until 0800. A **/DBRECOVERY DB** or **UPDATE DB STOP(ACCESS)** command stops additional transaction or CCTL access and allows offline processing for the database named in the command. Another example is a priority transaction, scheduled for database integrity purposes, that preempts other transaction processing.

The following table shows a checklist of initial availability requirements you need to collect.

Table 23. Availability requirements checklist

Availability requirement	Follow-up activity
What are the available hours: Daily? Weekly?	Check if a regular weekly time. Check daily start and stop times. Allow for weekends, holidays, and preventive maintenance.
Which are services critical to the user?	Summarize the transactions, programs, and database needs. Summarize the CCTL's dependencies for programs (PSBs) and databases.
What is the target maximum down time?	Establish guidelines for duration.
Are there any planned interruptions?	Check requirements for normal restart. Determine how queues are to be handled.
Are there variations of nucleus?	Match nucleus suffix to system definition schedule.
Is there any special message region scheduling?	Develop IMSMSG procedure variation. Find conditions that prompt scheduling.
What are the BMP scheduling requirements?	Tailor IMSBATCH procedure for BMP. Find z/OS file requirements.
What databases are not always online?	List the databases, their conditions, and the reasons for their being offline.
What batch programs are to run during online schedule?	Find any database conflicts. Find the required database levels and restart instructions.
What is the timing of database reorganization and related jobs?	Find jobs that run during online execution. Obtain master schedule from DBA.
Do any application programs require exclusive processing?	Find the effect on other online processing.
Are there configuration variations or multiple IMS subsystems?	Develop procedure support: plan for MSC, data sharing, DL/I separate address space, DBCTL, DCCTL, APPC/IMS, OTMA, or ETO.
Are you using a data propagation manager?	Make sure both IMS and Db2 for z/OS are available on the same z/OS system. Make sure the program libraries that contain the data propagation user exit routines are available to all dependent IMS regions.

Devices available to the network

Document which devices are to be initially available, those that should always be active, and those that have restricted usage. The important attribute is whether the MTO needs to activate the connection or must make any alternative connections in case of terminal or line failures.

A second aspect of availability is to ensure proper control of the network. You need to document what terminals are connected and when.

The following table identifies some connection data you need to collect.

Table 24. Checklist for network control connection data

Connection data	Follow-up activity
Which terminals are associated with services critical to the user?	List the connection requirements. List the LTERMs.
Are VTAM terminals to be continuously connected?	List those nodes and their LTERMs.
Are VTAM terminals to be connected on demand?	Identify how each is controlled (MTO, VTAM operator, or remote terminal operator).
Are there planned reconfigurations?	Develop sets of terminals and identify how they should be controlled.
Are there alternate connection plans?	Identify if there are alternate pairings, or occasional usage.
Are intelligent stations as connected subsystems used?	Identify operations for connected subsystems or terminals, and monitoring requirements.
Are switched lines used?	Identify the users and give the LTERM names in the group.
Is there batched data entry or high-volume input?	Identify the pattern of batched devices.
Are there special output storage devices?	Identify use of printers or data devices.

Operations responsibilities for online control services

It is important to understand why an application program places restrictions on what commands you can issue. For example, online image copy is required at end-of-shift, but, because of the requirements of an IMS restart, no conflict in database level is permitted. This kind of restriction might require you to prevent certain programs from executing.

A third checklist identifies the use of support services and expected MTO control actions. The following table presents this information; your emphasis should be on the operational requirements.

Table 25. Checklist for online control services

Check requirement for:	Follow-up activity
Dynamic allocation of databases	Requires coding of DFSMDA macro and prior cataloging of databases.
Online database image copy	List databases and schedule timing. Is image copy restart allowed?
Database surveyor	When to be scheduled. Who is the contact for range specification?
Database recovery control	Arrange for DASD dump entries. How are RECON data sets to be controlled? How are JCL generation facilities to be used?

Table 25. Checklist for online control services (continued)

Check requirement for:	Follow-up activity
Simple checkpoint **/CHECKPOINT PURGE** **/CHECKPOINT DUMPQ** **/CHECKPOINT FREEZE** **/NRESTART**	What conditions require the checkpoint? Is SNAPQ required? Must all queues be emptied? Can queues be held? What conditions warrant use of this command? Is queue rebuilding required?
Online system log management	Determine archive procedures and whether WADS and OLDS are to be dynamically allocated.
Online change management	Determine the scope and timing of online changes and how to control processing.
Offline dump formatting management	Establish transfer, archive, and formatting procedures.
IBM 3850 Mass Storage System (MSS)	Obtain scenario of IBM 3850 usage.
Monitoring	How will IMSMON data set be handled? What requirements exist for traces or statistics?
Security Signon Transaction	Any override conditions? Any override conditions? What is update frequency and control procedure?

MTO abilities and operator responsibilities

The MTO's primary duty is to monitor the condition of the total IMS online environment, and to maintain normal service. Any alterations in committed service or recovery decisions are really part of a wider set of business responsibilities, not necessarily those of the MTO.

When you are defining the duties of the master terminal operator, you must decide what actions are part of the normal duties, and what decisions require consultation with other groups. From the end-user viewpoint, the MTO could represent the data-processing staff, as well as the application-program contact. You need to arrange ahead of time what kind of support can be provided by the MTO, and if there is any way of shielding the MTO from unnecessary interruption.

The master terminal operator has the following abilities and responsibilities:

- Responsibility for running IMS

  The MTO starts and shuts down the IMS subsystem, starts regions, and manages the system log.
- Knowledge of the ongoing status of the IMS subsystem

  The MTO continuously monitors processing and the detects any error situations.
- Control over contents of the system and the network

  The MTO can control the network, connect other IMS systems, and perform other prearranged tasks.
- Privileged commands

  In addition to the routine work, the MTO responds to error conditions, changes the scheduling algorithm, alters passwords, and reconfigures the system as necessary.

Normal operator actions (DB/DC or DCCTL)

The z/OS system operator needs to initialize the IMS control region, but you should plan to reduce the need for further involvement by the z/OS operator in IMS operations. In a busy system, responses

to messages or instructions sent by an application program to the z/OS operator can be erratic. Such requirements are better handled by output messages to the IMS master terminal.

You should ask the z/OS system operator to relay any IMS messages (ones beginning with any of the following prefixes: DFS, DSP, DXR, ELX, BPE, or CQS) to the IMS master terminal. You might also give the z/OS operator a list of the IMS job names. Because often the z/OS system console is physically separated from the IMS master terminal, you need a local telephone or a paging line to maintain communication between them.

You need to decide whether the z/OS operator or the IMS MTO should start IMS regions. If you want the z/OS operator to start them, one of the following IMS procedures can be used:

- IMSBATCH procedure to start batch message processing regions
- IMSRDR procedure to start message processing regions

The advantage of allowing the z/OS operator to start IMS regions is that the operator can monitor any abnormal termination patterns and I/O requests.

The following table shows the actions usually performed by the MTO and the commands usually reserved for the IMS master terminal.

Table 26. Master terminal operator actions usually performed

Activity	IMS command
Activate IMS (cold start)	/ERESTART COLDSYS
Start a message region	/START REGION IMSMSG1
Start communication lines	/START LINE ALL
Display message queues	/DISPLAY
Start another message region	/START REGION IMSMSG3
Prepare for VTAM communication	/START DC
Initiate static VTAM sessions	/OPNDST NODE ALL
Initiate dynamic VTAM sessions	/OPNDST NODE AOI
Send message to terminals	/BROADCAST
Shut down VTAM terminals and IMS	/CHECKPOINT FREEZE QUIESCE
Restart IMS (warm start)	/NRESTART

The IMS master terminal also works well as a display device. On the bottom of the screen are two input lines, with room for a warning message. The remainder of the screen is divided into two areas, each of 10 lines. The top portion shows the 10 most recent IMS messages; new messages overlay old ones, so the display cycles through the 10 lines. The lower portion is used to page through the output from the **/DISPLAY** command.

If you set the master terminal to use a 3270 format, some of the program function keys are predefined as follows:

PF Key
Command or function

1
/DISPLAY

2
/DISPLAY ACTIVE

3
/DISPLAY STATUS

4
/START LINE

5
/STOP LINE

6
/DISPLAY POOL

7
/BROADCAST LTERM ALL

11
Next message page (NEXTMSGP)

12
Copy function

Tip: Create a PF key panel that shows these settings and place it with the keyboard as a reminder for the operator.

Normal operator actions (DBCTL)

The DBCTL MTO can control the DBCTL environment from any z/OS system console. If your MTO operates from a different console (secondary system console) from the primary one where the z/OS operator sits, you must ensure a means of communication between the two operators.

DBCTL requests for tape or DASD mounts must be given to the system console operator. Some IMS messages (DFS prefix) originate from the CCTL region and thus appear on the primary system console; these messages should be relayed to the MTO in case some action needs to be taken. For example, when the CCTL tries to connect to a DBCTL environment and the DBCTL subsystem is not yet running, the IMS database resource adapter (DRA), which is part of the CCTL region, issues message DFS690A. In general, communication between the CCTL operator and the MTO is important.

As in the DB/DC or DCCTL environments, the z/OS operator can start IMS BMPs by using either the IMSRDR or the IMSBATCH procedure. Any activity of IMS dependent regions or utilities the z/OS operator controls should be recorded or communicated so that the MTO is aware of all IMS-related activity.

The following table shows the actions usually performed by the MTO and the commands usually reserved for the IMS DBCTL master terminal.

Table 27. Master terminal operator actions usually performed

Activity	IMS command
Activate IMS (cold start)	**/ERESTART COLDSYS**
Start a BMP	**/START REGION IMSBMP**
Display dependent regions	**/DISPLAY ACTIVE**
Shut down IMS	**/CHECKPOINT FREEZE**
Restart IMS (warm start)	**/NRESTART**

Resources controlled by the MTO

Resources controlled by the master terminal operator (MTO) are listed, including communications, scheduling, and integrity.

The following table summarizes the resources controlled by the MTO.

Table 28. Resources controlled by the MTO

Communications	Scheduling	Integrity
Telecommunication lines	Regions	Checkpoint

Table 28. Resources controlled by the MTO (continued)

Communications	Scheduling	Integrity
Physical terminals	Databases	Restart
Logical terminals	Transactions	Database copy
Spooled Output	Programs	Database availability
Users		System status

Connections for network operations

The major part of any detailed instructions for the MTO is the required connections or any alternate connections (in the case of terminal or line failures). A useful aid for the MTO is a chart showing the following:

- The network
- The device type
- PTERM and LTERM names
- Availability requirements
- A summary of the application usage

The chart should specify the contacts for problem resolution, whether they are user liaison, VTAM operator, or some other form of network control.

Session initiation and termination options

Because of their functional capability, many VTAM-supported terminal devices are shared by IMS and other users. You have several choices for handling connect and disconnect. You can have the remote operator or master terminal operator be responsible, or you can call upon intervention by a VTAM operator. The following table summarizes your choices.

Table 29. Session initiation and termination options

Control by	Initiate by	Terminate by
MTO	/OPNDST	/CLSDST
Remote operator	LOGON IMS	/RCLSDST
(VTAM,Defn)	Initialization list	-
VTAM operator	VARY command	VARY command

Secondary master terminal and auditing operational control

Some IMS commands are automatically copied to the terminal designated as SECONDARY in the system definition NAME macro. If you specify COPYLOG=ALL in the COMM macro during system definition, then when other terminals besides the master terminal enter any of these commands, copies of an entered command and its response are sent to the secondary master terminal.

To understand the actions taken by the MTO, you should obtain a permanent printed copy of the command sequences issued by the MTO. This function is termed *hardcopy logging*.

The MTO can use the **/SMCOPY MASTER OFF** command to override the logging of commands to the secondary master terminal. The MTO can also use this command to turn off logging of command entry and responses from other terminals. To resume logging to the secondary master terminal, the MTO can use the **/SMCOPY MASTER ON [TERMINAL ON]** command.

This more permanent record is valuable for several reason, but the principal one is that of accountability. You can also use the output to audit whether the operations command sequence and timing are providing committed service. The checklist in the following table identifies the types of activities and related commands, and shows some of the ways you can use the secondary master terminal output.

Table 30. Using secondary master terminal output to audit performance

Type of activity	Command	Output usage
Network control	**/OPNDST**	Check VTAM connection procedure.
	/CLSDST	Check VTAM disconnect procedure.
	/RCLSDST	Check expected disconnect.
	/START	Check network availability.
	/RSTART	Coordinate with terminal usage.
	/STOP	Investigate stopped resource.
	/IDLE	Investigate if there is lost or waiting output.
	/MSASSIGN	Check if planned MSC connect.
Device control	**/COMPT**	Check planned VTAM component switch.
	/RCOMPT	Check remote VTAM component allowed.
Resource control	**/ASSIGN**	Check that all uses fit operations plan.
	/START	Check region availability and queue control.
	/STOP	Investigate all events.
	/PSTOP	Investigate why input stopped.
	/PURGE	Investigate why service was restricted.
	/MODIFY	Audit the extent and timing of online changes.
	/CHANGE	Check size of MFS dynamic directory.
Security	**/MODIFY**	Set timing of authorization and password changes.
Recovery	**/DBRECOVERY**	Investigate database errors.
	/DBDUMP	Investigate database errors and subsequent processing.
	/DEQUEUE	Investigate lost message or output.
	/CHECKPOINT	Coordinate with reported errors.
	/SWITCH	Investigate reasons for unplanned takeover activity.
	/UNLOCK SYSTEM	Investigate takeover activity.
Monitoring	**/DISPLAY**	Incorporate in monitoring feedback. Check use for scheduling.
	/TRACE	Coordinate with trace output.

Table 30. Using secondary master terminal output to audit performance (continued)

Type of activity	Command	Output usage
HALDB OLR	**/INITIATE**	Initiate or resume HALDB Online Reorganization (OLR).
	/TERMINATE	Stop HALDB OLR.
	/UPDATE	Change the rate or the disposition flag of the DEL or NODEL data set of a HALDB OLR, or the disposition flag of REL or NOREL to release ownership of OLR at normal or abnormal shutdown of IMS.

Operator control of conversational transactions

The master terminal operator should be aware of the potential use of conversational transactions because their queue characteristics are somewhat different from those of nonconversational transactions. The number of conversational transactions that can be active at one time is 65535 per user or terminal. Also, the same individual transaction can be scheduled many times, but the processing patterns can be quite different among the set.

For example, one conversational transaction could involve an extended search of a database, while another, using multiple-choice menus, might retrieve a single item of data.

Another consideration for conversational transactions is whether the conversations are to be active until they are complete or whether they can be held while the end user is engaged in an offline activity.

Controlling conversational transactions

The following table summarizes the type of information that influences the content of your operational procedures when the online IMS subsystem includes conversational transactions.

Table 31. Operator information for conversational transactions

Transaction attribute	Information required	Procedure impact
Which transactions are conversational?	Summary of purpose and processing characteristics.	Multiple messages might need the message format buffer pool and message queues to be monitored.
Any dedicated terminals? What is the maximum number allowed to run?	Name LTERM and PTERM. Find maximum storage using DASD and expected loads.	Know impact of terminal problem. Advise remote terminal operator of restrictions and status.
Can conversations be held or released?	If so, how long? Is operator allowed to terminate?	MTO awareness of status. Use of **/PSTOP** and **/EXIT** commands. Use of **/MODIFY** warnings. Use of the **ENDCONV=** parameter in the DFSOTMA descriptor of the DFSYDT*x* member of the IMS PROCLIB data set.
What recovery actions are to be taken?	At restart. Are abnormal termination exit routines used?	Notify remote terminal operator and other recovery personnel of output loss.

Remote terminal operator control for conversational transactions

A remote terminal operator can use a **/FORMAT** command to begin a conversation. This command formats an MFS-supported terminal screen. Or, the terminal operator can enter the initial transaction. To terminate the conversation before the normal program end, the terminal operator can use the **/EXIT** command.

The operator can suspend the conversation (if the operator needs to leave the terminal area) by using the **/HOLD** command. In response, IMS issues message DFS999I, which displays the identification number needed to resume the conversation with the transaction program. The terminal is then available for other transaction activity.

On return to the terminal, the terminal operator can issue the **/RELEASE CONVERSATION** *nnnn* command to resume the conversation, where *nnnn* is the conversation ID from the DFS999I message. As a convenience, IMS re-sends the last output message to the terminal. Or the operator can issue the **/EXIT CONVERSATION** *nnnn* command to terminate the conversation.

If the MFS formatting or the output content of the screen is lost during a conversation, enter the **/HOLD** command followed by the **/RELEASE** command. These commands resend the first physical page of the current message and restore the format.

MTO control for conversational transactions

The master terminal operator can obtain a status of all conversations, held or active. The following output from the **/DISPLAY CONVERSATION** command displays the terminal address, the conversation identification number, and status and type of all conversations:

```
TERMINAL USER    ID STATUS
11-   2          0001 HELD
 4-   2          0002 ACTIVE,SCHEDULED
 4-   1          0011 HELD
VT03             0012 ACTIVE
```

The MTO can terminate a conversation. For example, the **/EXIT CONVERSATION 0001 LINE 11 PTERM 2** command terminates the first conversation shown in the command output above. A **/START LINE**, **/START NODE**, or **/START USER** command also terminates the conversation, unless a program is already scheduled to process a portion of the conversation exchange.

Recommendation: Before terminating a conversation, the MTO should broadcast a warning message to that terminal.

Ending idle OTMA conversational transactions using the ENDCONV= parameter of the DFSYDT*x* member

A timeout value for idle OTMA conversational transactions can be specified in the **ENDCONV=** parameter, which is in the DFSOTMA descriptor of the DFSYDT*x* member of the IMS PROCLIB data set. If the conversational transaction remains idle for the specified period after the prior iteration of the conversational transaction completes, OTMA ends the transaction. OTMA also removes from IMS storage the resources that were allocated to the idle conversational transaction.

Exit routines for abnormal termination of conversations

When a conversation terminates abnormally, a termination exit routine named DFSCONE0 can gain control. Its use might be to initiate cleanup activity, or simply to notify the input terminal of the canceled conversation. If the exit routine determines that more elaborate correction is required, it can schedule a transaction. This transaction can include the latest SPA, or a modified version of it, as a message segment. IMS provides a default exit routine (DFSCONE0) which schedules a transaction that includes the old SPA.

You can use the conversational termination exit routine to determine the effect of operator interruption.

Related reference

Conversational Abnormal Termination exit routine (DFSCONE0) (Exit Routines)

DFSOTMA descriptor syntax and parameters (System Definition)

MTO and tracing operations

The MTO or an authorized remote terminal operator can initiate several kinds of type-1 trace activity with the **/TRACE** command. Even though you might use the IMS Monitor fairly regularly, you should regard all tracing operations as "special requests".

Because trace or monitoring actions affect the performance of the online IMS system, they are not part of normal operations. The validity of the trace depends upon the processing taking place. Also, the operator can control the trace interval with the **/TRACE** command, so the operator must be aware of the trace timing.

Some types of trace requests do not require the person requesting the trace to be present during normal operations. The request should state the approximate duration of the trace and the type of monitoring. The main section should specify the initial start and termination conditions, as well as any monitoring constraints to be applied. Other sections should prompt for other details, such as queue levels, transactions to be active, or the time of day.

The types of tracing that can be requested are summarized in the following table.

Table 32. Trace and monitor options

Type of trace	/TRACE parameters	Operational effect
IMS Monitor	SET MON opt	Initiates IMS Monitor trace.
	SET OFF MON	Terminates the trace.
	opt=ALL	Traces all events.
	opt=APDB	Traces activity between application programs and databases.
	opt=APMQ	Traces activity between application programs and message queues.
	opt=LA	Traces line and logical link activity.
	opt=SCHD	Traces scheduling and termination events.
DL/I Image Capture	SET PSB name opt	Traces all DL/I calls for the named PSB.
	SET OFF PSB name	Terminates the trace.
	opt=COMP	Generates PCB and data-compare statement images.
	opt=NOCOMP	Generates no compare statement images. Default.

Table 32. Trace and monitor options (continued)

Type of trace	/TRACE parameters	Operational effect
	SET PI OPTION opt	Traces program isolation activity, including ENQ, DEQ, and DL/I calls. Trace data is written to system log.
	SET PI	Traces PI activity. Trace data is kept in storage, and not written to system log.
	SET OFF PI	Terminates the trace.
	opt=LOG	Writes trace data to system log. Default.
	SET OFF PI OPTION LOG	Continues trace, but does not write buffers to system log.
	opt=TIME	Adds elapsed time for each ENQ and DEQ call when IMS has to wait.
	SET OFF PI OPTION TIME	Continues trace, but only records time of day.
	opt=ALL	Equivalent to opt=LOG TIME.
Line, node, and PTERM tracing	SET LINE nn opt	Traces line activity.
	SET OFF LINE nn	Terminates the line trace.
	SET NODE nn opt	Traces node activity.
	SET OFF NODE nn	Terminates the node trace.
	opt=LEVEL n	Sets detail level (1-4) for trace.
	opt=MODULE mod	Traces modules for IMS communications manager.
	mod=DDM	Traces device-dependent modules.
	mod=MFS	Traces MFS modules.
	mod=ALL	Traces both DDM and MFS modules.

Table 32. Trace and monitor options (continued)

Type of trace	/TRACE parameters	Operational effect
Type-1 Trace Tables	SET TABLE opt OPTION LOG	Initiates tracing into specified trace tables. IMS writes trace tables to log before reusing them.
	SET TABLE opt OPTION NOLOG	Initiates tracing into specified trace tables. IMS does not write trace tables to log. Default.
	SET OFF TABLE Opt	Terminates trace.
	opt=ALL	Specifies that IMS enable or disable all traces. Default.
	opt=DISP	Traces dispatching events.
	opt=DL/I	Traces DL/I calls.
	opt=DLOG	Traces log activity.
	opt=FAST	Traces Fast Path activity.
	opt=IDC0	Traces errors in modules DFSCNXA0 and DFSIDC00.
	opt=LATC	Traces latching activity.
	opt=LOCK	Traces locking and PI activity.
	opt=LUMI	Traces LU 6.2 and APPC activity.
	opt=OTMT	Traces OTMA activity.
	opt=RETR	Traces DL/I retrieve calls.
	opt=SCHD	Traces scheduling events.
	opt=STRG	Traces storage manager activity.
	opt=SUBS	Traces external subsystem events.
Link Tracing for MSC	SET LINK nn opt	Traces logical link.
	SET OFF LINK nn	Terminates link trace.
	opt=LEVEL n	Sets detail level (1-4) for trace.
	opt=MODULE mod	Traces modules for IMS communications manager.
	mod=DDM	Traces device-dependent modules.
	mod=MFS	Traces MFS modules.
	mod=ALL	Traces both DDM and MFS modules.

IMS can trace several different types of activity concurrently. The IMS Monitor writes its output to a data set specified on the IMSMON DD statement or dynamically allocated. IMS writes other traces to type X'67' log records.

Related reference

/TRACE commands (Commands)

Offline dump formatting operations

IMS operators need to be aware that their intervention might be required if a dump is requested at a time when the dump data sets are unavailable or when some other task in the z/OS system is currently taking an SDUMP.

In these situations, IMS issues message DFS3906 or DFS3907. In addition, CCTL operators must be aware that the CCTL can issue these messages when the IMS database resource adapter (DRA) is trying to generate an SDUMP.

Designing operating procedures

You can document operating procedures using flowcharts or a narrative style.

Operating procedure design using flowcharts and other graphic techniques

Flowcharts have several obvious advantages over narrative presentations of procedures. For example, with a flowchart it is easy to get an overall picture of what needs to be done. Another advantage is that when text accompanies each flowchart, you can skip reading any text associated with a step that you already understand.

If the flowchart, for example, says to start IMS and this step is understood, the reader does not need to read that "the system operator enters the **/NRE** command from the system console...".

When you use flowcharts or other graphics to document operating procedures, you generally accompany them with an expanded textual description of the various steps in the flowchart. The main disadvantage to using flowcharts is that they are often difficult to maintain. Small changes to an operating procedure could require you to redesign an entire flowchart.

You might not need to use flowcharts for the IMS MTO. However, it is probably the preferred technique when designing procedures for a recovery specialist or others who need an overview of how something works or what actions are to be performed.

Operating procedure design using narrative

The advantage to using a narrative style is that it is relatively easy to maintain. There are no graphics to redraw, and changes to the procedures seldom require much redesign.

The disadvantage is that the overall operation being performed is not always obvious. In addition, if the users of the procedure suspect that they have taken a wrong branch in the procedure, it is difficult to backtrack through the steps.

There are many ways to document procedures using a narrative style, rather than graphics. For example, you could use a numbered list, where the reader starts at number 1, compares the situation with the information in number 1, then continues to number 2 or maybe jumps to number 15 depending on the decision made in number 1. This technique is essentially a narrative flowchart. As another example, you could write scenarios, wherein the reader picks a scenario that matches their situation, and follows all the steps in the scenario.

IMS-to-IMS TCP/IP connection operations

The communication path of an IMS-to-IMS TCP/IP connection passes through multiple IMS components: the IMS control region, the Structured Call Interface (SCI) of the Common Service Layer (CSL), IMS Connect, and either the IMS Multiple Systems Coupling (MSC) component or the IMS Open Transaction Manager Access (OTMA) component.

Operational tasks for an IMS-to-IMS TCP/IP connection in one segment of an IMS-to-IMS TCP/IP connection impacts the connection in the other segments. Also, general operational tasks in the IMS system are also likely to impact any IMS-to-IMS TCP/IP connections.

Related concepts

“MSC operations” on page 197

In a non-sysplex environment, each IMS system in a Multiple Systems Coupling (MSC) configuration is operationally an independent unit. Each IMS system exclusively owns its own communication resources, and is controlled by its own master terminal.

Viewing configuration and status information for IMS-to-IMS TCP/IP connections in IMS Connect

You can view configuration and status information for IMS-to-IMS TCP/IP connections by issuing IMS Connect commands.

About this task

The connection information you can view is generally the same regardless of whether the connection is used for Multiple Systems Coupling (MSC) or Open Transaction Manager (OTMA). However, if you use MSC, you can also display information about the MSC link.

Viewing MSC connection information for IMS-to-IMS TCP/IP communications

You can view configuration and status information for IMS-to-IMS TCP/IP connections that are used for MSC by issuing IMS Connect commands.

About this task

Each MSC link requires two socket connections: a send socket connection and a receive socket connection. You display information about the send socket connections by querying RMTIMSCON connection information. You display information about the receive socket connection by querying the port information.

You can also display information about the definition and status of an MSC logical link by querying the MSC configuration information in IMS Connect.

Procedure

- To display connection information on the sending side of a connection that is used for MSC, issue any one of the following IMS Connect commands that display the RMTIMSCON connection information:
  - In the IMS type-2 format, **QUERY IMSCON** TYPE(RMTIMSCON)
  - In the WTOR format, **VIEWRMT**
  - In the z/OS MODIFY format, **QUERY RMTIMSCON**
- To display connection information on the receiving side of a connection that is used for MSC, issue any one of the following IMS Connect commands that display port information:
  - In the IMS type-2 format, **QUERY IMSCON** TYPE(PORT)
  - In the WTOR format, **VIEWPORT**
  - In the z/OS MODIFY format, **QUERY PORT**
- To display link information on either side of a connection that is used for MSC, issue any one of the following IMS Connect commands that display link information:
  - In the IMS type-2 format, **QUERY IMSCON** TYPE(MSC)
  - In the WTOR format, **VIEWMSC**
  - In the z/OS MODIFY format, **QUERY MSC**

Example: Viewing an MSC TCP/IP connection at the local and remote IMS Connect instances

In the following example, the VIEWRMT command is issued at a local IMS Connect instance for a RMTIMSCON connection, ICON2, that is used by MSC to send messages to the remote IMS installation.

```
nnVIEWRMT ICON2
```

The example output shows that two send socket connections, MSCBB435 and MSC84CF7, are in CONN status on the RMTIMSCON connection ICON2. The SENDCLNT IDs are auto-generated by the local IMS Connect instance when establishing the send socket connections to remote IMS Connect instance. In the example, PORT=5555 shows that the local IMS Connect expects the remote IMS Connect to be listening on port 5555. The presence of the LCLPLKID values indicate that these are MSC connections and identify the local MSC physical link that they are using.

```
HWSC0001I    RMTIMSCON=ICON2     STATUS=ACTIVE
HWSC0001I      IP-ADDRESS=009.030.221.055  PORT=5555
HWSC0001I      HOSTNAME=ECSER14.VMEC.SVL.IBM.COM
HWSC0001I      AUTOCONN=N  PERSISTENT=Y
HWSC0001I      IDLETO=0
HWSC0001I      RESVSOC=2   NUMSOC=2
HWSC0001I      SENDCLNT LCLPLKID STATUS       SECOND SENDPORT
HWSC0001I      MSCBB435 MSC12    CONN              6 1028
HWSC0001I      MSC84CF7 MSC12    CONN             56 1027
HWSC0001I      TOTAL SENDCLNTS=2 RECV=0 CONN=2 XMIT=0 OTHER=0
```

In the following example, the VIEWPORT command is issued at the local IMS Connect instance for port 9999.

```
nnVIEWPORT 9999
```

The example output shows the receive socket connections, MSCC73E0 and MSC0EBB0, that are in RECV state on port 9999 at the local IMS Connect instance. Again, the presence of the LCLPLKID values indicate that these connections are being used for MSC, as well as the local MSC physical link that they are using.

```
HWSC0001I    PORT=9999     STATUS=ACTIVE     KEEPAV=0 NUMSOC=3      EDIT=       TIMEOUT=0
HWSC0001I      CLIENTID LCLPLKID  STATUS      SECOND CLNTPORT IP-ADDRESS
HWSC0001I      MSCC73E0 MSC12     RECV             6 1026     0:0:0:0:0:FFFF:91E:DD37
HWSC0001I      MSC0EBB0 MSC12     RECV            56 1025     0:0:0:0:0:FFFF:91E:DD37
HWSC0001I      TOTAL CLIENTS=2  RECV=2 READ=0 CONN=0 XMIT=0 OTHER=0
```

The following examples for a remote IMS Connect instance correspond to the preceding examples for a local IMS Connect instance.

In the following example, the VIEWPORT command is issued for port 5555 at the remote IMS Connect instance.

```
nnVIEWPORT 5555
```

The output for the VIEWPORT command shows two receive socket connections, MSCBB435 and MSC84CF7, on port 5555. These CLIENTID values match the SENDCLNT values shown in the preceding example of the VIEWRMT command at the local IMS Connect instance.

```
HWSC0001I    PORT=5555     STATUS=ACTIVE     KEEPAV=0 NUMSOC=3      EDIT=       TIMEOUT=0
HWSC0001I      CLIENTID LCLPLKID  STATUS      SECOND CLNTPORT IP-ADDRESS
HWSC0001I      MSCBB435 MSC21     RECV            18 1028     0:0:0:0:0:FFFF:91E:7326
HWSC0001I      MSC84CF7 MSC21     RECV            69 1027     0:0:0:0:0:FFFF:91E:7326
HWSC0001I      TOTAL CLIENTS=2  RECV=2 READ=0 CONN=0 XMIT=0 OTHER=0
```

In the following example, the VIEWRMT command is issued at the remote IMS Connect instance for RMTIMSCON connection, ICON1.

```
nnVIEWRMT ICON1
```

The example output shows that two send socket connections, MSCC73E0 and MSC0EBB0 under SENDCLNT, are in CONN status on the RMTIMSCON connection ICON1. These SENDCLNT values match the CLIENTID values shown in the preceding example of the VIEWPORT command at the local IMS

Connect instance. In the example, PORT=9999 shows that the remote IMS Connect expects the local IMS Connect to be listening on port 9999. The presence of the LCLPLKID values indicate that these are MSC connections and identify the local MSC physical link that they are using.

```
HWSC0001I    RMTIMSCON=ICON1     STATUS=ACTIVE
HWSC0001I      IP-ADDRESS=009.030.115.038  PORT=9999
HWSC0001I      HOSTNAME=ECSER13.VMEC.SVL.IBM.COM
HWSC0001I      AUTOCONN=N  PERSISTENT=Y
HWSC0001I      IDLETO=0
HWSC0001I      RESVSOC=2   NUMSOC=2
HWSC0001I      SENDCLNT LCLPLKID STATUS       SECOND SENDPORT
HWSC0001I      MSCC73E0 MSC21    CONN             18 1026
HWSC0001I      MSC0EBB0 MSC21    CONN             69 1025
HWSC0001I      TOTAL SENDCLNTS=2 RECV=0 CONN=2 XMIT=0 OTHER=0
```

Related concepts

"Displaying information about an MSC network" on page 206
There are two commands that you can use to view information about an MSC network: the type-1 command **/DISPLAY** and the type-2 command **QUERY**.

Related tasks

"Viewing MSC link information in IMS Connect" on page 186
When an IMS Connect to IMS Connect connection is used for MSC, you can view configuration and status information about the MSC links by issuing an IMS Connect command.

Related reference

QUERY IMSCON commands (Commands)
VIEWPORT command (Commands)
VIEWRMT command (Commands)
VIEWMSC command (Commands)
IMS Connect QUERY MSC command (Commands)
IMS Connect QUERY PORT command (Commands)
IMS Connect QUERY RMTIMSCON command (Commands)

Viewing MSC link information in IMS Connect

When an IMS Connect to IMS Connect connection is used for MSC, you can view configuration and status information about the MSC links by issuing an IMS Connect command.

About this task

You can display configuration and status information for MSC links by the MSC physical link that are defined to IMS Connect or by logical link name.

Procedure

- To display connection information about the configuration and status of MSC physical links, issue any one of the following IMS Connect commands:
  - In the IMS type-2 format, **QUERY IMSCON** TYPE(MSC). This command also displays some information about the MSC links.
  - In the WTOR format, **VIEWMSC**
  - In the z/OS MODIFY format, **QUERY MSC**
- To display status information for MSC logical links, issue the following command:
  - In the IMS type-2 format, **QUERY IMSCON** TYPE(LINK)

Related concepts

"Displaying information about an MSC network" on page 206

There are two commands that you can use to view information about an MSC network: the type-1 command **/DISPLAY** and the type-2 command **QUERY**.

Related tasks

"Viewing MSC connection information for IMS-to-IMS TCP/IP communications" on page 184
You can view configuration and status information for IMS-to-IMS TCP/IP connections that are used for MSC by issuing IMS Connect commands.

Related reference

QUERY IMSCON commands (Commands)
VIEWMSC command (Commands)
IMS Connect QUERY MSC command (Commands)

Viewing OTMA connection information for IMS-to-IMS TCP/IP communications

You can view configuration and status information for IMS-to-IMS TCP/IP connections that are used for OTMA by issuing IMS Connect commands.

About this task

The information that you can view for an IMS-to-IMS TCP/IP connection that is used for OTMA is different depending on whether you are viewing the information on the sending side of a connection or the receiving side of a connection. For this reason, the commands you use to display the connection information are also different on the sending and receiving sides of the connection.

Procedure

- To display connection information on the sending side of a connection that is used for OTMA, issue any one of the following IMS Connect commands that display the RMTIMSCON connection information:
  - In the IMS type-2 format, **QUERY IMSCON** TYPE(RMTIMSCON)
  - In the WTOR format, **VIEWRMT**
  - In the z/OS MODIFY format, **QUERY RMTIMSCON**
- To display connection information on the receiving side of a connection that is used for OTMA, issue any one of the following IMS Connect commands that display port information:
  - In the IMS type-2 format, **QUERY IMSCON** TYPE(PORT)
  - In the WTOR format, **VIEWPORT**
  - In the z/OS MODIFY format, **QUERY PORT**

Example: IMS Connect to IMS Connect connections for OTMA

In the following example, the VIEWRMT command is issued at a sending IMS Connect instance for a RMTIMSCON connection, ICON2B.

```
nnVIEWRMT ICON2B
```

The example output shows that there is one active send client socket connection, OTM924FA, on the RMTIMSCON connection ICON2B. The SENDCLNT ID is auto-generated by the sending IMS Connect instance when establishing the connection with the receiving IMS Connect instance. In the example, PORT=5555 shows that the sending IMS Connect expects the receiving IMS Connect to be listening on port 5555.

```
HWSC0001I    RMTIMSCON=ICON2B    STATUS=ACTIVE
HWSC0001I      IP-ADDRESS=009.030.221.055  PORT=5555
HWSC0001I      HOSTNAME=ECSER14.VMEC.SVL.IBM.COM
HWSC0001I      AUTOCONN=N  PERSISTENT=Y
HWSC0001I      IDLETO=3000
HWSC0001I      RESVSOC=4   NUMSOC=1
```

```
HWSC0001I      SENDCLNT USERID   STATUS       SECOND SENDPORT
HWSC0001I      OTM924FA APOL1    CONN           5941 1026
HWSC0001I      TOTAL SENDCLNTS=1 RECV=0 CONN=1 XMIT=0 OTHER=0
```

In the following example, which corresponds to the preceding example, the VIEWPORT command is issued at a receiving IMS Connect instance for port 5555.

```
nnVIEWPORT 5555
```

The example output shows that there is one receive socket connection, OTM924FA, in RECV state on port 5555. The CLIENTID OTM924FA of the socket connection matches the SENDCLNT ID shown in the example command output for the sending IMS Connect instance.

```
HWSC0001I    PORT=5555     STATUS=ACTIVE     KEEPAV=0 NUMSOC=2       EDIT=        TIMEOUT=0
HWSC0001I      CLIENTID USERID TRANCODE DATASTORE STATUS  SECOND CLNTPORT IP-
ADDRESS               APSB-TOKEN
HWSC0001I      OTM924FA APOL1  APOL11   IMS2      RECV        11 1026
0:0:0:0:0:FFFF:91E:7326
HWSC0001I      TOTAL CLIENTS=1  RECV=1 READ=0 CONN=0 XMIT=0 OTHER=0
```

Related reference

QUERY IMSCON commands (Commands)

VIEWPORT command (Commands)

VIEWRMT command (Commands)

IMS Connect QUERY PORT command (Commands)

IMS Connect QUERY RMTIMSCON command (Commands)

Stopping a connection to a remote IMS Connect instance for IMS-to-IMS TCP/IP communications

You can stop an IMS-to-IMS TCP/IP connection to a remote IMS Connect instance by issuing an IMS Connect command from the local IMS Connect.

About this task

Depending on whether the connection is used for Multiple Systems Coupling (MSC) or Open Transaction Manager (OTMA), the implications of stopping an IMS-to-IMS TCP/IP connection to a remote IMS Connect instance are different.

To stop an IMS-to-IMS TCP/IP connection to a remote IMS Connect instance:

Procedure

Issue any one of the following IMS Connect commands to the local IMS Connect instance:

- In the IMS type-2 format, **UPDATE IMSCON** TYPE(RMTIMSCON) NAME(*rmtimscon_name*) STOP(COMM)
- In the WTOR format, **STOPRMT**
- In the z/OS MODIFY format, **UPDATE RMTIMSCON** STOP(COMM)

Results

IMS Connect issues message HWST3505I after the connection is terminated.

If you are stopping a connection to a remote IMS Connect instance that is used for MSC, the following actions occur when the connection is stopped:

- The local IMS Connect issues an informational message to the console and notifies MSC that the connection was stopped.
- IMS stops the MSC logical links that use the IMS-to-IMS TCP/IP connection.

- The local IMS Connect disconnects the TCP/IP sockets associated with the IMS-to-IMS TCP/IP connection.
- If the local IMS Connect receives any messages from MSC for the stopped connection, IMS Connect rejects the message and returns an error message to MSC.

If you are stopping a connection to a remote IMS Connect instance that is used for OTMA, the following actions occur when the connection is stopped:

- The local IMS Connect disconnects the TCP/IP sockets associated with the IMS-to-IMS TCP/IP connection.
- If an acknowledgement (ACK) response is pending on any socket that uses the connection, the local IMS Connect returns a negative acknowledgement (NAK) to OTMA in the local IMS system. IMS Connect issues message HWST3570E.
- If the local IMS Connect receives any OTMA messages from the local IMS system for the stopped connection, IMS Connect returns a NAK to OTMA with sense code 002A/0008 and issues message HWST3575W. OTMA leaves the message at the front of the output queue.

Related reference

IMS Connect commands (Commands)

UPDATE IMSCON commands (Commands)

Restarting a connection to a remote IMS Connect instance for IMS-to-IMS TCP/IP communications

You can restart a stopped IMS-to-IMS TCP/IP connection in a local IMS Connect instance by issuing an online IMS Connect command.

About this task

Depending on whether the connection is used for Multiple Systems Coupling (MSC) or Open Transaction Manager (OTMA), the implications of restarting an IMS-to-IMS TCP/IP connection from IMS Connect are different.

If you are restarting a connection that is used for MSC, MSC communication does not resume until the MSC physical link is also started in the local IMS Connect instance.

If you are restarting a connection that is used for OTMA, the following actions occur when the connection is started:

- If the IMS-to-IMS TCP/IP connection is defined locally with AUTOCONN=Y, the local IMS Connect establishes socket connections with the remote IMS Connect instance. The number of socket connections opened is determined by the RESVSOC parameter in the RMTIMSCON configuration statement.
- The local IMS Connect notifies the local OTMA that the connection has been restarted and OTMA resumes sending messages on the connection.

To restart an IMS-to-IMS TCP/IP connection from the local IMS Connect instance:

Procedure

1. Issue any one of the following IMS Connect commands:
   - In the IMS type-2 format, **UPDATE IMSCON** TYPE(RMTIMSCON) NAME(*rmtimscon_name*) START(COMM)
   - In the WTOR format, **STARTRMT**
   - In the z/OS MODIFY format, **UPDATE RMTIMSCON** START(COMM)
2. If the connection is used for MSC, start the MSC physical link in the local IMS Connect by issuing any one of the following IMS Connect commands:

- In the IMS type-2 format, **UPDATE IMSCON** TYPE(MSC) NAME(*lclPlkid*) START(COMM)
- In the WTOR format, **STARTMSC** *lclplk_id*
- In the z/OS MODIFY format, **UPDATE MSC** NAME(*lclPlkid*) START(COMM)

Results

IMS Connect issues message HWS3500I after the connection is restarted.

Related reference

IMS Connect commands (Commands)

UPDATE IMSCON commands (Commands)

Stopping IMS Connect send client sockets on IMS-to-IMS TCP/IP connections

You can stop IMS Connect send client socket connections on an IMS-to-IMS TCP/IP connection in a local IMS Connect by issuing an IMS Connect command, such as the WTOR format command STOPSCLN.

About this task

Usually, the STOPSCLN command and the equivalent IMS type-2 and z/OS MODIFY commands are used only to stop send client socket connections that are used for IMS Open Transaction Manager Access (OTMA).

Recommendation: Do not stop IMS Connect send client socket connections if they are used for Multiple Systems Coupling (MSC). If you need to clean up sockets that are used for MSC, use the STOPLINK command or the equivalent IMS type-2 and z/OS MODIFY command.

In IMS Connect, the send client socket connection is represented by a client ID that is automatically generated by IMS Connect when the socket connection is established.

When you stop a send client socket that is connected but idle, as indicated by a CONN status in IMS Connect, the following actions occur:

- IMS Connect disconnects the send client socket and cleans up the associated control blocks.
- IMS Connect notifies the remote IMS Connect and the remote IMS Connect disconnects the related socket and cleans up its control blocks.

When you stop a send client that is waiting for an acknowledgement from the remote IMS Connect, as indicated by a RECV status in IMS Connect, the following actions occur:

- IMS Connect disconnects the send client socket and cleans up the associated control blocks.
- IMS Connect issues a negative acknowledgement (NAK) to OTMA that direct OTMA to reroute unacknowledged message to the IMS Connect dead letter queue HWS$DLQ.
- IMS Connect issues message HWST3570E to the console.
- When the remote IMS Connect attempts to return the acknowledgement, the remote IMS Connect receives a socket error and disconnects the related socket and cleans up its control blocks.

To stop an IMS Connect send client socket of an IMS-to-IMS TCP/IP connection:

Procedure

Issue any one of the following IMS Connect commands in the local IMS Connect instance:

- In the IMS type-2 format, **UPDATE IMSCON** TYPE(SENDCLNT) NAME(*sendclient_name*) RMTIMSCON(*rmtimscon_name*) STOP(COMM)
- In the WTOR format, **STOPSCLN**
- In the z/OS MODIFY format, **DELETE RMTIMSCON** NAME (*rmtimscon*) SENDCLNT(*clientid*)

Results

IMS Connect issues message HWS3525I after the connection is terminated.

Related reference

IMS Connect commands (Commands)

UPDATE IMSCON commands (Commands)

Stopping IMS Connect communication with an IMSplex when MSC TCP/IP links are supported

You can stop communications between a local IMS Connect instance and a local IMSplex by issuing an IMS Connect command, such as the WTOR format command STOPIP.

About this task

Stopping IMSplex communications breaks all MSC TCP/IP physical links that are supported by the local IMS Connect in the local IMSplex.

When a STOPIP or similar command is issued, the following actions occur for MSC TCP/IP links:

- The local IMS Connect disconnects the parallel send and receive socket that support the physical link and sends a shutdown directive to MSC in the local IMS system.
- When MSC receives the shutdown directive, MSC cleans up the physical link and issues message DFS3176E.
- IMS Connect changes the status of the MSC TCP/IP physical link to DISCONNECTED.

To stop communication between a local IMS Connect instance and a local IMSplex from the local IMS Connect instance issue any one of the following IMS Connect commands:

Procedure

- In the IMS type-2 format, **UPDATE IMSCON** TYPE(IMSPLEX) NAME(*imsplex_name*) STOP(COMM)
- In the WTOR format, **STOPIP** *imsplex_id*
- In the z/OS MODIFY format, **UPDATE IMSPLEX** NAME (*imsplex_name*) STOP(COMM)

Related reference

IMS Connect commands (Commands)

UPDATE IMSCON commands (Commands)

Starting IMS Connect communication with an IMSplex when MSC TCP/IP links are supported

You can start communications between a local IMS Connect and a local IMSplex by issuing an IMS Connect command, such as the WTOR format command STARTIP.

About this task

When a STARTIP or similar command is issued, communication resumes between IMS Connect and the IMSplex.

IMS Connect changes the status of the MSC TCP/IP links that it supports in the IMSplex to ACTIVE.

To start communication between a local IMS Connect and a local IMSplex from the local IMS Connect instance issue any one of the following IMS Connect commands:

Procedure

- In the IMS type-2 format, **UPDATE IMSCON** TYPE(IMSPLEX) NAME(*imsplex_name*) START(COMM)
- In the WTOR format, **STARTIP** *imsplex_id*

- In the z/OS MODIFY format, **UPDATE IMSPLEX** NAME (*imsplex_name*) START(COMM)

Related reference
IMS Connect commands (Commands)
UPDATE IMSCON commands (Commands)

Cleaning up MSC logical link resources in IMS Connect

If an MSC logical link has terminated in IMS, but the IMS Connect resources associated with the logical link did not clean up automatically, you can clean up the resources by using an IMS Connect command, such as the WTOR format command STOPLINK.

About this task

Recommendation: Under normal circumstances, use the IMS command /PSTOP to terminate MSC logical links. Stop MSC links in IMS Connect only when the IMS Connect resources associated with an MSC logical link that has already terminated have failed to clean up correctly.

A logical link on one MSC physical link can have the same name as a logical link on another physical link. To ensure that a logical link with a duplicate name is not stopped on another physical link, include the ID of the physical link that the logical link is assigned to on the command when you stop a logical link. The ID of the physical link can be found on the on the LCLPLKID parameter of the MSC configuration statement.

When you clean up MSC logical links in a local IMS Connect instance by using a command like STOPLINK, IMS Connect takes the following actions:

- Stops any communication on the MSC logical link
- Notifies the local IMS that communication has stopped on the logical link
- Deletes the control blocks associated with the logical link and frees the associated storage
- Issues message HWSF3310I

Procedure

To stop an MSC logical link and clean up its associated resources in a local instance of IMS Connect issue any one of the following IMS Connect commands in the local IMS Connect instance:

- In the IMS type-2 format, **UPDATE IMSCON** TYPE(LINK) NAME(*linkname*) MSC(*lclplk_id*) STOP(COMM)
- In the WTOR format, **STOPLINK** *linkname lclPlkid*
- In the z/OS MODIFY format, **DELETE LINK** NAME (*linkname*) LCLPLKID(*lclPlkid*)

Results

IMS Connect issues message HWS3310I after the link is stopped.

Related reference
IMS Connect commands (Commands)
UPDATE IMSCON commands (Commands)

Stopping an MSC physical link in IMS Connect

Do not stop an MSC physical link in IMS Connect, unless you need to clear the affinity of the link with an IMS system when TCP/IP generic resources are used.

About this task

Recommendation: Under normal circumstances, use the IMS command /PSTOP to terminate MSC physical links. Stop MSC physical links in IMS Connect only when you need to clear the affinity of an MSC physical link with a particular IMS system that uses TCP/IP generic resources.

When you stop an MSC physical link in a local IMS Connect instance by using a command like STOPMSC, IMS Connect takes the following action:

- Stops communication on the specified MSC physical link, including communications on all the MSC logical links that are assigned to the physical link.
- Informs IMS that communication has stopped on the physical link so that IMS can also terminate the physical link and any logical links that are assigned to the physical link.
- Changes the status of the MSC physical link and its assigned logical links to NOT ACTIVE.
- For TCP/IP generic resources, clears affinity of a physical link to the IMS system
- Issues message HWSF3305I

Use the VIEWMSC command or the QUERY MSC command to display information about MSC physical links that are defined to IMS Connect.

To stop an MSC physical link in a local instance of IMS Connect:

Procedure

Issue any one of the following IMS Connect commands in the local IMS Connect instance:

- In the IMS type-2 format, **UPDATE IMSCON** TYPE(MSC) NAME(*lclPlkid*) STOP(COMM)
- In the WTOR format, **STOPMSC** *lclPlkid*
- In the z/OS MODIFY format, **UPDATE MSC** NAME (*lclPlkid*) STOP(COMM)

Results

IMS Connect issues message HWS3310I after the link is stopped.

Related concepts

MSC TCP/IP generic resources (Communications and Connections)

Related reference

IMS Connect commands (Commands)

UPDATE IMSCON commands (Commands)

Operating ISC TCP/IP connections

The communication path of an ISC TCP/IP connection passes through multiple IMS components: the IMS control region, the Structured Call Interface (SCI) of the Common Service Layer (CSL), and IMS Connect.

About this task

Operational tasks for an ISC TCP/IP connection in one segment of an ISC TCP/IP connection impacts the connection in the other segments. Also, general operational tasks in the IMS system are also likely to impact any ISC TCP/IP connections.

Under normal circumstances, you start, stop, and restart an ISC link from IMS by using the /OPNDST and /QUIESCE type-1 commands. The /OPNDST command starts or restarts an ISC link to a remote CICS subsystem. The /QUIESCE command stops an ISC link to a remote CICS subsystem after any in-progress work completes.

Occasionally, you might find it necessary to stop an ISC TCP/IP link from IMS Connect, such as might be the case if a link is shut down from IMS, but IMS Connect fails to properly clean up its resources that support the link. In IMS Connect, you can shut down an ISC link or you can shutdown a TCP/IP connection to a remote CICS subsystem. When you shut down an ISC link in IMS Connect, all sessions on that link are terminated, but communications with the remote CICS subsystem might continue on other ISC links. If you shutdown a connection to a remote CICS subsystem, all ISC links that connect to that remote CICS subsystem are stopped, including all sessions on those links.

In rare extreme cases, you could also stop an ISC link by shutting down communications between SCI and either IMS or IMS Connect; however, doing so would stop all communications in IMS or IMS Connect that require SCI, not just those that use ISC TCP/IP links.

Cleaning up an ISC parallel session in IMS Connect

If an ISC parallel session has terminated in IMS, but the IMS Connect resources associated with the parallel session did not clean up automatically, you can clean up the resources by using the IMS type-2 command **UPDATE IMSCON TYPE(ISCUSER)**.

About this task

Recommendation: Under normal circumstances, use the IMS **/QUIESCE NODE** command to terminate ISC parallel sessions. Stop an ISC parallel session in IMS Connect only if the IMS Connect resources that are associated with the ISC parallel session fail to clean up correctly after the session is terminated in IMS.

When you clean up ISC parallel sessions in a local IMS Connect instance by using the **UPDATE IMSCON TYPE(ISCUSER)** command, IMS Connect takes the following actions:

- Stops any communication on the ISC parallel session
- Notifies the local IMS that communication has stopped on the parallel session
- Deletes the control blocks associated with the parallel session and frees the associated storage
- Issues message HWSG4010I

Procedure

To stop an ISC parallel session and clean up its associated resources in IMS Connect, issue the **UPDATE IMSCON TYPE(ISCUSER)** command.

Results

IMS Connect issues message HWSG4010I after the link is stopped.

Related reference

QUERY IMSCON TYPE(ISCUSER) command (Commands)

/QUIESCE command (Commands)

UPDATE IMSCON TYPE(ISCUSER) command (Commands)

Stopping an ISC link in IMS Connect

You can stop an ISC link in IMS Connect, but avoid doing so unless the ISC link did not clean up properly after the link was terminated in IMS or you need to prevent sessions from starting on the ISC link.

About this task

Under normal circumstances, stop ISC links from IMS by using the IMS type-1 command **/QUIESCE NODE**, which terminates the ISC link after any in-progress work completes. If the link must be terminated immediately without waiting for in-progress work to complete, use the IMS type-1 command **/CLSDST NODE**.

To stop an ISC link in a local instance of IMS Connect:

Procedure

Issue the IMS type-2 format command **UPDATE IMSCON** TYPE(ISC) NAME(*iscstmtid*) STOP(COMM) in the local IMS Connect instance.

Results

In response to the command, IMS Connect takes the following action:

- Stops communication on the specified ISC link, including communications on all the ISC sessions that are assigned to the link.
- Closes the associated send and receive sockets.
- Informs IMS that communication has stopped on the ISC link so that IMS can also terminate the link and any sessions that are assigned to the link.
- Changes the status of the ISC link and its assigned logical links to NOT ACTIVE.
- Issues message HWSG4005I.

Tip: Use the QUERY IMSCON TYPE(ISC) command to display information about the ISC links that are defined to IMS Connect.

What to do next

Before the link can be used again, the link must be started in IMS Connect by the **UPDATE IMSCON** TYPE(ISC) NAME(*iscstmtid*) START(COMM) command.

Related tasks

"Restarting an ISC link in IMS Connect" on page 195
You need to restart an ISC link in IMS Connect after an ISC link was stopped previously by an UPDATE IMSCON TYPE(ISC) NAME(*isclnkid*) STOP(COMM) command.

Related reference

QUERY IMSCON TYPE(ISC) command (Commands)

UPDATE IMSCON TYPE(ISC) command (Commands)

Restarting an ISC link in IMS Connect

You need to restart an ISC link in IMS Connect after an ISC link was stopped previously by an UPDATE IMSCON TYPE(ISC) NAME(*isclnkid*) STOP(COMM) command.

About this task

In rare circumstances, you might need to restart an ISC link after the link failed unexpectedly in IMS Connect.

ISC communication does not resume until the ISC session is also restarted in IMS by issuing the /OPNDST command.

To restart an ISC link in a local instance of IMS Connect:

Procedure

Issue the IMS type-2 format command **UPDATE IMSCON** TYPE(ISC) NAME(*isclnkid*) START(COMM) in the local IMS Connect instance.

Results

IMS Connect issues message HWSG4000I when the link restarts successfully.

What to do next

Related tasks

"Stopping an ISC link in IMS Connect" on page 194
You can stop an ISC link in IMS Connect, but avoid doing so unless the ISC link did not clean up properly after the link was terminated in IMS or you need to prevent sessions from starting on the ISC link.

Related reference

QUERY IMSCON TYPE(ISC) command (Commands)

UPDATE IMSCON TYPE(ISC) command (Commands)

Stopping a connection to a remote CICS subsystem from IMS Connect

You can stop an ISC TCP/IP connection to a remote CICS subsystem from IMS Connect.

About this task

Recommendation: Avoid stopping a connection to a remote CICS subsystem unless the connection did not clean up properly after the connection was shut down in IMS or you need to prevent sessions from opening to the remote CICS subsystem.

Procedure

To stop an ISC TCP/IP connection to a remote CICS subsystem, issue the IMS type-2 command **UPDATE IMSCON** TYPE(RMTCICS) NAME(*rmtcics_id*) STOP(COMM)

Results

IMS Connect performs the following actions when an ISC TCP/IP connection to a remote CICS subsystem is stopped in IMS Connect:

- Informs IMS that communication stopped on the ISC link so that IMS can also terminate the link and any sessions that are assigned to the link.
- Disconnects the TCP/IP sockets associated with the ISC TCP/IP connection.
- Changes the status of the connection to STOPPED.
- Issues a message HWSV4405I to the console and notifies IMS that the connection was stopped.

If IMS Connect receives any messages from IMS for the stopped connection, IMS Connect rejects the message, returns a NAK response message to IMS, and issues message HWSG4040W.

Related tasks

“Restarting a connection to a remote CICS subsystem in IMS Connect” on page 196
You can restart an ISC TCP/IP connection that is stopped in IMS Connect by issuing an online IMS Connect command.

Related reference

QUERY IMSCON TYPE(RMTCICS) command (Commands)
UPDATE IMSCON TYPE(RMTCICS) command (Commands)

Restarting a connection to a remote CICS subsystem in IMS Connect

You can restart an ISC TCP/IP connection that is stopped in IMS Connect by issuing an online IMS Connect command.

About this task

In rare circumstances, you might need to restart a connection to a remote CICS after the connection failed unexpectedly in IMS Connect.

ISC communication does not resume until the ISC session is also restarted in IMS by issuing the /OPNDST command.

Procedure

To restart an ISC TCP/IP connection in IMS Connect, issue the IMS type-2 command **UPDATE IMSCON** TYPE(RMTCICS) NAME(*rmtcics_id*) START(COMM).

Results

IMS Connect issues message HWSV4400I when the connection is restarted.

Related tasks

"Stopping a connection to a remote CICS subsystem from IMS Connect" on page 196
You can stop an ISC TCP/IP connection to a remote CICS subsystem from IMS Connect.

Related reference

QUERY IMSCON TYPE(RMTCICS) command (Commands)

UPDATE IMSCON TYPE(RMTCICS) command (Commands)

MSC operations

In a non-sysplex environment, each IMS system in a Multiple Systems Coupling (MSC) configuration is operationally an independent unit. Each IMS system exclusively owns its own communication resources, and is controlled by its own master terminal.

Related concepts

"IMS-to-IMS TCP/IP connection operations" on page 183
The communication path of an IMS-to-IMS TCP/IP connection passes through multiple IMS components: the IMS control region, the Structured Call Interface (SCI) of the Common Service Layer (CSL), IMS Connect, and either the IMS Multiple Systems Coupling (MSC) component or the IMS Open Transaction Manager Access (OTMA) component.

Related tasks

"Modifying Multiple Systems Coupling resources" on page 34
After modifying MSC resources, use the **/MSVERIFY** command to ensure that the assignment produced a valid configuration. The **/MSVERIFY** command verifies the consistency of MSC system identifications (SYSIDs) and logical link paths (MSNAMEs) across two systems.

MSC initialization

To start Multiple Systems Coupling (MSC) communications between two IMS systems, issue either the IMS type-1 command **/RSTART LINK** or the IMS type-2 command **UPDATE MSLINK** NAME(*linkname*) START(COMM).

If the physical link type is either a channel-to-channel connection or a main-storage to main-storage connection (MTM), you must issue the command in both IMS systems at each end of the link.

If the physical link type is either TCP/IP or VTAM, you need to issue the command in only one IMS system. The normal procedure is for the operator to issue this command when a system has started. Communication is allowed only if the characteristics of the specified links are compatible. If a required link is not successfully started, messages wait until the links are reassigned.

The SYSID of each IMS system running in a shared queues group can either be cloned across all systems, or not. If the SYSIDs are not cloned, initialize all IMS systems in the shared queues group before starting regions to process the transactions on the shared queue to avoid occurrences of pseudoabend U0830.

After all IMS systems have been initialized, transaction processing can begin, and individual IMS systems can be brought down and later restarted as needed. Initialization allows all IMS systems to exchange MSC SYSIDs and MSNAMEs, and creates a SYSID table that contains all of the SYSIDs for that shared queues group. If the SYSIDs for all IMS systems in a shared queues group *are* cloned, then you do not need to perform this initialization.

Attention: If you cold start a non-shared queues system that has messages queued, the queued messages are lost. Because the messages that were lost can be from or to terminals and programs in other systems, the impact of a cold start is not limited to the cold-started system.

In a shared-queues environment, IMS systems in the IMSplex exchange SYSIDs and MSNAMEs at initialization. During this SYSID exchange, each IMS system creates dynamic MSNAMEs for any MSNAMEs that are defined in other IMS systems that it does not define itself. The dynamic MSNAMEs result in paths to all of the IMS systems in the IMSplex. The SYSIDs are merged to create a common SYSID routing table.

The table is the same in each IMS. Therefore, any local SYSIDs are local in all IMS systems and override any remote or undefined SYSIDs. Remote SYSIDs override only undefined SYSIDs.

MSC termination

You terminate Multiple Systems Coupling (MSC) links from either of the two linked IMS systems by issuing either a type-1 command **/PSTOP LINK** or a type-2 command **UPDATE MSLINK NAME**(*linkname*) **STOP(COMM)**. When transmission terminates on one side, its partner in the other system terminates its own transmission and notifies the MTO.

Use the command forms shown in the following table to terminate links between partner MSC systems.

Table 33. Commands to terminate links

Command	Keyword	MSC use and effect
/IDLE	LINK	IMS forces termination of all transmissions on the physical link associated with the named logical link. Use only after a shutdown checkpoint.
	NOSHUT	For TCP/IP and VTAM link types only, IMS forces the idle condition without taking a shutdown checkpoint.
/PSTOP	LINK	IMS halts transmissions associated with the logical link between two partner systems, but continues queuing for the remote resources. IMS enqueues, but does not send, broadcast messages that would use the link. IMS stops the logical link in the partner system, and sends message DFS2161 to its MTO.
	LINK PURGE	IMS forces PSTOP condition even if other system failed while channel-to-channel (CTC) I/O was in progress. CTC link type only.
	MSPLINK	Stops logons to an MSC TCP/IP or VTAM physical link and enables the operator to issue the /MSASSIGN command to reassign logical links to the physical link. Any links in sessions that have not been stopped by the /PSTOP command are not affected by an /MSASSIGN command. For physical links in a TCP/IP generic resource group, after the physical link is stopped, all logical links must terminate normally to clear link affinity before they can be moved to a different IMS system.

Table 33. Commands to terminate links (continued)

Command	Keyword	MSC use and effect
UPDATE MSLINK NAME(*linkname*)	STOP(COMM)	Halts transmissions associated with the logical link between two partner systems, but queuing for the remote resources continues. Broadcast messages that would use the link are queued but not sent. This command is equivalent to the type-1 command **/PSTOP LINK**. The logical link is stopped in the partner system and its MTO is notified by message DFS2161.
	OPTION(FORCE)	This command applies to TCP/IP and VTAM link types and forces a **/PSTOP** condition. OPTION(FORCE) leaves the link in PSTOPPED IDLE ERE mode. The subsequent restart of the link is an emergency restart. For VTAM link types, STOP(COMM) without OPTION(FORCE) must be issued before OPTION(FORCE) is used. For TCP/IP links, OPTION(FORCE) can be used any time. OPTION(FORCE) might need to be issued on both sides of the link.
UPDATE MSPLINK NAME(*msplinkname*)	STOP(LOGON)	Stops logons to the physical link. This applies to MSC TCP/IP and VTAM links only. It stops logons to the physical link and enables the operator to reassign logical links to the physical link.
	STOP(GENLOGON)	For MSC TCP/IP links in a TCP/IP generic resource group only, stops logons to the IMS system on the physical link. After the command stops logons to the physical link and all logical links on the physical link have terminated normally to clear link affinity with the IMS system, the logical links can be reassigned to a physical link on another IMS system.

Changing logical link assignments

You define initial logical link assignments (logical link to physical link) as part of the IMS system definition process. However, you can change logical link assignments dynamically.

About this task

To dynamically make or change a logical link assignment, use either the type-1 command **/MSASSIGN** or the type-2 command **UPDATE MSLINK NAME***(linkname)* **SET(MSPLINK***(msplinkname)*).

Recommendation: Use the **/MSASSIGN** command or the **UPDATE MSLINK NAME***(linkname)* **SET(MSPLINK***(msplinkname)*) command to create logical links only for unscheduled reassignments resulting from failing physical connections or systems.

Because a logical link must always communicate with its partner, the operators for the two systems must coordinate their assignments of corresponding physical links. You can replace any type of physical link with any other type of physical link.

Changes to logical link assignments remain in effect until you change them using an **/MSASSIGN** command, **UPDATE MSLINK NAME***(linkname)* **SET (MSPLINK***(msplinkname)*) command, or cold start IMS. The commands only alter the relationship of MSC resources in the local system.

The logical link relationships you can change and the commands you can use to change them include:

- Logical link to physical link
  - Type-1 command **/MSASSIGN**.

  - Type-2 command **UPDATE MSLINK NAME***(linkname)* **SET (MSPLINK***(msplinkname)*).
- Remote SYSID to logical link
  - Type-1 command **/MSASSIGN**.
  - Type-2 command **UPDATE MSNAME NAME***(linkname)* **SET (MSPLINK***(msplinkname)*) assigns an MSNAME with the desired remote SYSID to a logical link. Alternately, you can modify the remote SYSID of the MSNAME already associated with the logical link by issuing the type-2 command **UPDATE MSNAME NAME***(msname)* **SET(SIDR** *(remote_sysid)*).
- Logical link path to logical link:
  - Type-1 command **/MSASSIGN**.
  - Type-2 command **UPDATE MSNAME** NAME(*msname*) SET(MSLINK(*mslinkname*)).

You can also use the **/MSASSIGN** command to complete the following tasks:

- Change a remote program to local
- Change a local program to remote
- Assign a logical path

When you modify logical link assignments:

- Operating logical or physical links must be assigned one-to-one, except for TCP/IP and VTAM link types, which can have parallel sessions in effect. In this case, multiple logical links can be assigned to one physical link.
- If the link uses TCP/IP or VTAM, before you assign a logical link either to or from an MSPLINK name you must stop the physical link by executing one of the following commands:
  - The type-1 command **/PSTOP MSPLINK**
  - The type-2 command **UPDATE MSPLINK NAME***(msplinkname)* STOP(LOGON)
  - In a TCP/IP generic resource group, the type-2 command **UPDATE MSPLINK NAME***(msplinkname)* STOP(GENLOGON)
- Before reassigning a logical link, you must stop the logical link by using either a type-1 command **/PSTOP** or a type-2 command **UPDATE MSLINK NAME***(linkname)* **STOP(COMM)** and the logical link must be idle.
- A destination SYSID cannot be reassigned to a logical link unless its currently assigned logical link is idle after a type-1 command **/PSTOP** or a type-2 command **UPDATE MSLINK NAME***(linkname)* **STOP(COMM)**.
- MSC communication can occur only when logical links in two IMS systems share the same partner identification and have assignment to an operating communications facility between the systems.
- IMS does not determine whether the requested logical link assignment is reasonable or results in a valid configuration for MSC communication. This is only done by communication with appropriate remote systems. You can accomplish this using the **/MSVERIFY** command after making assignment changes.

Restarting a logical link

If a restart is pending on a logical link because of a physical link failure, reestablish communications between both systems through an alternative physical link.

About this task

Restriction: In a TCP/IP generic resource group, where a physical link is defined in multiple IMS systems, if a logical link is in a PSTOPPED ERE state on any IMS system in the group, do not start a logical link that uses the same physical link on any other IMS system in the group. Before you can start a logical link on the same physical link on another system, you must clear the affinity of the ERE link either by shutting it down normally or by resetting it to a COLD state.

As a precaution, display the affinity status of an MSC link on its current IMS system before moving the link by restarting it on another IMS system. If the affinity is still active, shutdown the link normally, which resets the affinity status. If the link cannot be shut down normally, set the link to COLD status by issuing either the type-1 command **/CHANGE LINK** *linknum* FORCSESS | SYNCSESS COLDSESS or the type-2 command **UPDATE MSLINK**NAME(*linkname*) SET(SYNCOPT(COLDSESS)

Attention: Changing the link status from ERE to COLD or moving a link in ERE status and restarting it on another IMS system prevents the synchronization of the message sequence numbers during restart, which can cause the duplication or loss of messages.

Procedure

1. If the link uses TCP/IP or VTAM, before you assign a logical link either to or from an MSPLINK name, stop the physical link by issuing one of the following commands:
   - The type-1 command **/PSTOP MSPLINK**
   - The type-2 command **UPDATE MSPLINK** NAME*(msplinkname)* STOP(LOGON)
   - For links in a TCP/IP generic resource group only, the type-2 command **UPDATE MSPLINK** NAME*(msplinkname)* STOP(GENLOGON)
2. Reassign the logical link to the alternate physical link by using either of the following commands:
   - The type-1 command **/MSASSIGN**
   - The type-2 command **UPDATE MSLINK NAME***(linkname)* SET(MSPLINK*(msplinkname)*)
3. If the link uses TCP/IP or VTAM, restart the physical link by issuing one of the following commands:
   - The type-1 command **/RSTART MSPLINK**
   - The type-2 command **UPDATE MSPLINK** NAME*(msplinkname)* START(LOGON)
   - For links in a TCP/IP generic resource group only, the type-2 command **UPDATE MSPLINK** NAME*(msplinkname)* START(GENLOGON)
4. Start the logical link by issuing one of the following commands:
   - The type-1 command **/RSTART LINK**
   - The type-2 command **UPDATE MSLINK NAME***(linkname)* **START(COMM)**

Switching TCP/IP and VTAM physical link types

Because of the operational similarities between MSC physical links that use TCP/IP and those that use VTAM, the VTAM- and TCP/IP-type physical links are well-suited to serve as a back-up link types for each other.

Before you begin

Prerequisite: In advance of any failure, you must define a redundant set of physical links that use the alternative link type and that duplicate the primary set of MSC physical links. Both TCP/IP and VTAM physical links are defined by using either the MSPLINK system definition macros or the type-2 CREATE MSPLINK command.

Any supporting components required by the backup physical link type must also be pre-configured and ready for use. For example, if TCP/IP is your backup link type, the Structured Call Interface (SCI) component of the Common Service Layer (CSL) and IMS Connect must also be configured and ready for use.

About this task

The following steps assume a scenario in which the primary physical links in use are VTAM physical links and the back up physical links are TCP/IP. However, the steps are also a general guide for switching from TCP/IP to VTAM.

The steps reassign logical links on a failed VTAM physical link between a front-end IMS system and a back-end IMS system to a backup TCP/IP physical link that was previously defined between the two systems.

In a failure scenario like this, any terminals on the front-end IMS system that were waiting for a response are likely hung and message DFS3222I has been issued to the front-end master terminal operator (MTO). On the back-end IMS system, the IMS applications have stopped receiving messages for processing and message DFS3222I has also been issued to the back-end MTO.

The operators at the front-end and back-end IMS systems must coordinate the timing of the steps to ensure that messages from the front-end IMS system do not start flowing before the backend IMS system is ready to receive them.

Procedure

1. At the front-end IMS system, stop the remote transactions that route messages on the failed physical link. When the remote transactions are stopped, any terminal that submits requests for the remote transactions receive message DFS065.
2. At the back-end IMS system:
   a) Confirm that a local instance of IMS Connect is active and appropriately configured to support the backup physical link from the back-end IMS system.

      For each MSC TCP/IP physical link that the IMS Connect instance supports, at least one RMTIMSCON statement must be defined and one MSC statement for each physical link must be defined in the IMS Connect configuration member in the IMS.PROCLIB data set.
   b) Confirm that a local instance of SCI is active and available for use by the back-end IMS system and IMS Connect.
   c) Reassign the logical links from the failed VTAM physical link to the TCP/IP physical link by issuing either of the following commands:
      - The IMS type-2 command UPDATE MSLINK NAME(*linkname*) SET(*plinkname*)
      - The IMS type-1 command /MSASSIGN LINK *xxx* to MSPLINK *plnkname*
3. At the front-end IMS system:
   a) Confirm that a local instance of IMS Connect is active and appropriately configured to support the backup physical link from the front-end IMS system.

      For each MSC TCP/IP physical link that the IMS Connect instance supports, at least one RMTIMSCON statement must be defined and one MSC statement for each physical link must be defined in the IMS Connect configuration member in the IMS.PROCLIB data set.
   b) Confirm that a local instance of SCI is active and available for use by the front-end IMS system.
   c) Reassign the logical links that use the failed VTAM physical link to the TCP/IP physical link by issuing either of the following commands:
      - The IMS type-2 command UPDATE MSLINK NAME(*linkname*) SET(*plinkname*)
      - The IMS type-1 command /MSASSIGN LINK *xxx* to MSPLINK *plnkname*
   d) After the logical links have been reassigned to the same TCP/IP physical link at both the front-end and back-end IMS systems, start the logical links by issuing either of the following commands:
      - The IMS type-2 command UPDATE MSLINK NAME(*linkname*) START(COMM)
      - The IMS type-1 command /RSTART LINK *xxx*
   e) If the remote transactions were stopped, restart them.

Related concepts

MSC physical links (Communications and Connections)

“Recovery considerations for multiple systems” on page 208

Each system in the multisystem configuration uses the full recovery capabilities of IMS. These capabilities assure that messages are not lost or duplicated within the single system.

Related tasks

Defining IMS-to-IMS TCP/IP connections for MSC (System Definition)

Related reference

MSPLINK macro (System Definition)

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

CREATE MSPLINK command (Commands)

MSC TCP/IP link operations

Operating procedures for MSC TCP/IP links are generally the same as the operating procedures for MSC VTAM links; however, a few differences do exist.

IMS Connect manages the TCP/IP connections that are required for MSC TCP/IP. Communication between MSC and IMS Connect is managed by the Structure Call Interface (SCI) component of the IMS Common Service Layer (CSL).

Under normal circumstances, the use of TCP/IP, IMS Connect, and SCI is not apparent during the operation of an MSC TCP/IP link. MSC TCP/IP links are started and stopped by using the same IMS commands that are used to initialize and terminate VTAM-type links. When you issue an **UPDATE MSLINK** NAME(*linkname*) START(COMM) command for a TCP/IP link in a local IMS system, the link is started in both the local and remote IMS and IMS Connect pairs.

IMS Connect also provides commands to control MSC TCP/IP links. Although you can stop a TCP/IP link by using IMS Connect commands, only do when a link has failed to clean up properly in IMS Connect after terminating in IMS.

Other differences between the operating procedures for MSC TCP/IP links and the other MSC link types are found in the error recovery procedures and in the management of affinities when TCP/IP generic resources are used.

Related concepts

“Monitoring IMS Connect connections” on page 108

IMS manages TCP/IP connections through IMS Connect, which serves as the TCP/IP gateway to IMS. The IMS Connect function can also be viewed as a TCP/IP socket server.

MSC and IMS-to-IMS TCP/IP communications (Communications and Connections)

Related tasks

IMS-to-IMS TCP/IP connections (System Definition)

Related reference

/PSTOP command (Commands)

QUERY IMSCON commands (Commands)

QUERY MSLINK command (Commands)

QUERY MSPLINK command (Commands)

/RSTART command (Commands)

UPDATE MSLINK command (Commands)

UPDATE MSPLINK command (Commands)

IMS Connect WTOR commands (Commands)

IMS Connect z/OS commands (Commands)

Restarting MSC TCP/IP links after errors

If an IMS system or an MSC TCP/IP link terminates unexpectedly, you need to confirm that the MSC TCP/IP link has also been cleaned up properly in the remote IMS system.

About this task

In the event of a failure, IMS attempts to clean up the MSC TCP/IP links at both the local and remote site; however, if the failing IMS system is unable to notify the remote site of the failure, the remote site might not clean up the link.

Procedure

- To determine if a link has been cleaned up in the remote IMS system, you can issue either the type-1 format **/DISPLAY** LINK command or the type-2 format **QUERY MSLINK** NAME(*linkname*) command.
- After an MSC TCP/IP link terminates at a local site, if the remote site does not also clean up the link, issue a **/PSTOP LINK** FORCE or an **UPDATE MSLINK** STOP(COMM) OPTION(FORCE) in the remote IMS system.

  If the link is restarted in the local IMS system before either the link or socket connection is cleaned up at the remote site, the link or socket connection is cleaned up at the remote site during restart processing.
- To restart a TCP/IP link after an IMS system is started, issue either an **/RSTART** command or the **UPDATE MSLINK** START(COMM) command.
- To set a stopped MSC TCP/IP link to its coldstart status, issue either the type-1 format command **/CHANGE LINK** COLDSESS or the type-2 format command **UPDATE MSLINK** SET (SYNCOPT(COLDSESS)).

  The link must be stopped first. When specifying the COLDSESS keyword, the SYNCSESS or FORCSESS keyword must also be specified.

Related reference

/CHANGE LINK command (Commands)
QUERY IMSCON commands (Commands)
QUERY MSLINK command (Commands)
QUERY MSPLINK command (Commands)
/RSTART command (Commands)
UPDATE MSLINK command (Commands)
UPDATE MSPLINK command (Commands)
IMS Connect WTOR commands (Commands)
IMS Connect z/OS commands (Commands)

Commands that help control resources in an MSC environment

Uses and effects of Multiple System Coupling (MSC) commands are described so that you can assess their effects in your MSC environment.

If an MSC error occurs when the MSC trace is not operational, check the log for X'67' records, because some error information is logged even when the link trace is not set on.

When you suspect link problems, you can use either of the following two commands to start a trace:

- Type-1 command **/TRA SET ON LINK X LEVEL 3 MODULE ALL**.
- Type-2 command **UPDATE MSLINK NAME(mslinkname) START(TRACE)**. This command uses the same level and module settings that were used the last time the **/TRACE SET (ON) LINK** command was issued. If a **/TRACE SET (ON) LINK** command has not been issued since the last cold start, this command defaults to MODULE=ALL and LEVEL=4.

Table 34. MSC environment commands

Command	MSC use and effect
/ASSIGN	Change transaction priorities.
/BROADCAST	Use of the LTERM keyword value ALL sends the broadcast message to all terminals of the local system only. No general broadcast capability exists for remote MSC terminals. Their LTERMs must be defined in the broadcasting system and specified in the command. Broadcasts to MTOs of remote MSC systems are possible even when the MTO terminals are not defined in the broadcasting system; use the keyword MASTER. The system must have defined the SYSIDs of such remote systems by using MSNAME macros or by using the type-2 CREATE MSNAME command.
/CHANGE	Change link mode table and session restart.
/PSTOP	Queue for remote processing, but do not send.
/PURGE	Reject subsequent primary requests for remote processing. Accept secondary or continued conversational requests. Continue sending those requests that are allowed to queue.
/STOP	If remote program or terminal stopped in input system: Return DFS065 to input terminal if a primary request. Queue but do not send secondary requests. If local program or terminal accessed by remote MSC systems: Queue secondary requests. Return message DFS065 to input terminal in input system for primary requests. Message DFS065 identifies, by SYSID, the MSC processing system that rejected the message. Message is logged, but canceled.
/START	Reset previous /START and /PSTOP effects.
/TRACE	Using the LINK keyword, you can invoke a trace of MSC operation. Trace data is recorded within type X'67' records on the IMS system log.
UPDATE MSLINK	Using the UPDATE MSLINK command, you can set or change various logical link attributes, including starting and stopping the logical link and starting and stopping traces of the logical link. The command requires the Operations Manager (OM) API. The command syntax for this command is defined in XML and is available to automation programs that communicate with OM. Using the UPDATE command, you can also modify MSLINK parameter values BUFSIZE=, MODETBL=, PARTNER (ID)=, and the MSLINK name (MSLINK label field).
UPDATE MSNAME	Using the UPDATE MSNAME command, you can set or change various logical link path attributes, including starting and stopping the queuing or sending of messages to logical link path. The command requires the Operations Manager (OM) API. The command syntax for this command is defined in XML and is available to automation programs that communicate with OM. Using the UPDATE command, you can also modify the MSNAME parameter value SYSID=.
UPDATE MSPLINK	Using the UPDATE MSPLINK command, you can set or change various physical link attributes, including preventing or allowing logons to the physical link. The command requires the Operations Manager (OM) API. The command syntax for this command is defined in XML and is available to automation programs that communicate with OM. Using the UPDATE command, you can also modify MSPLINK parameter values such as ADDR=, BACKUP=, BUFSIZE=, MODETBL=, SESSION=, and the physical link name (MSPLINK label field).

Displaying information about an MSC network

There are two commands that you can use to view information about an MSC network: the type-1 command **/DISPLAY** and the type-2 command **QUERY**.

/DISPLAY command and MSC

The IMS type-1 **/DISPLAY** commands show available information that helps you manage an MSC environment. These facilities operate only within the domain of the local system, however, and do not provide information that is available only from remote systems. The following table summarizes the information that is available to the MTO with the **/DISPLAY** command.

Table 35. MSC information returned by the /DISPLAY command

Command	Keyword	MSC use and effect
/DISPLAY AFFIN	LINK	Displays the current link affinities that logical links have within a TCP/IP or VTAM generic resource group. Also displays the name and number of the logical links and for VTAM links, the node.
/DISPLAY ASMT	LINK	The physical link, the local and remote system IDs, and logical link paths assigned to the specified logical link.
	MSNAME	Logical link number, physical link, local and remote system IDs, MSNAME label, and IMS ID.
	MSPLINK	Logical link number, physical link, physical link type (CTC, MTM, TCP/IP, VTAM), physical link address, 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.
	SYSID	The physical link, logical link, and logical link path assignments associated with the specified system identification.
/DISPLAY LINK	[None]	Logical link number.
		Partner ID.
		The local totals for the numbers of messages received, sent, enqueued, dequeued, currently queued for link, and link status.
	MODE	For VTAM links, displays various modes in which VTAM terminals can operate.
	OPTION BUFSIZE	Displays the link number, link name, bandwidth, buffer size, and the link status.
	QCNT	The global totals for the numbers of messages received, sent, enqueued, dequeued, currently queued for link, and link status.
/DISPLAY MSNAME	[None]	Logical link path name (MSNAME) and the local totals for the numbers of messages enqueued, dequeued, and currently queued to the MSNAME.
	QCNT	Logical link path name (MSNAME) and the global totals for the numbers of messages enqueued, dequeued, and currently queued to the MSNAME.

QUERY command and MSC

The IMS type-2 **QUERY** commands display available attribute and status information to help you manage an MSC environment. The following table summarizes the MSC information returned by the **QUERY** command.

Table 36. MSC information returned by the QUERY command

Command	Keyword	MSC use and effect
QUERY MSLINK	BANDWIDTH(ON|OFF)	Identifies logical links that are or are not in bandwidth mode.
	SHOW(ALL|*attribute*)	Returns the user-defined attributes of logical links. You can limit the information returned by specifying a valid attribute as a parameter of the SHOW() keyword.
	STATUS(*status_type*)	Identifies the logical links that have the specified status type.
QUERY MSNAME	QCNT()	Identifies logical link paths that have queue counts relative to a number you specify. For example, you can view all logical link paths with a queue count that is greater-than-or equal-to 100.
	SHOW(ALL|*attribute*)	Returns the user-defined attributes of logical link paths. You can limit the information returned by specifying a valid attribute as a parameter of the SHOW() keyword.
	STATUS(*status_type*)	Identifies the logical link paths that have the specified status type.
QUERY MSPLINK	SHOW(ALL|*attribute*)	Returns the user-defined attributes of physical links. You can limit the information returned by specifying a valid attribute as a parameter of the SHOW() keyword.
	STATUS(*status_type*)	Identifies the physical links that have the specified status type.
	TYPE(*link_type*)	Identifies physical links by their link types: CTC, MTM, TCP/IP, and VTAM. You can query one or more link types at a time.

Related tasks

"Viewing MSC link information in IMS Connect" on page 186
When an IMS Connect to IMS Connect connection is used for MSC, you can view configuration and status information about the MSC links by issuing an IMS Connect command.

"Viewing MSC connection information for IMS-to-IMS TCP/IP communications" on page 184
You can view configuration and status information for IMS-to-IMS TCP/IP connections that are used for MSC by issuing IMS Connect commands.

Logical link path control

You can use several commands to control logical link paths. The logical link path is the lowest level of control across a Multiple Systems Coupling (MSC) environment because it is the lowest level that is necessarily defined in intermediate MSC systems.

Logical link paths are defined by SYSID pairs that identify the sending and destination systems. You define logical link paths using the MSNAME system definition macro or by using the type-2 CREATE MSNAME command. The following table summarizes the commands you can use for link control.

Table 37. Commands used to control MSC link paths

Command	Keyword	MSC use and effect
/START	MSNAME	Start a previously stopped MSNAME.

Table 37. Commands used to control MSC link paths (continued)

Command	Keyword	MSC use and effect
/STOP	MSNAME	Stop the sending or receiving of primary request messages associated with the logical link path. When stopped by an input system, primary requests for remote programs or terminals associated with the stopped logical link path are canceled and IMS returns message DFS065 to the input terminal. Conversations in progress are allowed to continue. When stopped by a destination system, messages received from other systems over a stopped logical link path cause a logical link path to be stopped in the sending system (input or intermediate); IMS issues messages DFS2140 and DFS2142 (respectively) to the MTOs of the sending and receiving systems. Messages remain enqueued in the sending system until the logical link is subsequently started in both systems.
/PURGE	MSNAME	Halt enqueuing of primary requests for all remote terminals and programs represented by the MSNAME. Continuing conversations and secondary requests are still handled. Primary requests entered through an input terminal receive message DFS065. Requests from other systems that require use of the logical link path for a response are not accepted, but remain enqueued in the sending system (see /STOP MSNAME above).
UPDATE MSNAME	SET(), START(), and STOP()	Using the **UPDATE MSNAME** command, you can set or change various logical link path attributes, including starting and stopping the queuing or sending of messages to logical link path. The command requires the Operations Manager (OM) API. The command syntax for this command is defined in XML and is available to automation programs that communicate with OM.

Recovery considerations for multiple systems

Each system in the multisystem configuration uses the full recovery capabilities of IMS. These capabilities assure that messages are not lost or duplicated within the single system.

You will not lose messages under the following conditions:

- You do not cold start the subsystem or emergency restart (BUILDQ) if using an earlier checkpoint.
- No log records are lost.

VTAM provides message integrity for SDLC links in addition to the MSC control functions.

Related tasks

"Switching TCP/IP and VTAM physical link types" on page 201
Because of the operational similarities between MSC physical links that use TCP/IP and those that use VTAM, the VTAM- and TCP/IP-type physical links are well-suited to serve as a back-up link types for each other.

Message recovery

IMS assures that messages are not lost or duplicated across a Multiple Systems Coupling (MSC) link by logging information about a message in both the sending and receiving systems.

IMS restores this information during restart and exchanges it between the systems after the link is established. The sending system can then dequeue a message that was received by the receiving system but for which the acknowledgment was lost due to a link or system failure. The sending system can also

re-send a message that was sent but not enqueued by the receiving system. If an IMS subsystem fails to recover, the messages for which it has recovery responsibility are lost.

Because IMS provides commands to dynamically change link assignments, you can set up an alternate processor for an inoperable one. The IMS system that resides in the inoperable processor can run in the alternate processor after all involved links are properly reassigned by the master terminal operators (MTOs).

Stopped transactions

If IMS stops a destination transaction, the action taken by the destination system varies based on the type of request. Requests can be conversational or nonconversational.

For a primary request that is not conversational or that starts a conversation, IMS sends an error message to the input terminal and cancels the message.

For a primary request that continues a conversation or a secondary request, IMS enqueues the message. If the request is the first one received for that stopped transaction, IMS also sends a message to the MTO at that transaction's local system.

Application program abnormal termination

When an application program abnormally terminates, and the abnormal termination is not the result of a deadlock situation, IMS issues message DFS554 to the master terminal of the system in which the abnormal termination occurred. This message includes the logical terminal name of the input terminal. If the input message is still available, IMS issues message DFS555, which includes the first portion of the input message, to the input terminal. When IMS sends the DFS554 message, the message text includes the logical terminal name of the input terminal.

Dequeuing messages, responses, and transactions

You can use the **/DEQUEUE** command with the **PURGE** or **PURGE1** keyword to dequeue messages, responses, and transactions that are in error for a specific MSC link.

About this task

When you enter the **/DEQUEUE MSNAME PURGE** command, the Message Control/Error exit routine (DFSCMUX0) is called before the command executes. The exit routine protects you from inadvertently entering the command in error and potentially destroying a message, response, or transaction by dequeuing it before it reaches its destination. The exit routine can suppress the dequeue of a message.

The system programmer must be aware of each of:

- The existence of the **/DEQUEUE** command, and the consequences of using it
- The interaction of the **/DEQUEUE** command with the Message Control/Error exit routine

The MTO should be aware of each of these facts:

- Your installation policy for using the **/DEQUEUE** command
- The fact that any messages that are dequeued are destroyed
- The /DEQUEUE command should be reserved for emergency use only, or as dictated by your site

Recommendations:

- Only authorize the MTO to issue the **/DEQUEUE** command.
- Establish installation-specific standards for using the **/DEQUEUE** command. Be sure to identify in advance and validate the scenarios where the MTO should use this command. Consider the **/DEQUEUE** command when coding the Message Control/Error exit routine.

Related reference

/DEQUEUE command (Commands)

Message Control/Error exit routine (DFSCMUX0) (Exit Routines)

Establishing maintenance procedures

Running IMS utilities on a regular, prescribed basis is one of the main tasks for maintaining IMS. Maintaining IMS might also involve additional tasks such as reorganizing databases and installing service regularly.

Maintaining IMS might involve the following activities (done manually or automated):

- Running reports, analyzing the reports, and taking action based on the analysis
- Reorganizing databases on a regular basis
- Performing preventive maintenance on the message queues
- Running IBM-supplied or vendor-supplied tools on a regular basis
- Installing service on a regular basis
- Running IMS-supplied utilities on a regular basis

The frequency with which utilities are run is an installation-dependent decision.

Tip: You can determine the current maintenance level of your IMS system by using the **/DIAGNOSE SNAP MODULE**(*modname*) command.

Setting up standard JCL

You can set up standard JCL for the jobs that must be run (such as the utility jobs) as part of establishing procedures for your operations people.

Using DBRC can greatly assist you in this area, even though having completely "canned" JCL is not possible. For example, the MTO must supply the required log data sets when running the Database Batch Backout utility. If you use DBRC for recovery control, you may only have to set up JCL for DBRC itself.

Restriction: DBRC does not develop JCL for backout.

Operator test procedures

After you have prepared procedures for operating IMS, all those who operate the system should be given an opportunity to familiarize themselves with the procedures. After that, all procedures should be thoroughly tested.

Simulating system failures

To test some of the recovery procedures, certain types of system failures must be simulated. The following table shows possible failures and how they can be simulated.

Table 38. Simulating system failures

System failure	Simulated by
BMP or CCTL thread abended or canceled in error	Use MTO command **/STOP REGION n ABDUMP**.
IMS control region abended or canceled in error	Use z/OS **MODIFY** or **CANCEL** command.
MPP abended, looping, or in wait state	Use one of your own programs, or use the DL/I test program DFSDDLT0; use the special calls ABEND and ZING.
CCTL abended, looping, or in wait state	Use one of your own programs.
I/O error on SLDS or RLDS	Use an SLDS or RLDS that has previously been damaged. Reset, rewind, and reload tape drive.
z/OS error (loop or abend)	Unplug or switch off z/OS system residence drive. Cancel IMS control region.

Table 38. Simulating system failures (continued)

System failure	Simulated by
Hardware error, no loss of real storage	Unplug or switch off z/OS system residence drive.
Power® failure, or hardware error with loss of real storage	Press System Reset, and re-IPL.

Qualities of good tests

The specific nature of your IMS use and installation setup will determine the details of your testing program. However, all good tests have these qualities in common:

- They are realistic.
- They are specific—addressing each one of the many possible operating situations.
- They involve the same people who operate IMS in actual, production-mode situations.
- Their results are verifiable.

Improvement through feedback

Although formal testing of your procedures ends when production begins, each actual instance of operation and recovery can be considered a test—a chance to improve your procedures. If problems develop or procedures are found to be vague or inaccurate, you should use this experience to update and improve your procedures.

You might also want to consider conducting periodic reviews of your procedures by your operations staff and system administration staff. Such reviews are particularly important as you add maintenance to your IMS system or upgrade to a new release of IMS.

Retesting when the system changes

Any time you significantly change your system, you should do at least some retesting. Changes could be in the form of either a new release or major upgrade of IMS itself, or a redesign or expansion of your production setup.

The retesting should verify that your old procedures still function properly in the new environment.

Related concepts

DL/I test program (DFSDDLT0) reference (Application Programming APIs)

Chapter 7. Developing user procedures

End users (remote terminal operators) are the people who use the terminals connected to IMS. End-user procedures should focus on the needs of end users, not on IMS operation.

It is important that your end-user procedures be good ones, because the success of your online system depends largely on its acceptance by your end users.

The procedures should be prepared as an integral part of the application-development process. The layout and organization of the procedures should support the business function. Procedures differ for different types of applications. Procedures for a bank teller, for example, should cover more than just the entry of an IMS transaction. Procedures for a data-entry clerk, however, may consist largely of instructions for entering transactions.

End-user procedures are generally packaged in a guide. You can develop the guide yourself, or it might be appropriate for individual application-development teams to develop the guide. In the latter case, you should consider preparing "boiler plate" material to distribute to the application-development teams for basic procedures that are always included in end-user guides. Do not forget to update the boiler plate information whenever you make major modifications to the online IMS system. For example, if you convert from using non-VTAM to using VTAM, you probably need to change terminal logon procedures.

In general, your procedures need to familiarize end users with how to:

- Operate their terminal
- Establish a connection to IMS
- Communicate with IMS
- Disconnect from IMS
- Respond to error conditions

Related concepts

“Operations personnel” on page 159
It is important that you identify the operational requirements for IMS, and identify the people responsible for performing the various operations tasks. After assigning responsibilities, you are ready to set up the procedures necessary to coordinate the operations task.

Procedures for user terminal operators

The application-development cycle should include the development of end-user procedures, along with education about the transactions and terminal hardware.

The writers of the procedures need to consider the following points when developing the end-user guide:

- Consistency between remote and central control
- Which services should be provided by the online system rather than the terminal or workstation
- Coordination of support for end-users
- Definition of problem-reporting procedures

Related concepts

“Planning the content of procedures” on page 163

Your procedures for the master terminal operator (MTO) will vary depending upon their knowledge of IMS operations and the application-program logic

User operator tasks

An end user performs one or more tasks as part of his or her job. You need to provide your end users with instructions that are clearly written and accurate for the task that they will perform.

These tasks can involve many of your IT tools and systems, of which IMS is one. You should review those tasks to be sure your procedural instructions do both of the following:

- Help the user perform application-related tasks
- Safeguard the primary task

You should review existing instructions that have implications for administration services. The following table shows a partial checklist illustrating the characteristics of this review. You should expect other topics depending on the scope of the user procedure.

Your review contributes to the usability of the procedures. You should plan to correct them based on testing and user feedback during actual production.

Table 39. Checklist for administrative services

Review item	Related administration task
Are availability statements clear?	Check scheduling algorithm and operations procedures.
Are response claims reasonable and not misleading?	Reconcile with performance criteria. Ensure unusual delays are reported.
Are terminal connection procedures described?	Check whether the MTO or other operators help in connection.
Are security requirements explained?	Check security definition and maintenance procedure.
Is interpretative material included for output?	Check accuracy of IMS message and command output information.
Are instructions given for an alternative connection?	Check that the MTO has alternative configuration instructions.
Are instructions given for response to system and broadcast messages?	Check with developer on potential messages.
Are instructions given for service problems for the terminal, transaction input, and online system failure?	Coordinate failure reporting and information flow. Have MTO issue standard information messages.

Related concepts

“Operating instructions for terminal operators” on page 215

The process and goals of developing end-user documentation include developing information in several stages and to tailor the documentation to the experience level and information requirements of the end users.

Operating instructions for terminal operators

The process and goals of developing end-user documentation include developing information in several stages and to tailor the documentation to the experience level and information requirements of the end users.

The success and acceptance of an online system often depends on the quality of the end-user documentation. The operating procedures for terminal operators form an important part of this documentation, contributing to ongoing education and ease of use for the applications.

Develop and refine the procedures in several stages:

- Define initial structure and content
- Analyze existing procedures
- Test the new procedures
- Gather education feedback
- Gather production feedback
- Make any necessary modifications to the procedures

It is generally better to extract information from existing sources and tailor it to an application rather than point to total descriptions or general operation manuals. For example, describe the 3270 operational characteristics rather than assume that the operator is familiar with the content of the 3270 Data Stream Programmer's Reference.

Also, show representative screen formats rather than showing MFS output. You can obtain working copies of the screen layouts by using the copy feature on a remote 3277 terminal.

Do not ignore the last stage, the actual production cycle. In many cases, you can obtain useful feedback by selecting a group of end users to be pilot operators. Their mission is to validate and improve the end-user service. After making any necessary changes, publish the final procedures.

Do not neglect to update the procedures when you make modifications to the online IMS system. Sometimes the operational emphasis changes when you install subsequent application programs or IMS features. Your same pilot group could renew their activities when hardware changes or other significant application changes occur.

Some guidelines for the content of end-user operator documents follow. The goal is to have the document tuned to the experience level and information requirements of the end users:

- Make the main section procedural for the applications.

  Ensure the instructions are sequenced in the order in which they are performed. Place transactions that are more frequently used, or more important, ahead of others. If there are numerous transactions, an alphabetic order might be better, if you include an introduction (or graphic) that indicates which are critical or used frequently.
- Illustrate your procedures with simple examples.

  Show both entry and response formats. Where IMS commands are used, show only relevant keywords and explain the replies.
- Include instructions for error handling.

  Address not only input errors but also responses to warning messages from the application program. Where IMS connection or system problems might occur, provide alternative instructions. If this material is lengthy, you could place it in an appendix. Arrange the topics in the appendix by symptom. Provide an explanation of error-reporting forms.
- Optionally, include a brief introduction to IMS and describe the MTO.

The emphasis should not be on how IMS works, but on how it supports data access and processing at your site. Distinguish, as necessary, between what IMS provides and what the operating system and hardware provide.

- Identify the main hardware operational characteristics.

  Include advice that highlights efficient ways of using the terminal and warnings of any user actions that are prone to error. If appropriate, merge copies of relevant operating instructions into this section or appended to the document.

- Include only relevant IMS command operation details.

  Make sure that commands and options described are allowed for end-user use. Discourage the use of commands that reserve IMS resources, such as **/HOLD** for conversations or **/IDLE** which enqueues output messages.

- Include a list of contacts.

  Show names in preferred order and describe the conditions under which the user should make contact.

Related concepts

"User operator tasks" on page 214
An end user performs one or more tasks as part of his or her job. You need to provide your end users with instructions that are clearly written and accurate for the task that they will perform.

Potential use of IMS commands

Although the primary purpose of end-user procedures is to describe transaction input, you might also allow the users to enter a subset of IMS commands so that they can make more informed use of IMS resources.

For example, if the end users can choose the order in which transactions are submitted, they can use a **/DISPLAY** command to gauge the current queue levels.

Recommendation: If you do allow end users to enter IMS commands, consider disallowing the use of the ALL keyword to minimize the effect of these commands.

The following table presents a list of IMS commands and evaluates their potential use by a remote terminal operator. The **/DISPLAY TRAN** and **QUERY TRAN** commands are especially appropriate. However, because of the many keyword choices for these commands, you should specify individual keywords. For convenient entry, include command abbreviations.

The commands in the table that are marked with an asterisk (*) are available to the remote terminal operator using default command security. To allow the remote terminal operator to use other commands, you need to declare the full list of commands for that terminal through RACF (or an equivalent product). You can also use the DFSCCMD0 exit routine for command security.

Table 40. Command suitability for use by a remote terminal operator

Command group	Command	Default security	End-user use	Comments
End-user specific	/RCLSDST	*	YES	Acts as remote VARY offline (for VTAM terminals).
	/RCOMPT	*	YES	Decentralizes control of VTAM component.
	/BROADCAST	*	YES	Provides convenient user interaction; use of message switching is preferred.
	/SET	*	YES	Provides convenient entry for batched transactions.
	/RESET	*	YES	Used with /SET.
	/RDISPLAY	*	YES	Identifies MTO location; used with / BROADCAST.
	/CANCEL	*	YES	Cancels both single- and multisegment messages.
	/EXCLUSIVE	*	YES	Safeguards discrete application usage.
Connection commands	/OPNDST		NO	Allows access to VTAM network.
	/CLSDST		NO	Restricts access to VTAM network.
	/QUIESCE		NO	Restricts access to VTAM network.
	/SIGN	*	YES	Allows access to IMS resources; required for signon verification.
	/RSTART		YES	Starts IMS resources; allowed if /STOP allowed.
	/MSASSIGN		NO	Alters assignments for IMS resources; required for MFS transactions.
	/FORMAT	*	YES	Formats a screen; required for conversation terminations.
	/EXIT	*	YES	Terminates conversations; required for non-VTAM switched terminals.
	/HOLD	*	YES	Suspends conversations.
	/RELEASE	*	YES	Resumes conversations.
Connection commands for MTO	/START		NO	Starts IMS resources.
	/STOP		NO	Stops IMS resources.
	/CHANGE		NO	Allows changes to mode table and automatic session restart. Immediate termination best handled by the MTO.
	/IDLE		NO	Allows control of multiple systems.
Device control	/COMPT		NO	Sets terminal components; also available through /RCOMPT.
	/MONITOR		NO	Stops programmable remote stations; best handled by MTO.

Table 40. Command suitability for use by a remote terminal operator (continued)

Command group	Command	Default security	End-user use	Comments
Monitoring	/DISPLAY		YES	Displays information about IMS resources; convenient for status of queues and terminals.
	/TRACE		NO	Allows IMS to trace resource activity; IMS Monitor and traces affect system performance.
	/LOG	*	YES	Records user text on IMS system log.
Serviceability	/DIAG		NO	Retrieves diagnostic information for system resources (for example, IMS control blocks) without taking console dump.
Testing	/TEST	*	YES	Sets test mode; limit use for remote terminal operators.
	/LOOPTEST	*	NO	Tests I/O for terminal.
	/END	*	YES	Terminates test, looptest, or exclusive mode.
	/MSVERIFY		NO	Verifies IMS resources; used by administrator of multiple systems.
Resource control	/ASSIGN		NO	Assigns IMS resources; prime control of resources for MTO.
	/PSTOP		NO	Restricts use of resources.
	/PURGE		NO	Restricts input to resources.
	/LOCK	*	NO	Restricts resource use; use only for supervisor remote terminal operators.
	/UNLOCK	*	NO	Unlocks IMS resources.
	/RMxxxxxx		NO	Allows control of DBRC resources; database availability should be controlled by MTO.
	/SMCOPY		NO	Controls the logging of system messages, commands, and command responses to the secondary master terminal.
	/MODIFY		NO	Allows online changes to IMS system.

Table 40. Command suitability for use by a remote terminal operator (continued)

Command group	Command	Default security	End-user use	Comments
Recovery	/DBRECOVERY or UPDATE		NO	Controls database availability; restrict to MTO.
	/DBDUMP or UPDATE		NO	Controls database availability; restrict to MTO.
	/DEQUEUE		NO	Controls integrity of message queues; restrict to MTO.
	/CHECKPOINT		NO	Shuts down IMS; restrict to MTO; simple checkpoints can have a performance impact.
	/ERESTART		NO	Restarts IMS; restrict to MTO.
	/RECOVER		NO	Recovers full-function databases and Fast Path areas; restrict to a DBA or the MTO.
/SWITCH			NO	Switches data sets used in an XRF environment; restrict to MTO.

Related concepts

"IMS commands used to communicate with IMS" on page 223

Although the primary purpose of an end-user guide is to describe transaction input, you might allow end users to use certain IMS commands. If you do, you should document their use in the guide.

Related reference

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

/DISPLAY TRAN command (Commands)

QUERY TRAN command (Commands)

Command Authorization exit routine (DFSCCMD0) (Exit Routines)

Problem reporting for a remote terminal operator

The main flow of the end-user guide should present the normal operations sequence, but the operator needs to be prepared for unexpected or error responses.

When you examine the sequence of instructions for a remote terminal operator, you need to ask a series of "what-if" questions. The following are examples of these "what-if" questions:

- What if the terminal is not operational, behaves erratically, or loses contact with the system?
- What if the application program returns warning messages about the status of a database or of processing results?
- What if IMS system messages are received?
- What if master terminal or other operator actions interrupt processing?
- What if the z/OS or IMS system goes down?

You need to assess the importance of such unusual events and include appropriate procedures for them. The end-user guide could describe recovery actions inline with normal operations or include the information in symptom-based appendixes.

If your installation has a problem-reporting center, having a user-liaison group enables you to concentrate recovery skills rather than educating all end users. Many installations report terminal problems to network control personnel, and they have appropriate details about IMS and the application programs it runs. It is often a good idea to standardize problem reporting. You can prepare the answers to a series of questions, or elicit a problem's symptoms using a structured script.

Occasionally, the master terminal operator can be a point of contact. You should ensure such contact is only be on a formal, prearranged basis, because the master terminal operator has a primary duty of system control and monitoring. An example of acceptable contact is to request the scheduling of a BMP.

A good controlling technique is to require an error incident report. You can make preformatted forms, with examples of the information required, available to the end user. The data on the forms should indicate what action was taken or is pending. Administration personnel can then review these events on a regular basis. In this way, you can detect common problems or differences in operational procedures.

Related concepts

"Administration support for errors" on page 225
Administrative support for end users can be provided in a variety of ways, one of which is to use a user-liaison group. By using a user-liaison group, you can concentrate recovery skills, rather than educating all end users.

Connecting to IMS

Before a user can use a terminal, the master terminal operator (MTO) must initialize the communication network using one or more of the following commands: **/START DC**, **/START NODE**, and (possibly) **/OPNDST** for VTAM terminals.

Terminal startup for nonswitched lines

For startup to be successful, the remote terminal must be turned on. IMS notifies the remote terminal of a successful **/START** or **UPDATE** command with the DFS059 terminal started message. Communication with IMS can only begin after the startup message is received.

Exception: A remote terminal can communicate with the following terminals before receiving message DFS059: the master terminal, physical terminals on the master terminal line, and the system console.

VTAM terminals (nodes)

SLU type P, FINANCE (3600), and ISC terminal systems do not receive IMS-generated connect messages. Instead, they communicate with a user-written program that resides in the terminal system's storage. Type-1 SLUs in unattended mode also do not receive connect messages (because there is no operator to receive the message).

VTAM terminals receive message DFS3649, DFS3650, or both. The DFS3649 message requests signon information, and the DFS3650 message indicates successful signon (or connection) and session status. Message DFS3649 also appears in the following situations: when failures occur during the signon process, when the operator signs off, or (for ETO only) automatic session timeout due to inactivity.

Releasing VTAM terminals hung in I/O

Use the **/TRACE** command to specify what action, if any, IMS should take when it determines that a VTAM response has been outstanding for a longer time than you specify. This facility applies only to VTAM devices that have VTCBs or MSC links.

IMS can handle outstanding responses in one of the following ways:

- No action.
- Issue a message that I/O has been outstanding for a longer time than you have specified; that is, the device has timed out.

  In this case, the operator must determine what action, if any, to take. For example, the MTO or an AOI program could issue an **/IDLE** command followed by an **/ACTIVATE** command and an **/OPNDST** command.
- Issue a message that I/O is outstanding followed by a VTAM **VARY NET,INACT** command to deactivate the terminal, and a VTAM **VARY NET,ACT** command to reactivate it. If the device is operable and defined to IMS as non-shared, and IMS is not shutting down, IMS issues a **/OPNDST** command to establish a session with the node.

Use the **/DISPLAY** command to display all the nodes that have I/O which has been outstanding for a longer time than was specified during installation. The **/DISPLAY TRACE** command displays the status of all nodes that have I/O outstanding for a longer time than you specified using the **/TRACE SET TIMEOUT** command.

Fulfilling security requirements

Depending on the security options specified for the system, the end user might need to use the **/SIGN** command to provide a user ID for verification by IMS or RACF and supply an IMS or RACF password. Users with ETO terminals must use the **/SIGN** command. You need to document the security procedures the end user must follow.

Related concepts

IMS terminal network (Communications and Connections)

Related reference

Terminals and equipment supported by IMS 15.4 (Release Planning)

Communicating with IMS

You can communicate with IMS using transactions, sending messages to other users, and using IMS commands.

Transactions for communicating with IMS

When using a terminal (a non-programmable workstation), the operator must enter a transaction in a certain format in order for it to be recognized by IMS.

When the operator uses the **/FORMAT** command, IMS formats the screen using MFS. This format is defined by the installation.

Transaction code format for entering transactions

Use the transaction code format shown below to send a message to an application program:

```
transaction_code     (password)    text
```

where:

transaction_code
Is a 1- to 8-byte alphanumeric code defined during IMS system definition. Special characters (embedded blank, slash, dash, equal sign, comma, and period) are not allowed. A blank or left parenthesis must follow the transaction code. You can use the COMM system definition macro to eliminate the need for a blank in a segment consisting of only the transaction code.

(*password*)
Is a password that must be entered with the transaction code (when required by installation security). Enclose the password in parentheses and concatenate it to the transaction code. If a password is not required, IMS ignores any password you enter.

For terminals using MFS, password entry is controlled by installation-defined formats.

text
Is the single-segment or multi-segment input message. Only the first segment of a multi-segment message can contain the transaction code and password.

Use the **/SET** command to send a continuous series of messages to the same application program. The **/SET** command puts the terminal in preset mode and eliminates the need to enter the transaction code (and password, if required) for each message. Preset mode fixes the destination for all messages entered from a terminal.

The text included in the transaction is application dependent, and should be clearly documented for each transaction type an end user might use.

Response mode for transactions

When the response mode is in effect, IMS does not accept input from the line, terminal, or user until IMS has sent an output response to the previous input. Response mode can be defined for either full function or Fast Path.

Response mode describes a connection between IMS and a communication line, terminal, or user that occurs for certain types of terminals or users under conditions specified during system definition.

You can specify communication lines, terminals, or users as:

- Always operating in response mode
- Never operating in response mode
- Operating in response mode only if the transaction code being processed specifies it

Any transaction can be defined as a response-mode transaction. When IMS receives an input transaction that requires response mode, IMS makes the terminal unusable (locks the keyboard) until IMS processes the transaction and sends the response.

While in response mode, if the user enters a single transaction for which the application program generates no response, IMS automatically sends message DFS2082 as the response to the transaction. This message removes the terminal from response mode and unlocks the keyboard, if necessary.

While in response mode, if the user enters multiple transactions for which the application program generates no response, the keyboard remains locked and the terminal is unusable. The terminal remains in response mode even if the user attempts to unlock the keyboard (by, for example, pressing the RESET key on a 3270 terminal); any subsequent input is rejected. The first output from an application program whose input originated from this terminal satisfies the response.

User IDs defined with ETO user descriptors remain in response mode after abnormal session termination or autosignoff. In this situation, the end user should contact the MTO (using a means other than the terminal or user ID). If the terminal is non-operational, the MTO can enter an **/RSTART** command to restore terminal operation.

Attention: If you restart IMS while users are in response mode, what happens to transactions depends on whether they have reached a sync point:

- If the transaction is complete (it has reached a sync point), and the end user is waiting for a response, the user does not receive the normal response because IMS restart resets response mode. After the restart completes, the user might reenter the transaction, which could adversely affect data integrity.
- If the transaction is incomplete (it has not reached a sync point), IMS discards the transaction.

You should develop operating procedures to cover this situation for all end users who could be affected.

To reset static nodes and ETO dynamic users that are hung in Fast Path input response mode, issue the **/STOP** and **/START** commands in sequence with the appropriate parameter. For static nodes, issue the commands **/STOP NODE** and **/START NODE**. For ETO dynamic users, issue the commands **/STOP USER** and **/START USER**.

Conversation mode for transactions

Conversational processing allows an end user to have a continuing dialogue with an application program.

A user initiates a conversation by entering a transaction code that is defined as conversational. A conversation can follow an ETO user from one terminal to another. The maximum number of conversations that can be active for one user or one terminal at a time is 65535.

The following commands affect conversations:

- **/DISPLAY** displays the status of conversational processing in the system

- **/EXIT** terminates an active or suspended conversation
- **/HOLD** suspends an active conversation
- **/RELEASE** restarts a held conversation
- **/RSTART LINE | NODE | USER** recovers an active or suspended conversation on a stopped line, node, or user
- **/START LINE | USER** restarts the line or user but terminates the conversation

Using the appropriate keywords of the **/DISPLAY** command, the MTO can determine:

- The identification of all active (BUSY) or suspended (HELD) conversations in the system
- The number of conversations active or suspended in the system or related to a specific line, line and physical terminal, or user
- The number of inactive available conversations (the maximum number of conversations defined for the system minus the number of currently active or suspended conversations)

When you use the **/DISPLAY** command to display the status of a line or physical terminal, the displayed status includes, if applicable, the status of conversations related to that line.

Operators-to-operator messages

An end user can send messages to other operators by using the IMS message switch facility or by using the **/BROADCAST** command.

If your end users need to send messages, document clearly how they are to do this.

When using message switching, the input format is as follows:

```
logical_terminal_name     text
```

where:

logical_terminal_name
Is the name of the logical terminal to which the message is to be sent. The logical terminal names allowed are specified during IMS system definition.

A blank must follow the logical terminal name. You can use the system definition COMM macro to eliminate the need for a blank in a segment consisting of only the logical terminal name.

text
Is the single-segment or multi-segment input message

For a multi-segment message, only the first segment should contain the logical terminal name. If the logical terminal name is specified in subsequent segments, it is treated as text.

You can use the **/SET** command to send a continuous series of messages to the same logical terminal. The **/SET** command puts the terminal in preset mode and eliminates the need to enter the logical terminal name for each message. Preset mode fixes the destination for all messages entered from a terminal.

IMS commands used to communicate with IMS

Although the primary purpose of an end-user guide is to describe transaction input, you might allow end users to use certain IMS commands. If you do, you should document their use in the guide.

Related concepts

“Potential use of IMS commands” on page 216

Although the primary purpose of end-user procedures is to describe transaction input, you might also allow the users to enter a subset of IMS commands so that they can make more informed use of IMS resources.

Commands for special operating modes

IMS provides commands to enter and exit special operating modes, such as preset mode, test mode, MFSTEST mode, looptest mode, and exclusive mode. These modes alter the relationship between the terminal and the system.

A special operating mode remains active until one of the following events occur:

- The user issues a command to exit from the special operating mode
- The MTO stops and restarts the line, physical terminal, or user by issuing the **/START** command

To restart a terminal or user without affecting its special operating mode, use the **/RSTART** command.

Preset mode

Preset mode fixes the destination for all messages entered from a terminal. Use the **/SET** command to enter preset mode. To leave preset mode, use the **/RESET** command.

Test mode

Test mode (also called "echo" mode) performs input and output operations between a terminal and IMS. The **/TEST** command places a terminal in test mode. Messages entered from a terminal in test mode are transmitted back to that terminal. The **/END** command terminates test mode. In an ETO environment, the **/TEST** command applies only to ETO-defined users, not to ETO-defined terminals.

MFSTEST mode

MFSTEST mode tests MFS control blocks without interrupting normal production activity. The **/TEST MFS** command places MFS-supported terminals in MFSTEST mode. The **/END** command terminates MFSTEST mode. In an ETO environment, the **/TEST** command applies only to ETO-defined users, not to ETO-defined terminals.

Looptest mode

Looptest mode tests output operations between a terminal and IMS. Use the **/LOOPTEST** command to enter this mode. This command creates an output write loop that repeatedly transmits a single-segment message to the terminal being tested. The **/END** command terminates looptest mode.

Exclusive mode

Exclusive mode prevents output messages from being transmitted to the terminal.

Exception: Output messages that are responses to messages entered at the terminal during exclusive mode are transmitted to the terminal.

After the terminal is removed from exclusive mode, the output messages are held for transmission. Use the **/EXCLUSIVE** command to enter exclusive mode, and the **/END** to leave exclusive mode. In an ETO environment, the **/EXCLUSIVE** command applies only to ETO-defined users, not to ETO-defined terminals.

Disconnecting from IMS

If an ETO-defined terminal or user ID is idle (no transactions are performed) for a given amount of time, IMS automatically disconnects the terminal or user ID. This amount of time is determined by the MTO. The end user can sign back on by issuing the **/SIGN** command.

Related tasks

Autosignoff (ASOT) (Communications and Connections)
Autologoff (ALOT) (Communications and Connections)

Administration support for errors

Administrative support for end users can be provided in a variety of ways, one of which is to use a user-liaison group. By using a user-liaison group, you can concentrate recovery skills, rather than educating all end users.

It is a good idea to standardize end-user problem reporting. You can have the end user answer a series of questions before calling the user-liaison group, or you can have the user-liaison group elicit problem symptoms from the end user using a structured script. Among other things, each report should include which system and terminal experienced the error.

No matter how you decide to provide administrative support, the end-user guide should include a contacts list. This list identifies the people to be called for application, system, or terminal errors. Names, functions, and telephone numbers should be included in this list.

Related concepts

“Problem reporting for a remote terminal operator” on page 219
The main flow of the end-user guide should present the normal operations sequence, but the operator needs to be prepared for unexpected or error responses.

Chapter 8. Operations and IMS-supported devices

This topic describes the devices that IMS supports from an operations perspective. This information should help you develop those parts of your end-user procedures that are device related.

This topic assumes that you are familiar with the general features and operation of each device. The focus is on device operation when you use IMS.

3270 Information Display System

During IMS system definition, you can specify that a 3270 system attached to IMS should operate in forced-response mode, nonresponse mode, or transaction-dependent response mode. For an Extended Terminal Option (ETO) terminal, specify these modes in the ETO user descriptor.

When 3270 terminals are not attached to IMS through VTAM , the MTO can initiate their use by entering the appropriate **/START LINE** command for the lines to which the 3270 components are attached.

When 3270 terminals are attached to IMS through VTAM, they are initiated when one of the following occurs:

- The z/OS VTAM network operator enters a VARY NET command (for example, VARY NET,ID=*a*,LOGON=*b*).
- The IMS MTO enters the /OPNDST command.
- If the 3270 terminal components are defined in the VTAM definition as belonging to IMS, VTAM makes them automatically available.
- The 3270 terminal operator enters a request to VTAM to be connected to IMS.

Each operational terminal receives a DFS059I message indicating that the terminal has been started. The MTO can use a **/DISPLAY** command to determine which lines are attached.

Interacting with IMS

If you enter data with a keyboard, a selector light pen, a magnetic card reader, or program function key special features, use the **/FORMAT** command to display a format on the screen before you begin working with IMS.

About this task

After you receive either message DFS059I or DFS3650I telling you that the terminal or session is started, you can enter a transaction code, a command, or an IMS message switch.

When you enter a transaction code, IMS returns and displays the output data from the application program that processes that transaction. This output might include a prompt to solicit additional input. You can fill in only the blank fields on the screen.

After entering data, press Enter to send it to IMS for processing by the application program. Continue in this manner until the interaction with IMS or the application program is complete.

3270 terminal components that operate with IMS

You must understand the various functions of 3270 terminal to work with IMS.

Terminal display size

The 3270 display station can display from 480 to 3564 characters of information, depending on the model. These characters can be alphanumeric characters, special characters, certain device-control characters (visible on the screen), and attribute bytes (not visible on the screen). In addition, you can prevent display of information in certain fields on the screen.

The 3275 or 3277 terminal can display:

- 480 characters (Model 1)
- 1920 characters (Model 2)

The 3276 or 3278 terminal can display:

- 480 or 960 characters (Model 1)
- 1920 characters (Model 2)
- 1920 or 2560 characters (Model 3)
- 1920 or 3440 characters (Model 4)
- 3564 characters (Model 5)

The 3279 terminal can display:

- 1920 characters in 4 colors (Model 2A)
- 2560 characters in 4 colors (Model 3A)
- 1920 characters in 7 colors (Model 2B)
- 2560 characters in 7 colors (Model 3B)

3270 terminal page-advance function

A transaction's output can be more than one full screen of data. You request the additional output by pressing the PA1 (program access 1) key on the terminal keyboard. If you continue to press the PA1 key, the terminal displays successive pages until you reach the end of the current message. If you press PA1 when there are no more messages waiting to be sent to the display station, the display remains unchanged or IMS sends an appropriate error message.

3270 terminal message-advance function

If you press the PA2 key at any time during a message, IMS deletes the current output message and displays the next message. If no messages are waiting when you press the PA2 key, IMS does not modify the screen.

How to enter passwords from 3270 through MFS

Entering a password through Message Format Service (MFS) can be different from password entry on a cleared screen or on other IMS terminals.

The password can be created from:

- Data read from a 3270 ID card reader
- Data that you key into one or more fields or a portions of fields in the device format
- Data that you key into a field that is not displayed on the screen and is designated as the password field only in the device format
- Literals
- Combinations of the above

Terminal operator logical paging

If you do not specify operator logical paging, the terminal operator can advance pages only one at a time, by using the page advance function. If you do specify operator logical paging in the application program's message output description, the terminal operator can page forward or backward to any page in a message.

With operator logical paging, the terminal operator can enter any of the functions shown in the following table to display a particular logical page.

Table 41. Functions for Operator Logical Paging

Enter	To display
=	The next logical page
=*n* =*nn* =*nnn* =*nnnn*	The *n*th logical page of the message The maximum value for *nnnn* is 9999.
=+*n* =+*nn* =+*nnn*	The *n*th logical page after the current page The maximum value for *nnn* is 999.
=-*n* =-*nn* =-*nnn* =>*nnn*	The *n*th logical page before the current page The maximum value for *nnn* is 999.
=*L*	The last logical page of the message

Enter these functions in the screen area reserved for command and page request information. If no area is reserved, either clear the screen (using the Clear key) and then enter the paging request, or use the next logical page (NEXTLP) function if it has been defined as a PF key in the format definition.

If you request a page outside the range of the pages in the message, IMS issues an error message. After the error message displays, press the PA2 key to continue the message that was being displayed when the error occurred. If a logical page in the message is larger than one physical page, use the PA1 key to step from one physical page to the next within a logical page.

Modified data tags and the 3270 keyboard

IMS resets modified data tags each time it unlocks the 3270 keyboard.

Exception: IMS does not reset modified data tags when:

- IMS sends a message using the SYSMSG field.
- IMS sends output to its master terminal using the Message Format Service (MFS) master-terminal format. IMS resets modified data tags only after you enter input.

3270 terminal screen protection

Under certain conditions, IMS avoids sending output to a 3270 terminal that could possibly destroy the operator's current work. IMS protects the screen so that you can safely enter data or control requests.

You can:

- Set the screen mode to protect and:
  - IMS successfully sends output to the screen.
  - You press the Clear key and IMS responds by unlocking the keyboard.
  - You or the application program perform a successful Copy function.
- Requested the NEXTMSGP function (the PA3 key or a PF key so defined) when no output message was available.

The screen is not protected after you:

- Press one of the following keys:
  - Enter

  - PA1 or PA2
  - Any PF key (except a PF key defined as NEXTMSGP)
- Operate either of the following devices:
  - The selector light pen
  - The ID card reader

Attention: If the screen is protected, an IMS shutdown checkpoint cannot complete.

During IMS system definition (using the TERMINAL and TYPE macros or ETO logon descriptors) you can specify that a 3270 screen be left in unprotected mode when IMS sends an output message to it. In unprotected mode, IMS can send output to the device at any time without requiring input from the terminal (an input message or a PA1 or PA2 action). IMS provides this option on a terminal-by-terminal or message-by-message basis. The terminal option of unprotected mode applies to all messages for that terminal. If you do not select unprotected mode, MFS-formatted messages leave the screen protected or unprotected on a message-by-message basis, and messages that bypass MFS leave the screen protected.

By allowing the terminal to be unprotected, IMS can continually update the 3270 display terminal with current information without requiring the terminal operator to initiate any action to receive output messages. Use this option only with terminals that require limited input and require the most current IMS output message.

To avoid losing data, press the PA3 key (if the PA3 key is not used for copy) before entering data. When you press the PA3 key, you put the terminal in protected mode, and you can enter data. After you enter the data, IMS returns the terminal to unprotected mode.

3270 terminal lock and unlock option

During IMS system definition, by using the TERMINAL and TYPE macros or Extended Terminal Option (ETO) logon descriptors, you can specify whether the application program or IMS should be responsible for unlocking the terminal keyboard and resetting the modified data tags.

If you select LOCK, the application program unlocks the keyboard after it receives input. If you select UNLOCK, IMS controls the unlocking of the keyboard. This option only applies to messages that bypass Message Format Service (MFS) on output with a MOD name of DFS.EDTN.

If the application program terminates without unlocking the keyboard, you can unlock the keyboard by pressing the Reset key.

Attention: If the application has not terminated but is slow in responding, do not press the Reset key and start entering data, because the data might be lost when the application program sends another output message to the terminal.

3270 MFS bypass option

If you select the 3270 Message Format Service (MFS) bypass option (by using a MOD name of DFS.EDTN), the application program controls the functions performed by the Clear key, selector pen attention, PA1, PA2, or PA3 keys.

3270 MFS protected field validation

Set MFSPFV=Y in the DFSDCxxx member of the IMS PROCLIB data set to configure the 3270 MFS protected field validation option, which verifies whether the content of a protected field from the input is different from the original content sent out to the 3270 device.

3270 terminal programmed symbols

Programmed symbols (PSs), an optional feature of the 3270 terminal, allow storage and access of up to six sets of 190 characters, each of which can be user-defined and loaded by the program. You can load PS buffers with either a VTAM application or a user-defined procedure. After the PS buffers are loaded, you can use them on a field-by-field basis on the screens formatted by the Message Format Service (MFS).

If a hardware error occurs while you are loading a PS buffer, IMS:

1. Returns the message used to load the PSs to the IMS message queue.
2. Takes the terminal out of service (except for SLU-2 devices).
3. Logs the error to the IMS log.
4. Sends a message to the IMS master terminal.

After you correct the hardware error and the terminal is in service, IMS re-sends the message used to load the PSs.

If the PS load function failed because of an error in the message used to load them:

1. Use the **/DEQUEUE** command to dequeue the message. You might have to enter the **/DEQUEUE** command from the master terminal.
2. Correct the error.
3. Reenter the transaction to send the message to reload the PSs.

For SLU-2 devices, if you have not loaded the PS buffers and IMS sends a message requiring the use of a PS buffer, the 3270 terminal rejects the message and IMS:

1. Returns the invalid message to the IMS queue.
2. Logs the error to the IMS log.
3. Sends a DFS2078I message to the node, telling you that the output was rejected. However, if the message being rejected is in storage, IMS sends the DFS2078I message to the IMS master terminal and closes the node.

IMS leaves the terminal in protected mode after it receives the DFS2078I message.

3270 terminal screen formats

The 3270 can operate with either an unformatted or a formatted screen. Additionally, there are also input formats supplied by IMS, and installation-defined formats.

Unformatted mode

The screen is in unformatted mode:

- When you turn on the display station
- Whenever you press the Clear key on the keyboard

While the screen is in unformatted mode, you can enter the following types of messages:

- IMS commands
- Paging requests (when a message using logical paging is in progress)
- Transaction or message switch data that does not require Message Format Service (MFS)

Formatted mode

IMS MFS defines the 3270 screen to have various fields with predetermined uses. The layout of the fields on the screen and the data displayed depend on the application program. A format consists of all the fields on a screen, and each message sent to a 3270 terminal must indicate a specific format.

An application program might use several types of formats. The format on the screen can change between messages, between input and output sequences of a message, or between transactions.

Different formats define input areas at different locations on the screen. When the 3270 terminal receives a message, the format usually places the 3270 cursor at the beginning of the first input area. The input can consist of commands, transactions, or message switches.

IMS-supplied input formats

Some IMS-supplied formats allow one or two segments of input from the 3270 screen (one segment equals one line on the screen). When you want more than two segments of input, you can use the **/ FORMAT** command to specify the DFSM04 format. This format is an IMS-supplied format that allows up to four segments of input data. Each segment in the format consists of only the data entered without new line characters or other cursor-control characters.

Installation-defined formats

The installation can develop a variety of formats. Each format definition determines the information that you can enter and the way that you can use the format.

Installation-defined formats allow any of the following (either single-segment or multi-segment):

- Transactions
- Page requests
- Commands

If the current installation-defined format does not allow the necessary input, try using the page-advance function (PA1), the message-advance function (PA2), or the Clear key to produce a screen format that allows entry of the desired data or command. Remember that using PA2 discards the remaining portion of the current output message.

Related concepts

“Operation using the MFS master-terminal formatting option” on page 235
When the IMS master terminal is a 3270 device, IMS displays system messages either with IMS-supplied formats or with the MFS master-terminal format

Multiple physical page input

An input message can consist of more than one physical page. For example, a 3275 or 3277 Model 1 display (480 characters) requires four physical pages for an input message designed to be equal to the 3275 or 3277 Model 2 screen size (1920 characters).

Use the following operational sequence as a guide when preparing procedures for 3270 operators who use Message Format Service (MFS) and enter multiple physical pages of data to create one message.

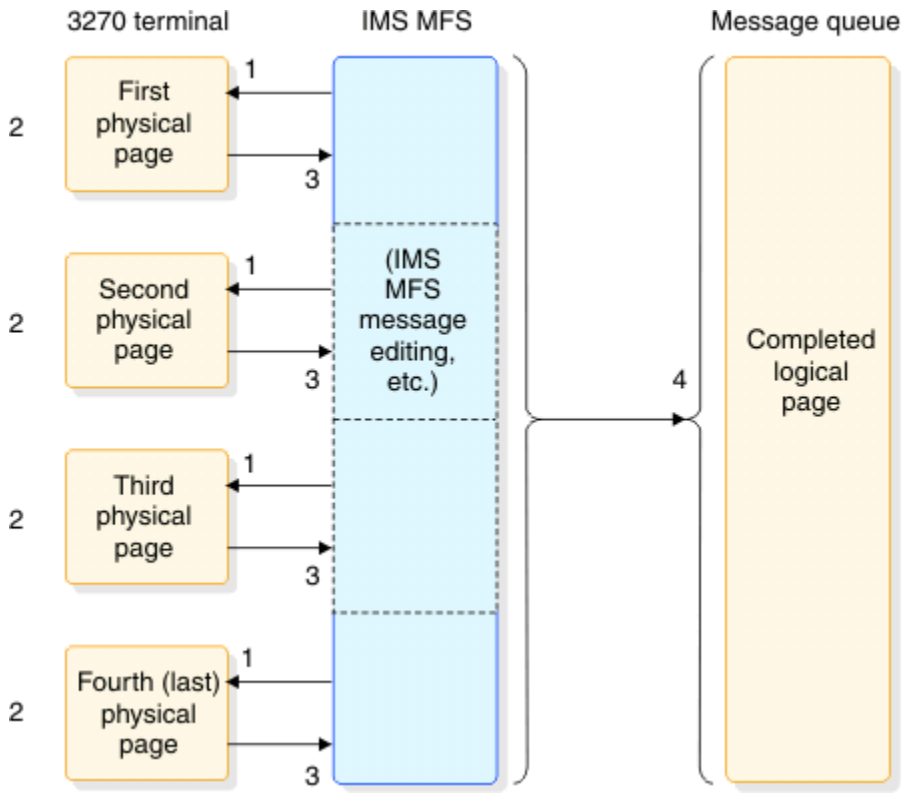


Figure 11. Data entry sequence for multiple physical pages

Note: The following notes describe the processing that occurs in the previous figure.

1. IMS sends the data-entry format for the physical pages to the 3270 screen.
2. The 3270 terminal is ready to accept your input whenever it displays the format of a physical page. You can input data (to start or continue creating a message) or use control functions.

   While the 3270 terminal displays the first physical page of an output message, your action determines what operation IMS performs:

   a. You can enter a message or choose not to enter a message.

      - When the 3270 terminal displays the first physical page and before you signal data entry, the following control functions are valid:
        - The PA1, PA2, and PA3 keys
        - The Clear key
        - The NEXTPP, NEXTMSG, NEXTMSGP, and PAGEREQ functions

        After you signal data entry, the rules for normal message creation apply.

   b. When you do not signal data entry on the first physical page, IMS assumes a display-only operation. When the 3270 terminal is displaying the first physical page, MFS can create an input message only if an output message is still available. An output message is not available if your previous request dequeued the current message and no other messages are enqueued. When IMS displays any page other than the first physical page on a display-only operation, only the following control functions are valid:

      - The Clear key
      - The PA1, PA2, and PA3 keys

      Attempting any other function results in an error message.

   c. You can create an input message and call for entry of that data:

Enter the required data into the physical page format displayed on the screen and then press Enter. Pressing a PF key that has not been assigned another specific function also causes the data to be entered.

You may want to advance to the next physical page without entering data from the currently displayed page. When the displayed physical page is not the first physical page, press the PA1 key.

d. You can signal the end of the current input message by entering the ENDMPPI function.

MFS edits the data keyed onto the screen and then performs end-of-message procedures to send the message to the message queue.

e. You can cancel the current input message by pressing one of the Clear, PA2, or PA3 keys, or by entering one of the NEXTLP, NEXTMSGP, or PAGEREQ functions.

IMS performs the respective function and discards all previous input for the current message.

3. You signal that the data on the screen is to be entered by pressing the Enter key.

MFS accepts the data from the 3270 terminal and performs any message editing that is required (inserting fill characters, literals, and so forth).

4. MFS sends the completed message to the IMS message queue.

You signal the end of the current message by entering the ENDMPPI function. MFS performs all required message editing and end-of-message procedures.

Additional rules that you must be aware of when you operate a 3270:

- You can enter the following from any physical page, but only once per message:
  - A PFK literal
  - An immediate pen-detect literal
  - An Op-ID card
- If you define DFLDs (device fields) with the same label on more than one physical page, MFS accepts data from only the first one defined as a part of the message.

3270 master terminal support

A 3270 terminal can be the IMS master terminal. When the IMS master terminal is a 3270 terminal, IMS sends all system messages to the primary device. IMS also sends selected system-generated messages critical to IMS operation to the secondary device.

A 3270 master terminal consists of two 3270 devices: a 3270 display (3275, 3276, 3277, or 3278) and a 3270 printer (3284, 3286, 3287, 3288, or 3289). IMS defines the 3270 display as the primary master terminal and the 3270 printer as the secondary master terminal. IMS does not support the 3284 Model 3 as a master terminal.

IMS responds only to the primary device in the following situations:

- When you enter erroneous information, such as a nonexistent transaction code or an invalid parameter, from the master terminal
- When another operator switches a message specifically to the primary device
- When an application program inserts a message to the primary device
- When IMS issues a command-completed message or output from a **/DISPLAY** command

If the secondary device is non-operational, IMS saves enqueued messages for transmission when it becomes operational.

The **/ASSIGN** command allows independent assignment of the 3270 display and 3270 printer.

Related reference

COMM macro (System Definition)

Operation using the MFS master-terminal formatting option

When the IMS master terminal is a 3270 device, IMS displays system messages either with IMS-supplied formats or with the MFS master-terminal format

The MFS master-terminal format is optional and is not available for all 3270 display devices. Choosing a 3270 display with 24 lines by 80 columns (a 1920 character display such as a 3275, 3276, 3277, or 3278 Model 2) as the IMS master terminal allows use of the MFS master-terminal formatting option.

When you use this option, MFS divides the 3270 master-terminal screen into four areas: a message area, a display area, a warning area, and a user input area. The following figure shows the relationship of the areas on the screen, followed by a description of each area.

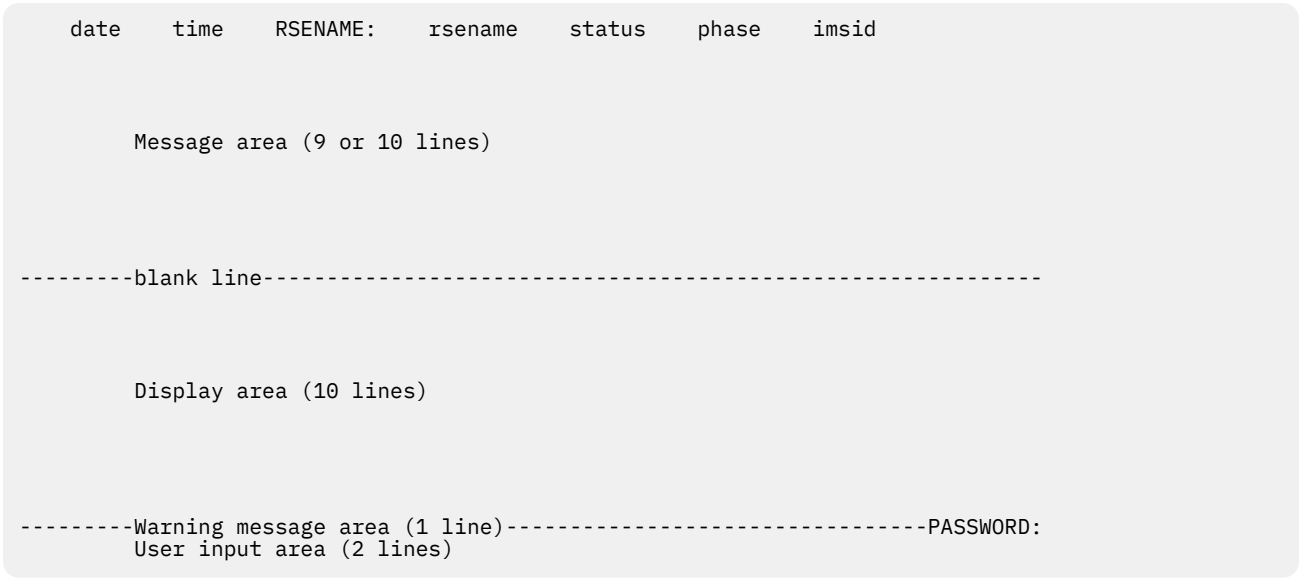


Figure 12. MFS screen format of 3270 (24-line, 80-column) master terminal

Related concepts

“3270 terminal screen formats” on page 231

The 3270 can operate with either an unformatted or a formatted screen. Additionally, there are also input formats supplied by IMS, and installation-defined formats.

Status line

If the IMS subsystem is XRF capable, the top line of the screen displays fields in high intensity.

date
The current date

time
The current time

RSENAME: *rsename*
The name specified for the RSENAME parameter in the DFSHSBxx member of IMS.PROCLIB

status
The current status, which can be null, ACTIVE, or BACKUP

phase
The current XRF phase, which can be any of the following:

Null status	ACTIVE status	BACKUP status
INITIALIZATION	null AWAITING I/O PREVENTION	AWAITING SNAPQ SYNCHRONIZATION TRACKING IN PROGRESS TAKEOVER REQUESTED TAKEOVER IN PROGRESS

imsid
The IMS ID of the active subsystem

Message area
Message Format Service (MFS) designates the first ten lines of the screen as the message area if the subsystem is not Extended Recovery Facility (XRF) capable, or the first nine lines of the screen under the status line, if it is XRF capable. When you signal readiness to receive output, IMS displays the new data on the top line in the area, overriding the previously displayed data, and inserts a blank line following the new line to separate it from the old messages.

These lines receive such messages as:

- Responses to **IMS** commands (except **/DISPLAY**, **QUERY**, and **/RDISPLAY**)
- Messages sent from other terminals or applications
- IMS system messages

When the message area is full and more data is ready for output, MFS sends a message in the warning message area.

Display area
Message Format Service (MFS) displays the output from the **/DISPLAY**, **QUERY**, and **/RDISPLAY** commands here, one page at a time. Each new page of the displayed data completely replaces the previously displayed page.

The ten lines below the message area are the display area.

Warning message area
The warning message area contains a literal, PASSWORD, followed by a field in which you can enter an IMS password. To enter a password, position the cursor immediately after the colon (:). You can press the back tab key to move the cursor from the user input area to the password field.

Line 22 is the warning message area. The four possible warning messages issued by Message Format Service (MFS) are:

MASTER LINES WAITING
Displayed when you receive a multisegment message that is larger than the currently available message area, or when the message area is full and additional lines need to be displayed. To display the remaining lines, press the PA1 key.

MASTER MESSAGE WAITING
Displayed when the message area is full and you receive a message from another terminal or from an application program. To display the message and reset the warning, press the PA1 key.

DISPLAY LINES WAITING
Displayed after the display area is full because the output of a /DISPLAY, QUERY, or /RDISPLAY command is more than 10 lines. To continue displaying the output, press the PA1 key.

USER MESSAGE WAITING
Displayed when you receive a message for the master terminal that is not formatted in one of the IMS default formats. To display the message (in its format), press the PA1 key. To return the screen to the master format, enter any IMS command, or place the terminal in unprotected status and wait to receive a message in the master format.

User-input Area
The two bottom lines (lines 23 and 24) on the screen are the user-input area. You can enter up to two 79-character segments of input. When you are not entering data, the cursor is located at the beginning of this area (column 2 of line 23).

Message Format Service (MFS) returns the cursor to this location when a response is received for the entered data. MFS leaves the entered data on the screen and you overlay it with your next input.

A 3270 master terminal using the MFS master-terminal formatting option is normally in unprotected status. This means that IMS can send output to the terminal at any time. You can put the terminal in protected status by pressing the PA3 key or the PFK11 key before entering input data.

The MFS-supplied master-terminal format defines literals for nine of the 3270 program function (PF) keys. The PF keys, when used with the 3270 master terminal formatting option, generate specific IMS commands. Pressing a PF key inserts a command into the first segment, in front of the data. The commands associated with the nine PF keys are:

PF Key
Command/Function

1
/DISPLAY
2
c
3
/DISPLAY STATUS
4
/START LINE
5
/STOP LINE
6
/DISPLAY POOL
7
/BROADCAST LTERM ALL
11
NEXTMSGP
12
Copy (3270 remote)

When pressing the PF7 key to enter the **/BROADCAST LTERM ALL** command, make sure the first line of input in the user-input area contains only the command. The second line of input should contain the data to be broadcast.

When you use the master-terminal format, MFS displays any message whose MOD name begins with DFSM0 (except DFSM03) in the message area. MFS displays any message whose MOD name is DFSDSP01 in the display area. Messages with other MOD names cause the warning message USER MESSAGE WAITING.

3270 terminal copy function

You can use the IMS Copy function to obtain a printed copy of the currently displayed screen on a remote terminal. You can define the Copy function for one or more terminals on a remote 3270 line.

You or an application program can use the Copy function through the use of the system control-area field. If you want to copy a screen from a display station, and the Copy function is allowed for your terminal, press the PF12 key.

The rules that govern the Copy function are:

In the SNA environment:

For SNA-defined nodes (in other words, SLU 2), IMS sends a copy request to the display node. The controller that selects the printer processes the copy command. The printer need not be defined to IMS.

In the non-SNA or bisync environment:

- If you request a copy from a 3275 display station, and a 3284 Model 3 printer is attached, the 3284-3 produces the hard copy.
- If you attempt a copy from a 3275 display station, and the 3284-3 printer is not ready, IMS issues an appropriate error message.
- If you request a copy from a 3277 display that is attached to a 3271 or 3274 control unit, or from a 3278 display attached to a 3274 control unit, or from a 3276 display (Models 1 through 4), IMS selects one of the following printers to receive the listing:
  - The printer that, during IMS system definition, you selected as eligible to receive copies from a specific 3270 display.
  - The first available printer of a group of 3270 printers that, during IMS system definition, you selected as eligible to receive copies from a specific 3270 display. If all printers are currently busy, the keyboard on the display station remains locked until a printer that can be used for the copy becomes free.
- If you attempt a copy from a 3270 display station attached to a 3271 or 3274 control unit or from a 3276 display station and no printers are currently available (all printers are stopped, pstopped, in exclusive status, or require intervention), IMS issues an appropriate error message.
- If you want all copy-function listings to be directed to a specific printer, you can use the **/STOP** command to prohibit the use of the other printers, or you can turn off the other printers.

  Restriction: The IMS copy function in the non-SNA/Bisync environment is not supported for ETO terminals.

To terminate a Copy function before it has completed, press the Reset key followed by either the Clear key or the Enter key. IMS terminates the copy function and unlocks the keyboard.

3270 terminal function keys

Certain keys on the display station keyboard are reserved for use by IMS. If you press any of these keys (or the selector light pen or operator ID reader) when the terminal is not screen protected, IMS might not be able to recognize your action.

Recommendation: Use these keys only when the screen is protected from receiving output.

These keys and their functions are:

PA1 — Page advance
The page advance key requests IMS to display the next screen of the current output message. The next screen can be the next physical page or the next logical page. The page-advance function is the only method of advancing from one physical page to the next physical page when a logical page consists of more than one physical page.

PA2 — Message advance
The message-advance key requests IMS to display the next message enqueued for the terminal. IMS deletes the current message, even if you have not displayed every page of it. If no more messages are waiting for the terminal, IMS leaves the current message on the screen.

PA3 — NEXTMSGP
The PA3 key usually performs the NEXTMSGP (message-advance protect) function, but could instead perform the Copy function.

PF12 — Copy
PF12 performs the Copy function for terminals that allow this function. The Copy function causes IMS to print the current display. IMS does not restore keyboard operation until it has either transferred the message to the 3270 printer (3276, 3277, or 3278) or completed the printing (3275). After a copy has completed on a 3275, reposition the cursor to an appropriate location before attempting any input.

Clear Key
To set the display screen to unformatted mode, press the Clear key once. While the screen is in unformatted mode, it is protected from receiving other output until you perform an input operation.

If IMS currently displays an output message, pressing the Clear key does not dequeue the output message.

If you press the Clear key when the terminal is not screen protected, IMS might not recognize the Clear request. However, if IMS does recognize the request, it unlocks the keyboard, displaying no data. If you press Clear and IMS responds by displaying data, the screen is invalid for input. You must press the Clear key again.

Enter Key
The Enter key sends to IMS either the fields currently modified by the operator or the last output message. The keyboard remains locked until IMS receives the message and restores keyboard operation.

3284 Model 3 printer operation

When attached to a 3275 terminal, IMS supports the 3284-3 printer as a component of the 3275 terminal. IMS considers the display portion of the terminal to be component 1 (COMPT 1) and the printer portion to be component 2 (COMPT 2).

As with any component-type terminal, IMS sends messages to the two components on a rotating basis.

If IMS cannot send messages to the printer component, it sends messages continuously to the display component, as if no printer component existed. The operator must intervene to allow printer output.

If IMS cannot send messages to the display component, it sends messages to the printer as if the display component did not exist. As long as IMS can send messages to the printer component, no operator intervention is required.

Any situation (such as a stopped LTERM or non-operational printer) that prevents sending messages to the LTERM or LTERMs assigned to a particular component causes IMS to stop sending messages to that component.

To suspend messages to either the printer or the display component, use the **/RCOMPT** command with the NOTRDY keyword. This command sets the component that you specify as non-operational (not ready) if the 3270 is attached using VTAM.

Using programmed symbols for IBM 3270

Programmed Symbols (PS), an optional feature on the IBM 3270 terminal, lets you store and access up to six character sets of 190 characters each that are user-definable and program-loadable.

About this task

After the coded graphic character sets have been loaded into the PS buffers, they can be used on a field-by-field basis by using Message Format Service (MFS).

If the PS buffers are desired and have not been loaded, they must be loaded by a VTAM application or by some other means.

Determining whether the buffers are loaded

If you know that the buffers have already been loaded with the desired programmed symbols, you do not need to resend the entire load programmed symbols data stream.

About this task

To determine if the buffers are loaded with the desired programmed symbols, use one of these methods:

- Maintain a handwritten log at the device and record the current status of the PS buffers.

- Run a test transaction that attempts to use the desired programmed symbols. Either the desired transaction's output is successfully displayed or an error message (DFS2078) is sent to the terminal and the message is returned to the IMS message queue, with the terminal left in protected mode. The message must be dequeued and a transaction started to load the programmed symbols buffer.

Loading the PS buffer

If the operator knows the PS buffers need to be loaded (because the device was just powered on), enter a response-mode transaction that loads programmed symbols by using an installation-written application program that loads PS buffers.

About this task

The load PS data stream should be the first part of the message sent by the application program and a notification, such as `THE PROGRAMMED SYMBOLS LOAD FOR psname COMPLETE` should be the remainder. The notification informs the terminal operator when the programmed symbols have been loaded. This transaction requires the Message Format Service (MFS) bypass option.

When the output device for which the PS buffers are to be loaded includes a printer or a different display, another technique must be used.

Example: These events illustrate one example of using an automated operator application program:

Procedure

1. The master terminal operator at `Display_A` enters a transaction (response or conversational) requesting that programmed symbols be loaded for `Display_A`, `Printer_B`, and `Display_C`.
2. In processing the transaction, an AOI program assigns LTERMs for `Printer_B` and `Display_C`, temporarily, to a special PTERM reserved for loading the PS buffers. The AOI program assigns dummy LTERMs to `Printer_B` and `Display_C`.
3. The AOI program inserts the message containing the load programmed symbols messages to the dummy LTERMs of `Printer_B` and `Display_C`.
4. The AOI program sends a programmed symbols message to `Display_A`.
5. The master terminal operator visually verifies messages on both displays and the printer and confirms to the transaction that the load was successful.
6. The transaction reassigns the LTERMs back to their original status.

Related reference

MFS bypass for the 3270 or SLU 2 (Application Programming APIs)

Solving load problems

If a hardware error occurs while loading a programmed symbols buffer, IMS performs series of predefined actions.

About this task

If a hardware error occurs while loading a programmed symbols buffer:

- The message used to load the programmed symbols is returned to the IMS message queue.
- The terminal is taken out of service.

  Exception: SLU 2 devices are not taken out of service.
- The error is logged on the IMS log.
- Message DFS2078 is sent to the IMS master terminal. For AOI purposes, provide the message number, if possible.

After the hardware error is corrected and the terminal is in service, the message to load the programmed symbols is resent.

If the loading of the programmed symbols fails because of an error in the message to load the programmed symbols, the operator must:

Procedure

- Dequeue the message. The master terminal operator might need to issue the dequeue (**/DEQ**) command.
- Correct the error.
- Reenter the transaction to resend the programmed symbols load message.

Results

If a method is available for informing the next user of the programmed symbols buffer status, the terminals with loaded PS buffers should not be powered off. When a power failure occurs or a terminal is powered off, the contents of the PS buffers might be lost.

When a terminal is powered on, if no IMS messages are waiting to be sent to the display, an IMS response mode transaction should be entered to load all required PS buffers or some non-IMS method should be used to load the PS buffers. However, if IMS messages are waiting to be sent, and these messages require the use of one or more PS buffers, sending the queued messages must be delayed until the PS buffers can be reloaded.

For SLU 2 devices, if the PS buffers are not loaded and a message requiring the use of a programmed symbols buffer is sent to the terminal, the message is rejected and IMS:

- Returns the invalid message to the IMS queue
- Logs the error on the IMS log
- Sends message DFS2078 to the IMS master terminal and to the node, indicating an output failure.

The terminal is left in protected mode after receipt of message DFS2078.

A user application program can be designed to queue to an LTERM an unsolicited message requiring a particular programmed symbols buffer to be loaded. The first part of the message can include a load programmed symbols data stream; however, this message cannot be processed by Message Format Service (MFS).

Reloading the PS buffer

When a message is waiting on the IMS queue for a terminal and requires programmed symbols that are not loaded, you must load the programmed symbols.

To load the programmed symbols, take one of the following actions:

- If the terminal is attached by VTAM, load the PS buffers using a VTAM application.
- If a queued message requires a programmed symbols buffer, and it is normal user output (for example, the output is not response mode or conversational), then use a response-mode transaction to load the programmed symbols buffer, which loads the programmed symbols buffer so that the queued message is properly displayed.
- Dequeue (**/DEQ**) the message (or have the master terminal operator dequeue the message) that requires the use of a programmed symbols buffer; enter a transaction to load the programmed symbols buffer required; and then reenter the transaction that originally generated the queued message.

Application design considerations

A message that uses programmed symbols is different from other IMS output messages; it relies on the correct programmed symbol (PS) buffer content for its presentation.

One application might require a PS buffer content that is incompatible with a different application. Further, the PS buffer can be erased by external action (such as powering off the terminal). IMS helps in

managing this PS buffer by keeping messages that cannot be displayed on the IMS message queue; this allows the PS buffer to be loaded and the message resent.

Multiple output messages can be queued for the same device. IMS uses a set algorithm to determine which message to send to the terminal next. For example, response-mode output is always sent before any nonresponse-mode output messages are sent; this is very useful for programmed symbols, because it allows a transaction to be entered to load the PS buffer before other queued messages are sent to the terminal.

Provide a response-mode transaction to load the PS buffer. This transaction allows its output to overtake any nonresponse-mode output that requires the PS buffer to be loaded. A database keyed by LTERM name can be used to record the program symbol set name that is to be loaded into the buffer. All IMS applications using program symbols need to update this value whenever the PS buffer is loaded. In this way the terminal operator never needs to know the program symbol name.

Output messages using PS should be sent as nonresponse-mode, allowing the PS buffer to be reloaded using a special response-mode transaction (described in the previous paragraph). An alternative is to use a non-IMS program to load the PS buffer.

An Automated Operator Interface (AOI) routine can be triggered by the DFS2078 message, indicating a possible need to reload the PS buffer. This avoids the need for any human intervention. If you choose this technique, provide for invalid or corrupted 3270 data streams that produce the DFS2078 message, rather than an actual PS buffer problem. Be careful to design the AOI routine to detect and avoid looping if it automatically reactivates the terminal.

Remote 3270 errors (VTAM)

A remote 3270 can inform IMS about certain unusual conditions that it detects during its operation. When IMS receives a message of this type, called a sense-status message, from a 3271, 3274, 3275, or 3276, IMS determines if it indicates an error or some normal condition. If the message indicates an error, IMS formats the message for printing and sends it to the z/OS system console.

The sense-status messages, in general, indicate the following 3270 conditions:

- Device end (for example, operation complete on printer)
- Intervention required (for example, out of paper or 3270 display power off)
- Busy (operator action with unprotected screen)
- Hardware-detected errors (for example, data check or control check)

An intervention-required condition can be caused when:

- A display station is turned off (not on 3275)
- A printer is turned off or has its covers open
- A printer is without paper
- A security keylock is in the off position

When IMS detects an intervention-required condition, it suspends output to the unit. When you make the unit ready (by turning power on or by putting paper in the printer, for example), IMS resumes output. If a message is in progress when IMS detects the intervention-required condition, IMS resends that message from the beginning. For a 3270 printer, IMS can only detect an intervention-required condition that is caused by a power-off condition when IMS attempts to select that device for printing.

To determine which printers and displays are in an intervention-required state on a remote 3270 line, enter a **/DISPLAY LINE PTERM** or **/DISPLAY NODE** command. Those units in an intervention-required state display as COMPINOP (component inoperative).

IMS handles the printer on the 3275 (3284-3) differently when it detects the intervention-required condition. IMS marks the printer as having a component inoperable. However, the 3275 display continues to receive output messages and retains all input capabilities. Operations on the 3284-3 printer resume when you make the printer ready.

IMS retries 3271, 3274, 3275, and 3276 hardware-detected errors several times before it considers the error to be permanent. If you do not correct the error, IMS sets the physical terminal to pstopped and inoperative status and considers it unformatted.

IMS tries to pinpoint the cause of any hardware failures. It tries to eliminate the smallest component possible (for example, a terminal rather than a control unit, or a control unit rather than a line) when a problem forces elimination of resources. For a VTAM-attached 3270 terminal, IMS issues a **/CLSDST** command to disconnect the terminal. After you make appropriate repairs, you can restore IMS operation by issuing a **/START LINE PTERM** or **/RSTART LINE PTERM** command for those terminals that are part of the failing unit.

IMS tries to ensure that a terminal correctly receives all data sent to it. However, certain printer operations, including the Copy function, cause mechanical actions on the printer after the data has been transferred. For some errors occurring under these circumstances, no error recovery is possible, and none is attempted.

System console

IMS communicates with the z/OS system console using z/OS write-to-operator (WTO) and write-to-operator-with-reply (WTOR) messages.

Enter messages from the console in WTO or WTOR format. z/OS passes only the data portion of the reply to IMS. Each reply constitutes one segment of an input message. Each segment can contain either uppercase or lowercase characters. Multi-segment messages must end with a period to identify end-of-message. A period can appear either at the end of the last segment (before the end quote), or as a segment itself. Because of WTO and WTOR restrictions, IMS translates any embedded carriage return (CR) characters in output messages to blanks and ignores them.

The following data formats are acceptable. IMS WTO and WTOR reply formats vary depending on the z/OS system and the level of JES being used.

Table 42. IMS WTO and WTOR Formats

Input Location	Data Format
Command Line	`R nn,'/ERESTART CHKPT 0 '`
Second Line	`R nn,'BUILDQ FORMAT RS.'`
Transaction Line	`R nn,'FIRST SEGMENT'`
Second Line	`R nn,'LAST SEGMENT.'`
Message Switch	`R nn,'DESTNAME TEXT'`
Second Line	`R nn,'TEXT 2ND LINE.'`

IMS does not print output messages enqueued for the system console on the terminal until you enter a message. This ensures that you are present when output messages are printed. If you do not need to enter an input message, enter a blank or period as a single-character message to cause enqueued output to be printed without input-message processing.

Table 43. IMS WTOR Example

Example	Explanation
`27 DFS996I *IMS READY`	IMS is ready for an output segment.
⋮	
`R 27,'.'`	Reply to cause enqueued output to be printed.

Table 43. IMS WTOR Example (continued)

Example	Explanation
`DFS000I WTOR OPERATOR ARE YOU THERE?`	Message 1 from output queue.
`DFS000I PLEASE START TRANS XYZ...OK?`	Message 2 from output queue.
28 DFS996I *IMS READY:	IMS is ready for next input segment.

Local card reader

IMS supports a local card reader (for example, 2540 or 2502) as an input device. Before operation, however, you must enter a **/START LINE** command from the master terminal to activate the card reader.

You can then place the card deck in the hopper and make the reader ready. The reader remains active and able to read cards until you enter a **/STOP LINE** command or until an I/O error occurs.

If you press the end-of-file button, IMS enqueues the last message read for processing regardless of whether it was terminated by a slash-period (/.) delimiter. IMS then issues a read command to allow you to add another deck of cards when needed. If you do not press the end-of-file button, IMS does not enqueue the last message for processing unless it is a single-card input or IMS reads a slash-period. This allows you to fill the hopper with a partial tray of cards without determining whether the last card is also the end of the IMS message. In this case also, a read is left outstanding so you can continue with the input deck by filling the hopper and making the reader ready.

Communicating with IMS

IMS delineates a segment by the presence of an end-of-media (EM) character (X'19', card punch 11-1-8-9) in any column of the input card. IMS ignores any text following the EM character. If no EM character occurs, IMS assumes a segment to be 80 characters in length. The character string slash-period (/.) immediately **preceding** the EM character (if present) or as the first two characters in the following input card terminates a message. This allows you to input variable-length data, even though BSAM is reading fixed-length records.

When you assign a SYSIN data stream to a local reader line, IMS reads the entire file of data and closes the device. You can restart the line using a **/START** command after IMS notifies you that it has processed the file.

Responding to errors

If an I/O error occurs, follow standard I/O error-recovery procedures for the device. If IMS detects invalid data, it stops the line. Correct the condition, reenter the deck beginning with the transaction that caused the failure, and restart the line from the master terminal. You should use the **/SET** command to identify the input destination in order to prevent data errors in the middle of input streams.

Local printer

IMS supports a local printer (for example, 1443 or 1403) as a high-speed online output terminal. After you enter a **/START LINE** command from the master terminal, the printer prints output as received from the application program without your intervention.

If a permanent I/O error occurs, IMS stops the terminal. Correct the error and restart the line from the master terminal using the **/START LINE PTERM** or **/RSTART LINE PTERM** command.

Replace the magnetic tape

If you use a tape for SYSIN or SYSOUT data sets, you must change tapes when z/OS notifies you of end-of-volume (EOV).

About this task

To remove a tape before EOV for offline processing:

Procedure

1. Issue a **/STOP LINE PTERM** command from the master terminal.
2. Remove the tape, mount a new one, and to continue processing, enter one of the following commands
   - **/START LINE PTERM**
   - **/RSTART LINE PTERM**

Disk data sets

You can allocate a disk data set for use as an intermediate data file accessed by an IMS application program or any other user program. The operation is the same as for a tape file, but IMS provides no means of dismounting disk volumes.

When you allocate a group of disk data sets for the spooled SYSOUT option, the IMS Spool SYSOUT Print utility (DFSUPRT0) can copy one or more of these data sets to SYSOUT. IMS can automatically schedule this utility at end-of-volume on one of the data sets. An end-of-volume condition is also simulated:

- If an I/O error occurs while writing to a spool data set.
- During system initialization, if a spool data set was in use when the system was last running and the spool data set was not printed.

  In this case, you must issue a **/START LINE** command to complete the scheduling procedure.

  Recommendation: Define at least two data sets for your spool line.

You can reschedule the utility with the **/STOP LINE PTERM** command. You can also start the utility at any time by issuing a **/START REGION** *procname* command from the master terminal. IMS generates line numbers and PTERMs in the sequence of the input definition statements during IMS system definition. After the system is generated and ready to run, you can display line numbers to determine what line number was generated for a spool data set.

You must establish your own conventions for controlling special forms. One possible method is to allocate a logical terminal for each report. After you set up the forms for a specific report, lock the other logical terminals using the **/LOCK LTERM** command, then place the terminal in exclusive mode using the **/EXCLUSIVE** command. Thereafter, IMS prints only the data enqueued for the report you want. If you do not use a technique like this one, messages alternate among the logical terminals associated with the line.

Programmable remote systems

Operators of programmable remote systems do not communicate directly with IMS, but rather with a user-written program that resides in the remote system's storage. If you use programmable remote systems, you must provide procedures for terminal operators.

SLU-1 devices

SLU-1 devices can include communication terminals and printers.

These topics describe the following SLU-1 devices as examples:

- IBM SLU-1 Communication Terminal.

- IBM SLU-1 Data Communication System (Programmable Models). The SLU-1 supports input from a transmit data set (TDS) or a user data set (UDS), in addition to print data set (PDS) and UDS output if you define them to IMS as unattended.
- IBM SLU-1 Data Communication System (Nonprogrammable Models).

MFS is optionally available for TDS input and PDS output. You can define a SLU-1 device for either attended or unattended operation.

Printers attached to IMS as SCS SLU-1 devices

Most of the printers have Cancel, PA1, and PA2 keys that can be used when the device is operating in SNA character string (SCS) mode and is attached to an appropriate controller, such as the 3174, 3274, 3601, or 4701. When these keys are pressed while IMS is printing a message, the following results occur:

- If the Cancel key is pressed, IMS stops transmission and discards the current output message (but does not dequeue it).
- If a PA key is pressed, a string of characters is sent to IMS. You can define this string of characters as transaction code APAK 01 for PA1 and APAK 02 for PA2, or you can use the appropriate IMS user exit routine to modify this string, for example to set it to a command or message switch.

Restriction: You must specify CONSOLE instead of PRINTER1 for the COMPT1 keyword of the TERMINAL macro (or the logon descriptor for the printer). In the TYPE macro or logon descriptor, specify the printer as UNITYPE=SLUTYPE1.

SLU-1 communication terminal

These topics describe IMS operations for a SLU-1 Communication Terminal or a SLU-1 Data Communication System (nonprogrammable models).

Connecting to IMS using SLU-1 devices

Secondary Logical Unit (SLU-1) devices can connect to IMS using a telephone data set.

About this task

Prepare the SLU-1 device for communication before attempting to log on. See the SLU-1 operator's guide for the proper switch settings.

You should log on to terminals that are connected with a switched line using the procedure below. Log on to terminals connected through a nonswitched line using the same procedure, but start with step 2.

Procedure

1. Establish connection with the computer through a telephone data set.
2. Press the Sys Req key. The Proceed light should go on.
3. Enter the logon message.

   For example, to log on to an IMS system named IMSA:

   - Enter: `LOGON applid (IMSA)`
   - Response: `DFS3650I SESSION STATUS`

Results

If you want host-initiated logon for a SLU-1 Data Communication System, you must leave the terminal in a line-to-console job. To prepare the SLU-1 terminal for host-initiated logon:

1. Press the Start Job key. For a 3774 or 3775, also press the S key.
2. Press the Line key.
3. Press the Console key.

4. Press the EOM (end of message) key.

Communicating with IMS using SLU-1 devices

You can begin communication with IMS after you complete the logon procedure. If an output message is enqueued for the terminal, IMS attempts to send it immediately after the DFS3650I message. If you begin to enter an input message before you receive the output message, IMS sends the output message later.

Entering single-segment input

You must remember to press the EOM key to let IMS know that you have entered a segment.

About this task

Enter single-segment input as follows:

Procedure

1. Enter the input data.
2. `Press the carriage-return key.`
3. Press the EOM key.

Entering multi-segment input

When you set the AUTO switch to AUTO, you send data in the buffer plus an EOM to IMS when you press the carriage-return key or the Form Feed key. For a SLU-1 Communication Terminal, set the AUTO switch to OFF when you enter multi-segment input.

About this task

Enter multi-segment input as follows:

Procedure

1. Enter the first segment of the input.
2. Press the carriage-return key.
3. Press the EOB (end of buffer) key to send the buffer contents, if required. If you want to enter another segment in this buffer, do not press the EOB key.
4. Enter the next input segment.
5. Press the carriage-return key.
6. If there are more segments to enter, return to step 3. If not, press the EOM key.

Receiving an output message

You can use the Cancel and Attn keys to interrupt output messages destined for SLU-1 devices.

About this task

IMS sends all output messages for a SLU-1 device in their entirety unless you interrupt them at the device using the Cancel and Attn keys as follows:

- The Cancel key stops the current message, which IMS then discards. If another output message is presently enqueued, IMS sends it.
- The Attn key causes IMS to finish sending the current message. When the message completes, you can enter an input message.

- The Attn key followed by the Cancel key stops the current message, which IMS then discards. When the message is stopped, you can enter an input message.

SLU 1 with MFS

During IMS system definition, you can define a SLU-1 terminal to work with Message Format Service (MFS). Like a 3270 terminal, system-supplied formats are available.

About this task

Even when defined to operate with MFS, the SLU-1 terminal operates in unformatted mode (using IMS basic edit) until one of the following occurs:

- You enter //*midname*.
- MFS processes an output message to the terminal using an MFS message output descriptor (MOD) that names an MFS message input descriptor (MID) to be used to process subsequent input data.

If you enter data on the same line as //*midname*, separate //*midname* from the data with a blank. IMS discards the //*midname* and the blank, and formats the data according to the named MID. If you enter //*midname* without any data on the same line, IMS considers the next line received from the terminal to be the first line of the message.

When IMS processes an output message to a SLU-1 terminal using a MOD that names a MID, MFS uses that MID to format the next input from that terminal. An application program, the IMS **/FORMAT** command, a message switch, or some IMS functions can create this output message.

When in formatted mode (using MFS), a SLU-1 terminal remains in formatted mode until one of the following occurs:

- You enter two slashes (//) or two slashes followed by a blank. The terminal returns to unformatted mode, and IMS discards the two blanks. The two slashes are escape characters.
- You enter two slashes, a blank, and data. MFS returns the terminal to unformatted mode, IMS discards the slashes and the blank, and formats the data using basic edit.
- IMS sends an output message that does not name a MOD to the terminal.
- IMS sends an output message whose MOD does not name a MID to the terminal.

IMS uses MFS default formats for many output messages. When a terminal receives such a message, the terminal goes into formatted mode, and MFS processes the next input message unless you use the escape characters. If in doubt, use the escape characters.

Disconnecting SLU-1 devices from IMS

To disconnect from IMS, press the Sys Req key and then enter a logoff command. For a SLU-1 Data Communication System, you can also press the Stop Job key.

SLU-2 devices

SLU-2 devices can initiate a session with IMS in several ways.

Related reference

Terminals and equipment supported by IMS 15.4 (Release Planning)

Using SLU-2 devices to connect and disconnect to IMS

When you connect to or disconnect from IMS using a SLU-2 device, the procedure is the same regardless of the device.

About this task

If you define the terminal as the master terminal, IMS automatically enables the terminal for a session.

Initiate a session by using any one of the following three methods:

Procedure

- The VTAM network operator can enter the **VARY NET** command (VARY NET,ID=*a*,LOGON=*b*).
- You can enter the **/OPNDST** command from the IMS master terminal.
- VTAM can make the unit automatically available if it is defined in the VTAM definition as belonging to IMS.

What to do next

Close a terminal session by issuing a **/RCLSDST** command.

Related reference

Terminals and equipment supported by IMS 15.4 (Release Planning)

3290 information panel

When communicating with IMS using a 3290, the terminal operator might need to know that how the PA1 key operates is based on what option is selected for presenting pages to a partition.

If an error occurs, the 3290 session usually terminates. The 3290, not IMS, handles device errors.

Related concepts

3290 information panel in partitioned format mode (Application Programming APIs)

NTO logical units

IMS allows start-stop devices such as TTY and TTY-compatible devices (for example, the 3101) to attach to IMS through VTAM.

The Network Terminal Option (NTO) Program Product handles the device characteristics associated with line control and communication protocols, and translates them into SNA-SLU 1 communication protocols handled by VTAM and IMS. The operational characteristics of each device do not change, but session initiation and termination are modified as described below.

Related reference

Terminals and equipment supported by IMS 15.4 (Release Planning)

Connecting to IMS using NTO devices

Before you can initiate a Network Terminal Option (NTO) session, IMS must recognize the NTO device. You need to define the device to IMS and enter the **/START DC** from the master terminal.

About this task

For IMS to recognize the NTO device, you must do both of the following:

Procedure

1. Define the NTO device to IMS during IMS system definition using the PU keyword in the TERMINAL macro. This keyword specifies the actual hardware that this NTO device represents.
2. Enter a **/START DC** command from the IMS master terminal.

Results

IMS initiates a session with an NTO device when one of the following occurs:

- You enter the **LOGON APPLID**(*nnnnnnn*) command at the NTO terminal.
- You enter an **/OPNDST** command (for nonswitched devices only) from the IMS master terminal.

- The VTAM network operator issues a **VARY** command with the LOGON= keyword (for nonswitched devices only).

The NTO device receives a DFS3650I message when session initiation has completed successfully.

How to communicate with IMS from an NTO device

The Network Terminal Option (NTO) handles the device characteristics associated with line control and communication protocols, and translates them into SNA-SLU 1 communication protocols handled by VTAM and IMS.

You can communicate with IMS from a network terminal option (NTO) device using control characters, single-segment messages, and multi-segment messages.

Control characters

IMS uses the following control characters to delineate messages and message segments:

. (period)
End-of-message. This character signifies the end of a multisegment message when you do not use MFS.

CR
Carriage return (X'0D').

LF
Line feed (X'25').

Single-segment messages

To send data to IMS, use the following format:

- Begin the message with a transaction code.
- End the message with the CR and LF characters.

Example:

```
transaction_code  text   CR,LF
```

Multi-segment messages

To send data to IMS, use the following format:

- Begin the message with a transaction code.
- End each segment with the CR and LF characters.
- End the message with a period (.), followed by the CR and LF characters.

Example:

```
transaction_code  segment_1  CR,LF
                  segment_2  CR,LF
                   . . .
                  segment_n  CR,LF
                             .,CR,LF
```

You can reverse the order of the CR and LF characters.

Operation of the NTO device is the same as for those devices without NTO and VTAM. For additional information on entering messages, receiving messages, using response mode, using MFS, and using IMS commands from the terminal, see the preceding sections on each type of physical device.

Disconnecting NTO devices from IMS

You can disconnect the session between a network terminal option (NTO) device and IMS by issuing certain commands. NTO devices will also disconnect when IMS terminates or when VTAM terminates.

About this task

The session between an NTO device and IMS terminates when any of the following occurs:

- IMS terminates.
- VTAM terminates.
- The network operator enters a **VARY INACT** command.
- You do one of the following at the NTO terminal:
  - Issue a **/RCLSDST** command.
  - Enter a logoff command.
  - Disconnect the phone connection for a switched device.
- You enter the **/CLSDST** command from the IMS master terminal.
- Data or transmission errors force termination of the session.

Restriction: You must not stop the NTO logical unit using the IMS **/STOP** command. The network operator must issue the VTAM **VARY** command (with ANS=ON for switched devices).

Related tasks

"Connecting to IMS using NTO devices" on page 249
Before you can initiate a Network Terminal Option (NTO) session, IMS must recognize the NTO device. You need to define the device to IMS and enter the **/START DC** from the master terminal.

Special operational notes and restrictions

TTY devices and devices acting as TTY devices use blank lines in the form of carriage returns and linefeeds both before and after a message to set off the message for easier identification.

For TTY devices, IMS system messages cause a carriage return (CR) and linefeed (LF) before and after the message. Other output does not cause this to happen.

For 3101 devices acting as TTY devices, IMS sends a carriage return (CR) as X'0D'. This has the same effect on the 3101 screen for IMS system messages as the carriage return (CR) and linefeed (LF) do for the TTY.

There are no component attachments to an NTO device. You must define each component as a unique NTO device to VTAM, the NTO Program Product, the NCP, and IMS.

An NTO device cannot be used as an IMS master terminal.

Master terminal

If you use a 3270 device, IMS requires that you use a 3270 display station with a 3270 printer for the master terminal (IMS does not allow the 3284 Model 3 printer). The master terminal cannot be an ETO terminal or an LU 6.2 device.

You can define any of the following devices as the master terminal for an IMS subsystem:

- System console
- IBM 3270 Information Display System (nonswitched)
- SLU-1 device (attended mode only)
- SLU-2 device

Chapter 9. Automated operations

Automated operations are tools and techniques that help you improve your installation's productivity. As your system grows more complex and your message traffic increases, automating certain tasks can increase efficiency of your system.

IMS operations can also be automated using Tivoli® NetView® for z/OS. Because Tivoli NetView for z/OS functions independently of IMS, it can gather information and issue commands that are not available to IMS.

You can use automated operations to:

- Minimize errors
- Increase availability
- Expedite problem diagnosis and prevention

IMS provides the following tools to help you automate your operations:

- Time-controlled operations (TCO)
- Automated Operator Interface (AOI)
- REXX SPOC API

Related reference
Tivoli Software Information

Advantages of automation

Many of the jobs that operators perform are simple, repetitive tasks, such as monitoring the system and issuing recovery commands. You can often automate these jobs and thereby free the operator for more complex activities, such as implementing operational procedures.

In the IMS environment, TCO and AOI can improve your operator productivity by:

- Improving operator productivity and accuracy. Automating operator tasks simplifies procedures, reduces operator input, and minimizes operator errors. For example, TCO can reduce operator input by automatically monitoring system status, starting message regions and telecommunication lines, and notifying users of system status.
- Expediting problem determination. An operator's primary job is to bypass or fix problems quickly. If adequate information about a problem is not available, it might not be possible for the operator to fix it quickly. Automated operations are especially suited for problem determination. For example, TCO can automatically gather necessary information and do diagnostic analysis so that a problem can be quickly identified and corrected.

Deciding what to automate

Identifying operations that can be automated and implementing them requires a sound understanding of your installation's operations. You must collect and analyze various resources to identify which tasks are good candidates for automation. Repetitive and predictable tasks are good candidates for automation.

Tip: The process of automating operations should be an iterative one. After you have developed and used automated procedures, you should evaluate them and, in the process, consider whether any new tasks can be automated.

Resources you should analyze include:

- System logs
- Problem management reports
- Record of calls to the help desk

- Operators' notes

Tools for automated operations

You can use the Automated Operator Interface (AOI), the REXX SPOC API, and the time-controlled operations (TCO) to help you automate the IMS subsystem.

IMS operations can also be automated using Tivoli NetView for z/OS. Because Tivoli NetView for z/OS is independent of IMS, it can gather information and issue commands that are not available to IMS.

IMS provides the following tools to help you automate your operations:

- Automated Operator Interface (AOI), of which there are two types:
  - Type 1, available for the IMS DB/DC and DCCTL environments
  - Type 2, available for all IMS environments (DB/DC, DCCTL, and DBCTL)
- REXX SPOC API, available in an IMSplex environment
- Time-controlled operations (TCO), available for IMS DB/DC and DCCTL environments

Related concepts

"Type-2 automated operator (AO) application program (GMSG, ICMD, and RCMD calls)" on page 285
You can write type-2 automated operator (AO) application programs that will retrieve messages from an AOIE type exit routine (GMSG call), issue a subset of IMS operator commands (ICMD call), or retrieve command responses for commands issued on the ICMD call (RCMD calls).

Related reference

Tivoli Software Information

IMS Automated Operator Interface (AOI)

The IMS Automated Operator Interface (AOI) lets an application program, using DL/I calls, issue most IMS operator commands and receive command responses.

AOI lets you write an exit routine to intercept:

- IMS system messages routed to the MTO
- Commands
- Command responses

Using AOI, you can examine these messages, commands, and command responses and issue a message to any terminal or to the IMS message queue to be processed by an application program.

You can use AOI to develop commands tailored to your installation and to automate some portions of the recovery process. AOI can relieve the MTO of numerous controlling and monitoring duties. This is especially useful if you manage a large terminal network. IMS can send single segment or multi-segment messages. Your AOI exit must be able to handle both types of messages in the event that messages run over one line.

Using AOI:

- Relieves operators from repetitive command entry.
- Increases accuracy for command entry, especially when the commands have LTERM and telecommunication line control parameters.
- Helps you control IMS resources at prearranged times, such as during region startup and database image copy.
- Lets you prompt the MTO to take specific actions.
- Allows entry of commands with long operand lists. This can be important for commands that are larger than can be entered from a terminal.

Automated operator programs can:

- Supervise terminals
- Control IMS monitor and trace operations
- Help detect database problems
- Assist in recovery by issuing the commands **/STOP DB**, **/LOCK DB**, and either the **/DBRECOVERY DB** or **UPDATE DB STOP(ACCESS)**

Related concepts

"Type-1 automated operator (AO) application program (GU, GN, CMD, and GCMD calls)" on page 275
You can write type-1 automated operator (AO) application programs that will retrieve messages from the AO exit routine DFSAOUE0 (GU and GN calls), issue a subset of IMS operator commands (CMD call), or retrieve command responses for the commands that are issued on the CMD call (GCMD call).

"Type-2 automated operator (AO) application program (GMSG, ICMD, and RCMD calls)" on page 285
You can write type-2 automated operator (AO) application programs that will retrieve messages from an AOIE type exit routine (GMSG call), issue a subset of IMS operator commands (ICMD call), or retrieve command responses for commands issued on the ICMD call (RCMD calls).

Security for AO application programs (System Administration)

AOI functions

The IMS AOI consists of an exit routine and application program DL/I calls. The exit routine is called the *Automated Operator (AO) exit routine*. An application program that issues the DL/I calls is called an *AO application program*. The term AOI refers to the overall function, rather than to either of its two parts.

IMS calls the AO exit routine continually for system messages, commands, and command responses. The AO exit routine can:

- Examine IMS messages, commands, and command responses
- Manipulate IMS messages
- Send messages to an AO application program

By using an AO application program, you can:

- Retrieve messages sent from an AO exit routine
- Issue various IMS operator commands
- Retrieve responses to those commands

You can use the AO exit routine and application program together or independently of each other. For example:

- You could use just an AO exit routine to monitor IMS messages and delete unwanted messages from the master terminal (z/OS system console in the DBCTL environment).
- You could use just an AO application program to issue a set of IMS commands, perhaps to start programs or databases. When used alone, you start the AO application program just like any other IMS application program.
- You could use both the AO exit routine and AO application program together. The exit routine could, for example, direct a message to the application program. The application program could then do specific processing based on the message it receives from the exit routine. In this case, the AO exit routine starts the application program.

AOI can be used in any IMS environment (DB/DC, DBCTL, or DCCTL). But, depending on the environment, there are some differences in what AOI can do and what you must do to use it.

Related concepts

"AOI and the IMS environments" on page 258
You can use one of two Automated Operator Interface (AOI) functions, depending on which IMS environment you are operating in and what you want to do. The AOI function for the IMS DB/DC

and DCCTL environments is called *type-1 AOI*. The AOI function for the DB/DC, DCCTL, and DBCTL environments is called *type-2 AOI*.

How to use AOI to complete different tasks

You can use the automated operator interface (AOI) to discard unwanted messages, set up a batch window, restart application programs after an abend, customize messages, create an audit trail, or supervise terminals.

Discarding unwanted messages

You can write an AO exit routine to prevent the master terminal (z/OS system console in the DBCTL environment) from being flooded with unwanted IMS messages. You can still log discarded messages to the secondary master terminal (by issuing the **/SMCOPY** command from an AO application program) in case, for example, the messages are subsequently needed for problem determination.

You can also prevent the logging of IMS system messages to the secondary master terminal by using the **/SMCOPY MSG OFF** command.

Messages that are logged to the secondary master can also be controlled using a type-2 AO exit routine (DFSAOE00 or another AOIE type exit routine).

Setting up a batch window

You can write an AO application to set aside time (usually each night) for running batch jobs, that is, to set up a batch window. When you run batch jobs, you usually take the databases offline. To set up your batch window using an AO application program, you can write a multiple step job that:

- Issues a **/DBRECOVERY DB** or **UPDATE DB STOP(ACCESS)** command to take the database offline (BMP step)
- Issues a **/DISPLAY DB** or **QUERY DB** command to determine when the database is actually offline (BMP step)
- Initiates your batch work and does routine operational tasks, such as making image copies (batch step)
- Issues a **/START DB** or **UPDATE DB START(ACCESS) SET(ACCTYPE())** command to bring the databases online (BMP step)

Restarting application programs after an abend

When an application program abends, IMS issues a message and immediately stops the program. You can write an AO exit routine that sends the message to an AO application program that can then issue the **/START PROGRAM** command to allow work to continue immediately, rather than waiting for the MTO to restart the program.

Customizing messages

You can write an exit routine that customizes IMS system messages before they are routed to the master terminal. You can delete parts of messages that have no meaning for the MTO. You can reformat a message to conform to a format established at your installation. Or, you can add text to a message to tell an MTO what action to take in response to the message.

Creating an audit trail

You can use AOI to track system or user actions. For example, if you find that certain commands are not being used appropriately at your installation, you can intercept them with an AO exit routine. The AO exit routine can send a copy of the command to an AO application, which can in turn send the command, with pertinent user information, to the MTO, or copy it to the log for later analysis.

Supervising terminals

If your network is extensive, you might make one or more terminal operators responsible for terminal availability. These operators also watch for communication problems and can reassign LTERMs to alternate devices. You can write an AO application program that performs the same tasks, and in effect, functions as a supervisory terminal. Either the MTO or someone entering the appropriate transaction from a remote terminal can schedule the AO application program.

Processing of messages, commands, and command responses

The automated operator interface (AOI) process system messages, command, and command responses. AOI lets an application program, using DL/I calls, issue most IMS operator commands and receive command responses.

Because AOI can manipulate system messages, commands, and command responses, these topics define these terms and explains how AOI processes them.

Processing system messages

A *system message* is any IMS message that is not a direct response to a command. When IMS issues a system message, it sends a copy of the message to one or more of the following:

- IMS master terminal (z/OS system console in the DBCTL environment)
- z/OS system console
- IMS secondary master terminal

IMS sends all system messages destined for the IMS master terminal to the AO exit routine. IMS does not send copies of system messages destined for the z/OS system console (in a non-DBCTL environment) or the secondary master terminal to the AO exit routine. The AO exit routine can change or delete a system message sent to the master terminal. In the DBCTL environment, the original copy of a changed or deleted message is lost.

Copies of a system message destined for the z/OS system console or the secondary master terminal, if the logging of system messages is not disabled, are sent before they are sent to the master terminal.

The logging of IMS system messages to the secondary master terminal is controlled by using the **/SMCOPY MSG ON|OFF** command.

Processing commands

A *command* is any IMS operator command. IMS sends a copy of all IMS operator commands entered from a terminal to the AO exit routine, but does not send the following:

- **/FORMAT**
- **/LOOPTEST**
- **/MSVERIFY**
- **/RELEASE**

In addition, in the IMS DB/DC and DCCTL environments, IMS does not send the following commands:

- **/NRESTART**
- **/ERESTART**

In addition, in the IMS DBCTL environment, IMS does not send the two restart commands listed above to the AO exit routine.

IMS also sends copies of the following types of commands to type-2 AO exit routines (DFSAOE00 and other AOIE type exit routines):

- Commands issued by ICMD calls in an AO application program
- IMS internally-generated commands

IMS does not execute the command until after IMS sends it to the AO exit routine.

Processing command responses

After a command is executed, IMS generates a *command response*. If an AO exit routine expressed interest in the command, IMS sends a copy of the command response to the routine. Any asynchronous system messages IMS produces as a result of a command are not command responses; they are system messages.

IMS does not send copies of command responses for the following types of commands to type-2 AO exit routines:

- Commands issued by ICMD calls in an AO application program
- IMS internally-generated commands

To process a command response, the AO exit routine must check for message segments after the first one. The **/DISPLAY**, **/RDISPLAY**, and **/RMxxxx** commands all generate multi-segment response messages.

AOI and the IMS environments

You can use one of two Automated Operator Interface (AOI) functions, depending on which IMS environment you are operating in and what you want to do. The AOI function for the IMS DB/DC and DCCTL environments is called *type-1 AOI*. The AOI function for the DB/DC, DCCTL, and DBCTL environments is called *type-2 AOI*.

Type-1 AOI uses the following DL/I calls:

- CMD to issue operator commands
- GCMD to retrieve responses to those commands
- GU and GN to retrieve messages routed to the AO application program from the AO exit routine

The AO exit routine you write in this environment is named DFSAOUE0. Type-1 AOI uses the IMS message queues for communication between an AO application and an AO exit routine.

Type-2 AOI uses the following DL/I calls:

- ICMD to issue operator commands
- RCMD to retrieve responses to those commands
- GMSG to retrieve messages routed to the AO application program from the AO exit routine

The AO exit routine you write in this environment is an AOIE type exit routine. Type-2 AOI does not use the IMS message queues for communication between an AO application and an AO exit routine; instead, it uses an AOI token (an eight-byte name that associates messages with AO application programs).

Although this topic pairs the AO exit routine and application program together based on the environments in which they operate, you could use a single AO application program to issue calls and retrieve messages from DFSAOUE0 and AOIE type exit routines. You can write such an application program to run in the IMS DB/DC or DCCTL environment, but not in the DBCTL environment.

Related concepts

“Restart and recovery considerations” on page 290
Messages directed to an automated operator (AO) application cannot be recovered at restart because the automated operator interface (AOI) does not use the IMS message queues in this environment.

“AOI functions” on page 255

The IMS AOI consists of an exit routine and application program DL/I calls. The exit routine is called the *Automated Operator (AO) exit routine*. An application program that issues the DL/I calls is called an *AO application program*. The term AOI refers to the overall function, rather than to either of its two parts.

Type-1 AOI

Type-1 AOI consists of the AO exit routine, DFSAOUE0, and AO application programs that use the GU, GN, CMD and GCMD calls. Type-1 AOI is applicable for the IMS DB/DC and DCCTL environments.

Type-1 AO exit routine (DFSAOUE0)

IMS calls the type-1 exit routine, DFSAOUE0, for:

- System messages destined for the master terminal
- A subset of operator-entered commands
- Associated command responses

DFSAOUE0 can examine these messages and commands before IMS sends or executes them. IMS does not call DFSAOUE0 for system messages destined for terminals other than the master terminal or for certain commands and command responses.

DFSAOUE0 can handle both single-segment and multi-segment messages. You can write DFSAOUE0 to do the following:

- Modify IMS system messages. In addition to modifying a message, DFSAOUE0 can add up to 20 bytes of text to the end of a message. A copy of the modified message can be sent to an alternate destination.

  Messages issued by IMS after a command has been entered (for example, the DFS058"command in progress" message) are not really messages; they are command responses and cannot be modified.
- Delete an IMS system message and, optionally, send a copy of the message to an alternate destination.
- Ignore selected segments of a message or an entire message.
- Send a copy of system messages, commands, or command responses to an alternate destination.
- Send a new message to an alternate destination in response to receiving a system message, command, or command response.
- Modify a copy of a command or command response and send the copy to an alternate destination. Note that, unlike system messages, the exit routine can only modify a **copy** of commands and command responses.
- Enter a transaction to be placed on a message queue. This function lets DFSAOUE0 schedule an AO application program. For example, after receiving startup messages DFS680 and DFS994I, DFSAOUE0 could invoke an AO application program to enter a series of commands to start the network.

  Restriction: In a shared-queues environment, any transactions to be processed by the local IMS subsystem (rather than using the shared queues) must be defined as SERIAL. These transactions cannot be processed by other IMS subsystems in the sysplex.
- Request the edited command buffer (when the input is a command) to determine which command parameters succeeded or failed.

DFSAOUE0 can modify or delete system messages only. It can modify or delete the system message that the original destination (the master terminal) receives. It can also send a modified copy to an alternate destination. You cannot, however, use DFSAOUE0 to modify or delete the original command or command response. You can use it to modify a copy of a command or command response that it sends to an alternate destination, but not the copy that the primary destination receives.

Your AO exit routine must be a standalone, 31-bit, reentrant module.

Specify your exit requirements by linking the exit routines you want to use in the IMS.SDFSRESL concatenation. You can specify both type-1 and type-2 AO exit routines.

Type-1 AO application program (GU, GN, CMD and GCMD)

An AO application program can:

- Issue DL/I calls to retrieve messages from DFSAOUE0 (or AOIE type exit routines)
- Issue a subset of IMS commands
- Retrieve the responses to those commands

Retrieving messages (GU and GN call)

An AO application program receives control when IMS enqueues a message with the appropriate transaction code. The AO application program retrieves the message from DFSAOUE0 by issuing GU and GN calls.

Issuing commands and retrieving command responses (CMD and GCMD)

The AO application program issues a command using the CMD call. You can only use the CMD call if you also use the I/O PCB. PCB status codes indicate the results of a CMD call.

If there are more than one command response segments, the first one is returned to the I/O area. Use the GCMD call to retrieve subsequent segments. This call, similar to a GN call, places subsequent command response segments in the I/O area.

Security

You can secure the CMD call by using RACF (or an equivalent product), the IMS Command Authorization exit routine (DFSCCMD0), or a combination of RACF and the exit routine. With the AOI= parameter on the TRANSACT system definition macro, you can specify which programs can issue operator commands. With RACF or DFSCCMD0, you can define which commands an AOI application program can issue.

Recovery

IMS puts messages sent from DFSAOUE0 to the AO application program on the IMS message queue. Therefore, all messages are recoverable at IMS restart.

Supported application program environments

The following table shows the types of application programs that can issue CMD and GCMD calls, by IMS environment.

Table 44. CMD and GCMD call support by application region type

	IMS environment		
Application region type	**DB/DC**	**DBCTL**	**DCCTL**
DRA thread	No	No	N/A
BMP (non-message-driven)	No	No	No
BMP (message-driven)	Yes	N/A	Yes
MPP	Yes	N/A	Yes
IFP	No	N/A	No

Related concepts

Type-1 AO command security (System Administration)

Type-2 AOI

The type-2 Automated Operator Interface (AOI) consists of AOIE type exit routines (including DFSAOE00), and AO application programs that use the GMSG, ICMD, and RCMD calls. Type-2 AOI is applicable for all IMS environments (DB/DC, DCCTL, and DBCTL).

Type-2 AO exit routines (DFSAOE00 and other AOIE type exit routines)

IMS calls AOIE type exit routines for:

- IMS system messages destined for the master terminal
- Commands entered from a terminal and the responses to those commands
- Commands issued from an AO application program using the ICMD call
- IMS internal commands

IMS does not call AOIE type exit routines for:

- Command responses to internal commands
- Command responses to an AO application program ICMD call
- Commands issued from an AO application using the CMD call
- System messages for which the destination is not the master terminal

You can write AOIE type exit routines to complete the following tasks:

- Modify the text of IMS system messages. An AOIE type exit routine can also add up to 20 bytes of additional text to the end of a message.
- Delete IMS system messages. In a DBCTL environment, if a message is deleted or modified, IMS does not keep the original message sent to the z/OS system console.
- Direct any system message, command, or command response to an AO application program.
- Filter the system messages sent to the secondary master terminal.
- Start a BMP job (for example, an AO application program). AOIE type exit routines can issue SVC 34 to start a BMP. You can use the APARM parameter of the EXEC statement to pass information to the AO application program. The AO application program can issue the DL/I INQY call to retrieve the value specified for the APARM parameter. You can use the START command to override the APARM parameter of the EXEC statement.

AOIE type exit routines must use IMS AOI callable services to communicate with an AO application program. AOI callable services let you:

- Direct messages, commands, and command responses to an AO application program
- Cancel messages previously directed to an AO application program but not yet sent

Your AO exit routine must be a standalone, 31-bit, reentrant module.

Specify your exit requirements by linking the exit routines you want to use in the IMS.SDFSRESL concatenation. You can specify both type-1 and type-2 AO exit routines.

Restriction: You cannot use AOIE type exit routines to modify or delete commands and command responses, including commands from the z/OS system console or an application program, or internally generated commands.

Type-2 AO application program (GMSG, ICMD, and RCMD)

An AO application program can issue DL/I calls to:

- Retrieve messages from AOIE type exit routine (or DFSAOUE0)
- Issue a subset of IMS commands
- Retrieve the responses to those commands

Retrieving messages (GMSG call)

AOIE type exit routines do not use the IMS message queues to route messages to an AO application program. Thus, your AO application program can retrieve messages when you cannot or do not want to use the message queues.

The AO application program retrieves messages using the GMSG call, which associates an AOI token name with a message. The application program passes an 8-byte AOI token to IMS, and IMS returns to the program only those messages associated with the AOI token. By using different AOI tokens, AOIE type exit routines can direct messages to different AO applications programs. This use of the AOI token is similar to the use of transaction names using the other AO exit routine, DFSAOUE0.

The GMSG call, specifying an 8-byte AOI token, retrieves the first segment of a message associated with the token. Subsequent GMSG calls, with no AOI token specified, retrieve the second through the last segments of a multi-segment message, one segment at a time. If a program needs all segments of a multi-segment message, it must issue GMSG (with no AOI token) until all segments are retrieved before issuing another GMSG call with an AOI token. When the program issues a new GMSG call with an AOI token, IMS discards all remaining message segments from the previous call.

An application program can specify a wait in the GMSG call. If there are no messages currently available for the specified AOI token, the application is put into a wait state until a message is available. The decision to wait is specified by the program, unlike a WFI (wait-for-input) transaction where the wait is specified in the transaction definition.

Restriction: In a shared-queues environment, any transactions enqueued to an AOI token are processed by the local IMS subsystem (rather than using the shared queues) and cannot be processed by other IMS subsystems in the sysplex.

Issuing commands and retrieving command responses (ICMD and RCMD calls)

The ICMD call issues an IMS command and retrieves the first command response segment if one exists. The RCMD call retrieves subsequent command response segments. When an ICMD call results in a multi-segment command response, you might not want to retrieve all segments. If you need all segments, you must issue RCMD calls until all segments are retrieved before issuing another ICMD call. When the program issues a new ICMD call, IMS discards all remaining message segments from the previous call.

Using AIBTDLI

The GMSG, ICMD, and RCMD calls do not require the AO application program to pass a PCB to IMS. The calls use the application interface block (AIB) interface (AIBTDLI), which lets an application issue DL/I calls without a PCB address. The AIBTDLI interface also lets IMS pass unique return and reason codes to the application in the AIB.

Security

You can secure the ICMD call using RACF (or an equivalent product), the IMS Command Authorization exit routine (DFSCCMD0), or both. RACF lets you specify which commands an application program can issue. DFSCCMD0 and RACF also allow you to check authorizations during ICMD call processing.

Recovery

The messages directed from AOIE type exit routines to the type-2 AO application program are not written to the IMS message queues. Therefore, messages directed to the AO application program cannot be recovered during IMS restart.

Supported application program environments

The following table shows the types of application programs that can issue GMSG, ICMD, and RCMD calls for different IMS environments. In addition, these calls are supported for CPI Communications driven applications programs.

Table 45. GMSG, ICMD, and RCMD call support by application region type

	IMS environment		
Application region type	**DB/DC**	**DBCTL**	**DCCTL**
DRA thread	Yes	Yes	N/A

Table 45. GMSG, ICMD, and RCMD call support by application region type (continued)

Application region type	IMS environment		
	DB/DC	DBCTL	DCCTL
BMP (non-message-driven)	Yes	Yes	Yes
BMP (message-driven)	Yes	N/A	Yes
MPP	Yes	N/A	Yes
IFP	Yes	N/A	Yes

Related concepts

Type-2 AO command security (System Administration)

Related reference

IMS callable services (Exit Routines)

Summary of exit routine processing

In an IMS DB/DC or DCCTL environment, you can use the Automated Operator exit routine (DFSAOUE0) and type-2 Automated Operator exit routines (DFSAOE00 and other AOIE type exit routines).

When an AOIE type exit routine is called as part of a list of AOIE type exits, each AOIE type exit routine is called with the same parameter list. IMS does not refresh the parameter list or use information from the parameter list between calls to the individual AOIE exit routines. If your exit sets the return code parameter (AOEORPLY) to anything other than zero (AOE0IGNR), your exit routine must set the SXPLCNXT exit parameter to SXPL_CALLNXTN so that IMS does not call any more of the AOIE type exit routines in the list. Otherwise, a subsequent AOIE type exit could modify AOE0RPLY and overwrite your modification.

If AOIE type exit routines and DFSAOUE0 are loaded, IMS first calls the AOIE type exit routines sequentially, then calls DFSAOUE0 afterwards. The AOIE type exit routine can process the message or command, or it can pass control to DFSAOUE0 to do the processing.

Processing for a system message using both AO exit routines

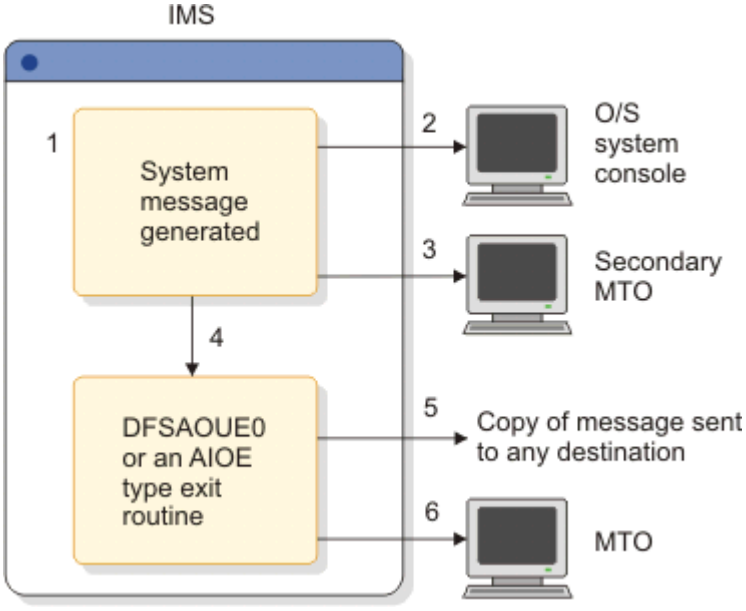


Figure 13. Processing for a system message using both AO exit routines

These steps describe the processing that occurs in the previous figure.

1. IMS generates a system message destined for the master terminal. Both AO exit routines have been specified.
2. Depending on the specific message, IMS sends a copy of the message to the z/OS system console.
3. If you specify that a copy of the message be sent to the secondary master terminal, IMS sends a copy of the message (assuming the secondary master terminal exists and that the logging of system messages to the secondary master terminal is not disabled).
4. IMS passes the copy of the message destined for the master terminal to the AOIE type exit routine. The AOIE type exit routine determines which exit routine should process the message. When both exit routines are specified, IMS only calls DFSAOUE0 if the AOIE type exit routine determines that DFSAOUE0 should process the message.
5. The exit routine that processes the message can send a copy of it to any destination. If the exit routine is an AOIE type exit routine, the destination is the AOI token, to which the message is enqueued. If the exit routine is DFSAOUE0, the destination is an LTERM or a transaction. The exit routine can alter or delete any segment of the message.
6. Unless the message has been deleted, IMS sends it to the master terminal.

Processing for a command using both AO exit routines

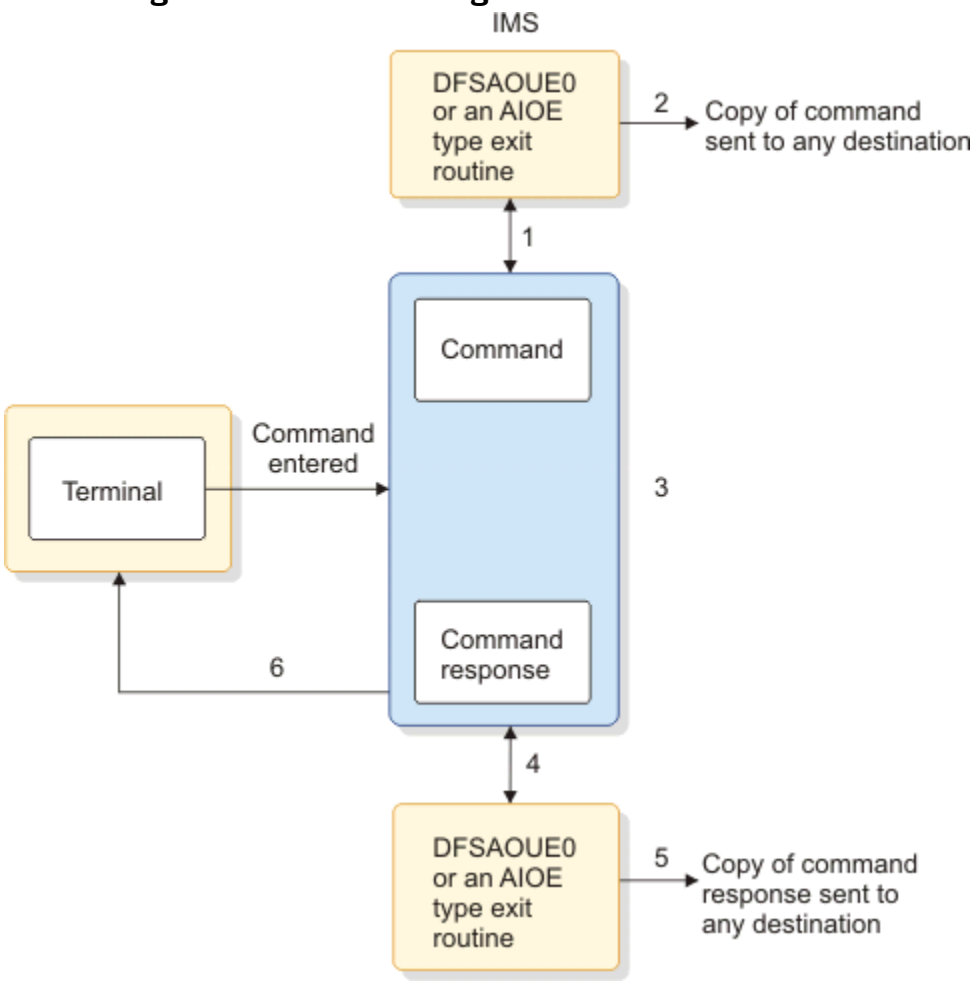


Figure 14. Processing for a command using both AO exit routines

These steps describe the processing that occurs in the previous figure.

1. When a command is entered from a terminal, IMS sends a copy of the command to the AOIE type exit routine before executing the command. The AOIE type exit routine determines whether to process the command itself or have DFSAOUE0 process the command.

2. The exit routine that processes the command can send a copy of the command to any destination. If the exit routine is an AOIE type exit routine, the destination is the AOI token, to which the message is enqueued. If the exit routine is DFSAOUE0, the destination is an LTERM or a transaction.
3. IMS executes the command and generates a command response.
4. If the exit routine requested a copy of the command response, before sending the command response, IMS passes it to whichever exit routine processed the command.
5. The exit routine that handles the command response can send a copy of it to any destination. If the exit routine is an AOIE type exit routine, the destination is the AOI token, to which the message is enqueued. If the exit routine is DFSAOUE0, the destination is an LTERM or a transaction.
6. IMS sends the command response to the terminal that originated the command.

Comparison of type-1 AOI and type-2 AOI

There are two different types of the automated operator interface (AOI): type-1 and type-2. The type-1 AOI uses the DFSAOUE0 exit routine and the type-2 AOI uses AOIE type exit routines (including DFSAOE00). You can use the type-1 AOI in the IMS DB/DC and DCCTL environments. You can use the type-2 AOI in all IMS environments, DB/DC, DBCTL, and DCCTL.

The following comparisons show the differences between the type-1 AOI and type-2 AOI:

- Type-1 AOI uses the IMS message queues; type-2 AOI does not.
- AOIE type exit routines can be used for control if messages are logged to the secondary master; DFSAOUE0 cannot.
- AOIE type exit routines can process internal commands and commands from a type-2 AO application; DFSAOUE0 cannot.
- DFSAOUE0 can send a message to an alternate destination and request the edited command buffer when the message is a command; AOIE type exit routines cannot.
- AOIE type exit routines use IMS AOI callable services; DFSAOUE0 does not.
- The `ICMD` call uses RACF (or equivalent), the Command Authorization exit routine (DFSCCMD0), or both, for security. The `CMD` call uses RACF (or equivalent), the Command Authorization exit routine (DFSCCMD0), or both for security.
- An AO application program issues `GU` and `GN` calls to retrieve a message from DFSAOUE0; it issues `GMSG` calls to retrieve messages from AOIE type exit routines.
- MPP and message-driven BMP regions can issue `CMD` and `GCMD` calls in the IMS DB/DC and DCCTL environments. MPP, BMP, and IFP regions, DRA threads, and CPI Communications driven applications can issue `ICMD` and `RCMD` calls in the IMS DB/DC, DBCTL, and DCCTL environments.
- Only specifically identified IMS applications are permitted to issue type-1 AOI `CMD` and `GCMD` calls. To identify IMS applications to issue these type-1 AOI calls, include the AOI= parameter on the system definition TRANSACT macro. All IMS applications are permitted to issue type-2 AOI `ICMD` and `RCMD` calls.

REXX SPOC API

The REXX SPOC API allows REXX programs to submit IMS operator commands to an IMSplex, to retrieve command responses, and to subscribe to the Operations Manager (OM) for unsolicited system messages, which can include responses to commands issued from any SPOC in an IMSplex that are routed back through OM.

The REXX language is a powerful command procedure language. Many operator automation programs are written in REXX and they can run in various environments. REXX programs, using the REXX SPOC API, have the following advantages over other IMS automation tools like AOI and TCO:

- Can issue type-2 commands to IMSplexes. REXX programs can also issue other commands, such as z/OS operator commands or TSO commands to operate the system.
- Can examine command responses, unlike TCO, and take further action based on those responses. Command responses that are returned to the REXX SPOC API are in the form of XML statements; these

responses are easier to parse than the command responses that are returned to the AOI interface. You can write REXX routines to improve the self-healing and self-operating abilities of your system.

For example, a REXX program issues the following command to query the system:

```
QUERY TRAN NAME(prod001) SHOW(QCNT)
```

If the command response from the REXX SPOC API shows that work is delayed (the queue count is greater than you want), the REXX program can issue additional operator commands to resolve the problem.

- Are independent of IMS (AOI runs as an IMS application program and TCO runs as an IBM subtask), and are therefore easier to implement and run.
- Can run in Tivoli NetView for z/OS.
- Can issue **/NRE** and **/ERE** commands, unlike TCO.

Related reference

Tivoli Software Information

REXX SPOC API and the CSL (System Programming APIs)

IMS time-controlled operations

The time-controlled operations (TCO) allows you to initiate time-driven procedures for any IMS operation.

TCO relieves the operator of any task that is performed routinely or at a specific time each day. TCO can issue any IMS command that an operator can issue manually, except the **/NRESTART** and **/ERESTART** commands. TCO can start and stop telecommunication lines and message regions, monitor periodic system status, and notify users of system status. TCO uses scripts that call specific exit routines that schedule transactions, execute commands, execute message switches, and deliver deferred output.

TCO is upwardly compatible with and replaces IBM's Time Initiated Input Facility (TIIF) Field Developed Program.

TCO can generate any IMS input that an IMS operator can. Specifically, TCO can:

- Execute time-initiated commands, transactions, and message switches.

  Operators usually execute transactions and message switches manually at specific times. TCO can perform these functions and can also execute commands at specified times. Using TCO eliminates errors caused by incorrectly typing commands.

  Restriction: TCO cannot execute the IMS restart commands, **/NRESTART** and **/ERESTART**, or initiate conversational transactions. TCO also does not support full-function response mode or Fast Path input transactions.
- Start automatically during IMS initialization and terminate when IMS shuts down.

In addition, you can:

- Start, stop, and change TCO scripts without disrupting operations.

  After starting IMS with the time-initiated function, you can:

  – Stop TCO processing
  – Add, delete, change, or replace a TCO script
  – Start processing an existing script
  – Begin processing a new script
  – Start TCO if it was not started automatically at IMS restart

  These actions do not affect IMS operations.
- Restart the initial script DFSTCF during IMS restart.

  When IMS is restarted, TCO loads the initial script, called DFSTCF, for processing. If you stop TCO before IMS terminates, TCO will not start processing when IMS restarts.

- Test TCO scripts without interfering with online execution or shutting down IMS.

  Using the TCO Verification utility ensures IMS processes only error-free scripts and can do so without disrupting normal operations.

The following are possible implementations of TCO facilities:

- Starting IMS resources
- Handling peak loads
- Scheduling low-priority and non-IMS jobs
- Handling operations across multiple time zones
- Monitoring the system
- Updating user status
- Scheduling different procedures for different days
- Starting large IMS TM networks in phases to prevent resource contention or deadlocks
- Shutting down IMS

TCO components

TCO (time-controlled operations) allows you to initiate time-driven procedures for any IMS operation. TCOs consist of many components and at least one exit routine.

TCO consists of:

- An initialization routine that performs initial housekeeping and starts the TCO timer service
- A TCO timer service that processes startup requests and timer requests, calls an exit routine, and waits for future requests
- A device-dependent module that passes the input generated by the timer service to the IMS control region
- An offline script-member verification utility that ensures that IMS only processes error-free scripts
- A termination routine that performs cleanup and tells the timer service when to terminate

You also need an exit routine. This can be the exit routine IMS provides or an exit routine you write.

TCO and the exit routines execute as a subtask of the IMS control region. An error in the exit routine terminates TCO only, and does not disrupt IMS online operation.

Installing time-controlled operations

There are no IMS system definition parameters for time-controlled operations (TCO). Everything you need for TCO is automatically included when you install IMS.

About this task

In order to initialize TCO when IMS starts, include a DFSTCF DD statement in the IMS procedure. You can use the data set name IMS.TCFSLIB, or you can use another data set for your TCO scripts.

IMS defines two logical terminals for TCO: DFSTCF and DFSTCFI. These terminal definitions allow TCO input to appear as though it came from a terminal. You do not need to specify either LTERM name in your IMS stage 1 input. If you do, IMS issues a G979 message telling you that your stage 1 includes duplicate LTERM names.

You can call RACF by specifying TCORACF=Y and call the Command Authorization Exit Routine (DFSCCMD0), if it exists. To sign on a user ID to IMS, you can add a /SIGN ON command at the beginning of the TCO script and a /SIGN OFF command at the end of the script. RACF checks all subsequent commands to determine if the signed-on user is authorized to issue the commands. The commands are then passed to the DFSCCMD0 exit routine.

How to use TCO to complete different tasks

IMS loads and processes time-controlled operations (TCO) scripts and exit routines in the IMS control region. So, you can modify either the script or the TCO exit routine without stopping and restarting IMS or disrupting operations.

TCO operates by processing a script that is an unblocked 80-character member in a partitioned data set. The script contains time schedule requests and, optionally, messages that are passed to the TCO exit routine.

At the time specified in a time schedule request, IMS loads a TCO exit routine and schedules it to perform a task. If the TCO exit routine generates input, IMS passes this input to the TCO device dependent module, which then passes the data to IMS using standard DL/I interfaces. If the data is a command, IMS executes it. If it is a transaction or a message switch, IMS enqueues it for processing.

IMS displays output generated from the input, including commands, on any non-ETO device you specify. Thus, you can log all input and output for automatically-generated commands or other activity.

Automatically starting IMS resources

After IMS starts, TCO can execute any non-restart commands. It can start message regions and telecommunication lines, and automatically display the system status.

Peak load handling

The TCO exit routine can assign extra message regions or terminate message regions that you no longer need based on the time of day or the status of the IMS message queues.

Start non-IMS jobs

TCO can start non-IMS-related jobs at a specific time by issuing the **/START REGION** *procname* command. For example, you can specify that:

- At 8:00 p.m., TCO notifies users that an application will terminate.
- At 8:15 p.m., TCO stops transactions for the application.
- At 8:16 p.m., TCO starts the non-IMS job.

Schedule low-priority jobs

You might have some types of transactions that are processed only once or twice a day, usually by BMPs. By collecting these transactions in the message queues at 0 priority, TCO can schedule and process the transactions at less busy times. For example, assuming your operators leave for lunch at 12:00 p.m., you can specify that:

- At 12:05 p.m., TCO issues an **/ASSIGN** command to reassign the transactions' priority (greater than 0), so processing can begin. If a the transactions need a BMP, TCO can generate one using the **/START REGION** command.
- At 12:50 p.m., TCO reassigns the transactions' priority to 0.

By the time the operators return from lunch, processing can continue for high-priority jobs.

Handle multiple time zones

If some of your IMS systems are in one time zone, for example United States Eastern-standard time, and other systems are in another time zone, for example United States Pacific time, the TCO exit routine can automatically start and stop terminals and other resources, taking the time difference into account.

System monitoring

TCO can automatically maintain periodic displays of the pools, message queues, and system status without operator intervention.

Update user status

You can automatically update the status or location of user terminals on a regular basis.

Start large networks in phases

TCO can start a large teleprocessing network in phases to prevent IMS, VTAM, or NCP resource overloads. By issuing a command for a few nodes on each physical line rather than all nodes on a few physical lines, the network starts more quickly because of increased parallel activity.

Starting large networks in phases example

While IMS is starting, TCO can issue an **/OPNDST** command to start 100 nodes. A minute later, TCO can issue another **/OPNDST** command to start another 100 nodes. Two minutes after startup, TCO can start another 100 nodes, and so on, until the entire network is started.

Shut down IMS

The TCO exit routine can notify users that IMS will shut down, and, for example, 10 minutes later, it can stop the telecommunication lines and message regions. TCO can even issue the **/CHECKPOINT** command to shut IMS down.

Stopping TCO

Use the following time-controlled operations (TCO) commands to stop and restart TCO.

To stop TCO, enter one of the following commands:

```
/PSTOP LTERM DFSTCF
/PSTOP LTERM ALL
```

To restart TCO, use one of the following commands:

```
/START LTERM DFSTCF
/START LTERM ALL
```

Time-controlled operations exit routine

The time-controlled operations (TCO) exit routine inserts messages on the IMS message queue for processing. The messages are the commands, transactions, and message switches that you specify in the time schedule requests and message sets that make up a TCO script.

The TCO exit routine passes any data found in columns 56 through 71 of the time schedule request to IMS for processing.

You do not need to write your own exit routine. You can schedule predefined commands, transactions, and message switches at predefined times using DFSTXIT0, the TCO exit routine IMS supplies.

You must bind user-written exit routines with the TCO language interface module (DFSTDLI0) and place them in the IMS.SDFSRESL data set. A user-written exit routine can have any name.

The following example shows a user-written exit routine, MYEXIT, being bound with DFSTDLI0:

```
//XIT       JOB ...
//BIND      EXEC PGM=IEWL
//SYSUT1    DD UNIT=SYSDA,SPACE=(TRK,(20,20))
//SYSPRINT  DD SYSOUT=A
//SYSLMOD   DD DSN=IMS.SDFSRESL,DISP=SHR
//INLIB     DD DSN=IMS.OBJ,DISP=SHR
```

```
//ADFSLOAD  DD DSN=IMS.ADFSLOAD,DISP=SHR
//SYSLIN    DD *
    INCLUDE  INLIB(MYEXIT)
    INCLUDE  ADFSLOAD(DFSTDLI0)
    NAME     MYEXIT(R)
/*
```

To load and execute a user-written exit routine, a time schedule request in a TCO script that is executing must refer to the routine. The following example shows a time schedule request in a TCO script that executes the user-written exit routine, MYEXIT:

```
*TIME 1200 MYEXIT
```

- Columns 1-5 indicate that this is a time schedule request.
- Columns 7-10 indicate that the request starts at 12:00 p.m.
- Columns 12-19 indicate the name of the user exit routine, in this case MYEXIT.

TCO CNT Edit exit routine

You can use the TCO CNT Edit exit routine DFSTCNT0 to restrict users (LTERMs) from loading TCO scripts.

When IMS loads a TCO script, IMS calls the CNT exit routine to validate the load request. If the routine rejects the load request, IMS sends an error message to the terminal attempting to load the script.

You do not need to write your own exit routine if you do not want to restrict any users from loading TCO scripts. IMS provides a default exit routine, DFSTCNT0, that is always called unless it is replaced with a customized exit routine. The default exit routine allows all users to load TCO scripts.

You can write your own exit routine to authorizes certain users (LTERMs) to load TCO scripts. You must name the exit routine DFSTCNT0 and replace the default routine.

IMS uses the DFSTCNT0 exit routine instead of the Message Switching (Input) Edit exit routine when the message switching destination is DFSTCF.

IMS does not use RACF to determine authorization loading for TCO scripts.

Time schedule requests

A time schedule request is a statement in a TCO script that tells IMS to process a particular task at a specified time. A time schedule request also specifies which exit routine is scheduled and what data should be passed to the exit routine.

A time schedule request can specify:

- A time of day, by hour and minute.
- A range of times, for example, every 10 minutes from 10:00 to 16:00.
- IMS startup. Requests are scheduled immediately after IMS restart, or when a new script is loaded.
- A specified delay after IMS startup, for example, 10 minutes after startup.

A time schedule request either contains 16 bytes of data or refers to a variable-length message set that contains one or more single segment or multi-segment messages.

The following figure shows two message sets and time schedule requests for broadcasting a message. The example also shows a scale to indicate the alignment of the fields.

```
.........1.........2.........3.........4.........5.........6.........7.........8

/BRO LTERM CTRL                                                    S00000001
THE IMS TIME-CONTROLLED OPERATION IS UP.                            00000002
*TIME      DFSTXIT0           S                      ****          00000003
/BRO LTERM CTRL                                                   S00000004
FIVE MINUTES HAVE ELAPSED SINCE STARTUP, AND TCO IS ACTIVE.         00000005
*TIME      DFSTXIT0      0005  S                      ****          00000006
```

In this example, the first message set is:

```
/BRO LTERM CTRL
THE IMS TIME-CONTROLLED OPERATION IS UP
```

The presence of the S in column 72 indicates the end of a segment and that the next segment is part of the same message. If there are any errors in this message set, the numbers will be added to the error message. The time schedule request indicates that this broadcast should take place as soon as the script is initialized.

The second message set in this example is:

```
/BRO LTERM CTRL
FIVE MINUTES HAVE ELAPSED SINCE STARTUP, AND TCO IS ACTIVE
```

The associated time schedule request indicates that this message should be broadcast 5 minutes after the script is initialized.

Required fields for time schedule requests

The required fields for a time schedule request specify the following: the time you want the task to be performed, and to which exit routine to pass the script.

- Columns 1-5 (ID) IDENTIFICATION

  Must contain the characters *TIME. All other requests are treated as message sets.
- Columns 12-19 (NAME) USER EXIT ROUTINE NAME

  Must include the one- to eight-character name of the exit routine in the IMS.SDFSRESL data set. If the name has fewer than eight characters, it must be left-justified and padded to the right with blanks. NAME must begin in column 12.

Optional fields for time schedule requests

By specifying one or more optional fields, you can request tasks to be processed during IMS startup or at a specified time following startup. You can also specify for processing to start, stop for an interval, and then restart.

- Columns 7-10 (HHMM) START TIME

  The time the task should start. A start time is required unless the request is for startup. Valid values for HH are 00 to 23, and valid values for MM are 00 to 59.
- Columns 21-24 (HHMM) END TIME

  The time the task should end. Use this field only if you specify an interval time in columns 26-29. Otherwise, the field is ignored. This time must be a later time than the start time specified in columns 7-10. Valid values for HH are 00 to 23, and valid values for MM are 00 to 59.

  If you request multiple dispatching for a task, the end time also stops the multiple dispatching. If you do not specify an end time, multiple dispatching continues until the end of the day (2359).
- Columns 26-29 (HHMM) INTERVAL TIME

  This field specifies the amount of time you want to defer a request. The interval time can be any time after the start time (or startup) to a time before this request is to be dispatched. Valid values for HH are 00 to 23, and valid values for MM are 00 to 59.
- Column 31 (Flag) RESIDENT INDICATOR

  Specifies whether the exit routine is resident in memory. If you specify any request for an exit routine as resident, all requests use the resident copy. The options for this column are:

  Blank
  The exit routine is resident and paged in and out as required.

  D
  The exit routine is loaded each time a request is generated.

There is an advantage and a disadvantage to specifying the exit routine as resident:

- Advantage: the request is performed more quickly because the system does not have to load the routine each time it is needed to process a time schedule request.
- Disadvantage: the resident copy is always used; IMS does not load a new copy. If you want to load a new copy of the exit routine, you must either change the resident indicator on the time schedule request statements to "D", or reload the TCO script using DFSTCF LOAD.

- Column 32 (Flag) DISPATCH INDICATOR

  Specifies when the request is dispatched (started). The options for this column are:

  Blank
  The request is dispatched each day at the start time specified in columns 7-10. The start time in columns 7-10 is required.

  O
  The request is dispatched the first day the script member is loaded. It is not dispatched again until the TCO script is reinitialized. A start time in columns 7-10 is required.

  S
  The request is dispatched as soon as the script member is initialized, either when IMS starts or when the script member is loaded. The start time, if specified in columns 7-10, is ignored.

  If you specify an interval time in columns 26-29, the request is delayed from startup before the request is scheduled. With the "S" option, the request is processed only once each time IMS is initialized or the script-member is reloaded.

- Columns 33-35

  Reserved

- Columns 56-71 MESSAGE FLAG

  This field includes 16 bytes of data to be passed to the exit routine, plus the time of day. You can choose any data, or you can leave the field blank. If you set this flag to ****, the previous message set, if any, is passed to the exit routine instead of the 16 bytes found in the time schedule request.

- Columns 73-80 LINE NUMBER

  Sequence numbers for the requests. If IMS or the TCO Verification utility detects an error, the number found in columns 73-80 is included in the error message to help you to identify the cause of the error.

  Recommendation: Include sequence numbers for all commands and message sets.

Message sets

A message set is made up of one or more messages in a TCO script-member to be passed to an exit routine for processing. A message can consist of one or more message requests.

A message request is an 80-character string that defines data to be passed to the exit routine specified in a time schedule request.

Required fields for message sets

- Columns 1-71 MESSAGE DATA

  IMS commands to be executed. Columns 1-5 must not contain "*TIME" (unless you want to specify a timed schedule request).

- Column 72 CONTINUATION

  Allows you to include single-segment or multi-segment messages and allows segments or messages to be longer than 71 characters.

  The options for this column are:

Blank
Indicates the end of the segment and the end of the message. The exit routine assumes that the next statement begins a new message. At the end of the message, you must include a time schedule request with "****" in columns 56-59.

S
Indicates the end of the segment only (that is, that the next statement begins a new segment of the same message).

X
Indicates the segment is continued in the next statement. Multiple statements are combined before the segment is passed to the exit routine. Only nine statements can be continued, for a total of 10 statements and 710 characters. You must be careful not to exceed the large message queue buffer size.

Optional field for message sets

- Column 73-80 LINE NUMBER

  Sequence numbers for the requests. If IMS or the TCO Verification utility detects an error, the number found in columns 73-80 is included in the error message to help you to identify the cause of the error.

  Recommendation: Include sequence numbers for all commands and message sets.

When IMS reaches the end of the TCO script, or at the end of the day, IMS starts again at the beginning of the script. Time schedule requests for startup are not re-executed, unless you reload the script. Requests with the dispatch indicator set to O (one-time execution) are not rescheduled unless the script is reloaded.

Statements do not need to be in time sequence in the script, nor do startup requests need to be first in the file. If TCO finds multiple startup schedule requests, or more than one time schedule request for the same time, TCO processes them in the order they are listed.

TCO Verification utility

Use the TCO Verification utility, DFSTVER0, to ensure your time-controlled operations (TCO) scripts are error free. You should run the utility before you execute any script online.

The utility detects any script errors that would occur during online execution, except an error caused by insufficient storage.

You must add a TCO script to the script library before you can execute the TCO Verification utility. You can verify more than one script at a time by assigning an input control statement for each script you want to verify.

Related reference
Time-Controlled Operations Verification utility (DFSTVER0) (System Utilities)

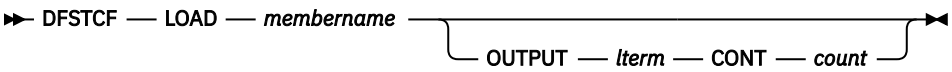
LOAD command and processing a TCO script

You can process a new time-controlled operation (TCO) script without stopping (and restarting) IMS or TCO by sending a message switch to the TCO LTERM, DFSTCF. An operator, a program, or an exit routine can issue the command to load a new script.

When you loaded a new script, and IMS processes it, IMS disregards all previous schedule requests. Being able to load TCO scripts dynamically allows you to do these tasks:

- Change or correct TCO scripts without stopping IMS
- Chain multiple scripts, so that they are processed on different days

The format of the load command is:



LOAD *membername*
Specifies the load request and the one- to eight-character script member name.

OUTPUT *lterm*
Specifies that you want IMS to send the output from the command to a specified LTERM, rather than the default LTERM, the IMS MTO. *lterm* specifies the one- to eight-character LTERM name to which IMS sends output.

CONT *count*
Specifies how many segments (continuation lines) a message can have. The default is 9. Use this keyword to change the maximum size of a message. Valid values for *count* can be 1 to 99.

You must include a blank space between each keyword and parameter.

Chapter 10. Type-1 automated operator (AO) application program (GU, GN, CMD, and GCMD calls)

You can write type-1 automated operator (AO) application programs that will retrieve messages from the AO exit routine DFSAOUE0 (GU and GN calls), issue a subset of IMS operator commands (CMD call), or retrieve command responses for the commands that are issued on the CMD call (GCMD call).

Related concepts

"Type-2 automated operator (AO) application program (GMSG, ICMD, and RCMD calls)" on page 285
You can write type-2 automated operator (AO) application programs that will retrieve messages from an AOIE type exit routine (GMSG call), issue a subset of IMS operator commands (ICMD call), or retrieve command responses for commands issued on the ICMD call (RCMD calls).

"Restart and recovery considerations" on page 290
Messages directed to an automated operator (AO) application cannot be recovered at restart because the automated operator interface (AOI) does not use the IMS message queues in this environment.

"IMS Automated Operator Interface (AOI)" on page 254
The IMS Automated Operator Interface (AOI) lets an application program, using DL/I calls, issue most IMS operator commands and receive command responses.

About the type-1 AO application program (GU, GN, CMD, and GCMD calls)

The Automated Operator Interface (AOI) consists of several IMS functions that allow installations to better monitor and control IMS activities. These functions can be used separately or in conjunction with user-written programs and the master terminal operator.

For example, one function includes a system definition option and an operator command that gives the master terminal operator more control of the command messages sent to the secondary master terminal.

An AO application can issue a subset of IMS commands normally reserved for the master terminal operator and receive responses to those commands by using the following DL/I calls:

- CMD (to issue commands and receive the first command response segment)
- GCMD (to receive subsequent command response segments)

If there are any response segments, the first is returned to the user's I/O work area. Subsequent segments are retrieved with GCMD. GCMD, similar to a get next (GN), places subsequent command response segments in the user's I/O work area.

The AO application receives control from IMS whenever a message with the correct transaction code has been queued for processing.

The following figure shows an AO application issuing a command with the CMD call (1) and receiving the initial response segment to the command (2). The AO application uses the GCMD call (3) to obtain additional response segments (4), if they exist.

Another AOI function lets an AO application retrieve messages. GU and GN calls retrieve messages. The retrieved message can be from either:

- A terminal from which the transaction was entered
- AO exit routine DFSAOUE0

An AO application using this function typically retrieves messages from the AO exit routine and then takes specific actions based on the content of the message.

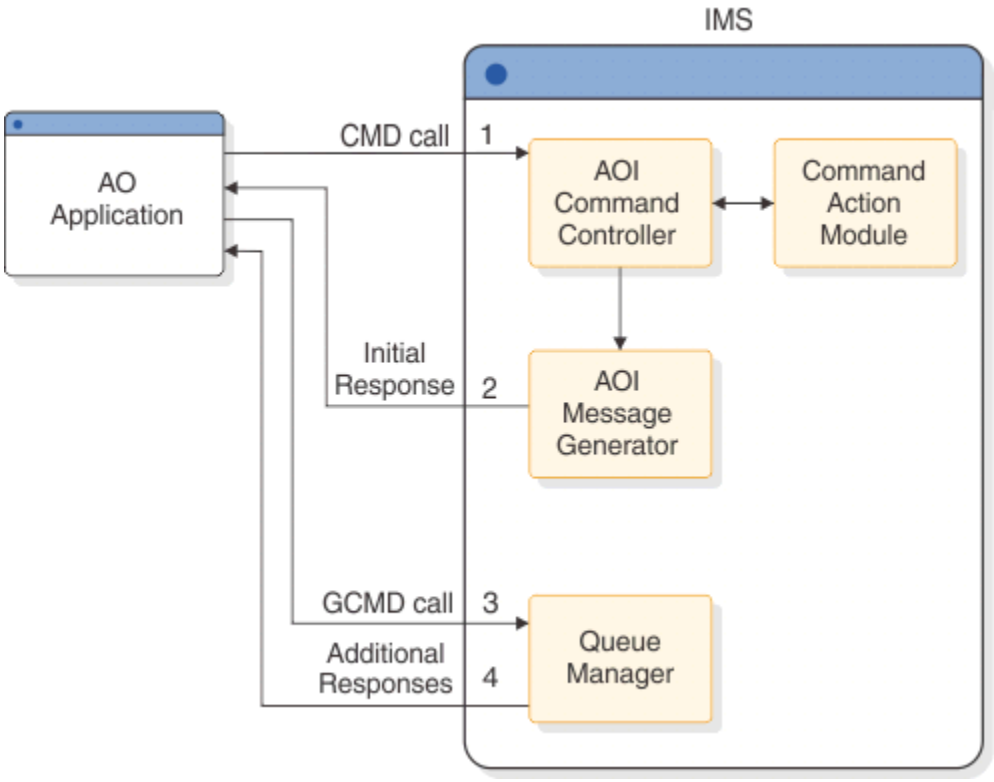


Figure 15. AO application processing

Supported application program environments

The types of automated operator (AO) applications that can issue GU, GN, CMD, and GCMD calls are listed by IMS environment.

Table 46. GU, GN, CMD, and GCMD call support by application region type

	IMS environment		
Application region type	**DBCTL**	**DB/DC**	**DCCTL**
DRA thread	No	No	N/A
BMP (non-message-driven)	No	No	No
BMP (message-driven)	N/A	Yes	Yes
MPP	N/A	Yes	Yes
IFP	N/A	No	No

AO applications for shared-queues

Some transactions must reside on the same IMS subsystem as DFSAOUE0 to process correctly. If your installation uses shared-queues, define these local transactions as SERIAL to guarantee that they are

processed on the local IMS subsystem. A transaction that is not defined as SERIAL can be processed on any IMS subsystem that has that transaction defined.

Commands used with AO applications (CMD)

To receive responses to these commands, use the GCMD call.

Format of commands

The syntax and synonyms for commands are the same whether the command is entered using a CMD call or from a terminal. All messages returned in response to a CMD call or a command entered from the terminal are the same.

The general format of your I/O work area on a CMD call is:

`LLZZ/`*verb* `KEYWORD1 P1 KEYWORD2 P2, P3.` *Comments*

LL
Two-byte field containing the length of the command text, including **LLZZ**

ZZ
Two-byte field reserved for IMS

/
Indicates that an IMS command follows

verb
The command you issued

KEYWORD*x*
Keywords that apply to the command you issued

P*x*
Parameters for the keywords you specified

. (period)
End of the command

The length of a command is limited by the size of the I/O area; the size is specified in the IOASIZE parameter in the PSBGEN macro during PCB generation. LL is the length of the command text. The size of the I/O area is the length of the actual command text, plus 4 bytes for LLZZ. The minimum size of the I/O work area is 132 bytes. The fifth byte must be a slash (/), and the verb must follow immediately. The **/BROADCAST** and **/LOOPTEST** commands must have a period between the command segment and text segment, and must be preceded by an LLZZ field that includes the size of the text. Comments can be added by placing a period (.) after the last parameter.

When issuing the **/SSR** command, do not code an end-of-message indicator (period). If a period is used, it is considered part of the text.

Responses to commands

The execution of a command with a CMD call might result in a response that comprises one or more segments. The response is similar to that received by a terminal operator upon entering the same command. The maximum size response returned by IMS to the AO application is 132 bytes (including the 4-byte LLZZ field).

DFS messages that are the initial response to the command will be time-stamped even if the NOTIMESTP keyword was coded on the COMM macro.

No response segments

This condition is indicated by a PCB status of blanks, showing that the execution of the command was successful or is in progress and would have resulted in one of the following command responses had the command been entered from a terminal:

```
DFS058 XXX COMMAND COMPLETED (No EXCEPT phrase)
DFS058 XXX COMMAND IN PROGRESS.
```

where XXX is the command issued by the automated operator.

DFS058 COMMAND COMPLETED indicates that no exceptional conditions are reported.

When DFS058 COMMAND IN PROGRESS is issued, one or more additional messages are issued to indicate whether the command was successful. These messages are sent to the master terminal rather than to AOI. Commands such as **/DBDDUMP**, **/DBRECOVERY**, **/START DATABASE**, **/STOP DATABASE**, and **/SSR** can result in message DFS058, followed by one or more asynchronous messages. Asynchronous messages are not returned as responses to the AO application.

Response segments

Response segments are identical to the response to a command entered by the operator, with preceding LLZZ and carriage control (optional) and trailing carriage controls (optional). A command might have executed successfully even though a non-blank PCB status code is returned. The status code CC indicates that one or more response segments have been produced, which occurs when a command partially executes. For example, the EXCEPT condition of this message might be returned as multiple segments.

Synchronization point processing

When IMS is restarted or resets to an earlier checkpoint, it is possible that a previously-issued command will be rescheduled for execution.

When the automated operator (AO) application regains control after a CMD call, IMS produces an initial response to the command. Command processing might be incomplete because additional IMS processing initiated by the command is still in progress.

Command responses that are not retrieved by a GCMD call are discarded at a synchronization point or by another CMD call.

A *synchronization point* is reached when any of the following events occur:

- The AO application requests the next input message with a GU call, assuming the program is defined as MODE=SINGLE on the TRANSACT macro statement.
- The program makes a CHKP call.
- The program makes a SYNC call.
- The program terminates.

Therefore, when IMS is restarted or resets to an earlier checkpoint, it is possible that a previously-issued command will be rescheduled for execution. An input message, generated by an AO exit routine and destined for an AO application, could be rescheduled although the resources are no longer held by the AO application. PCB status codes indicate more precisely which condition, or combination of conditions, exists.

Format identifiers for the /DISPLAY command

When a **/DISPLAY** command is issued from a CSLOMCMD or CSLOMI call, a command response segment is always produced. If OPTION=AOPOUTPUT is specified on the CSLOMCMD or CSLOMI call, the output from the **/DISPLAY** command also includes a format identifier (FID) on each output segment.

Your application can use the FID to determine what type of line it is processing. The mapping of the **/DISPLAY** output line with the FID can be found in the DISPLAY macro. A DFS message might be produced with no FID if a syntax error occurs.

The FID is 3 bytes of EBCDIC characters. It appears at the beginning of each line, after the LLZZ and any optional carriage control character. The FID indicates to an AOI program how to map the line of output. The FID can be useful if you are converting existing AOI programs to OM AOI programs. The format of the FID is:

FID character 1
Specific **/DISPLAY** command class (A-Z) as follows:

FID characters 2-3
Type of output line

The following are the values you can receive in FID characters 1-3 when issuing **/DISPLAY** using an ICMD.

FID char 1
Defines specific **/DISPLAY** command:

A
/DISPLAY ACTIVE, **/DIS AOITOKEN**, **/DISPLAY APPC**

B
/DISPLAY PSB

C
/DISPLAY CONVERSATION

D
/DISPLAY AREA, **/DISPLAY DATABASE**, **/DISPLAY DESCRIPTOR**

E
/DISPLAY MSNAME

F
/DISPLAY LTERM

G
/DISPLAY SHUTDOWN STATUS

H
/DISPLAY ASMT LINE/PTERM/LTERM/NODE/USER, **/DISPLAY HSB**

I
/DISPLAY ASMT LINK/MSPLINK/SYSID/MSNAME

J
/DISPLAY DBD

K
/DISPLAY LINK

L
/DISPLAY LINE, **/DISPLAY LUNAME**, **/DISPLAY PTERM**

M
/DISPLAY MASTER (/RDISPLAY)

N
/DISPLAY NODE

P
/DISPLAY PROGRAM

Q
/DISPLAY Q

R
/DISPLAY RTCODE

S
/DISPLAY STATUS

T
/DISPLAY TRANSACTION, **/DISPLAY TIMEOVER**, **/DISPLAY SYSID TRANSACTION**

U
/DISPLAY USER, **/DISPLAY TRACE**

V
/DISPLAY CCTL, **/DISPLAY OASN SUBSYS**, **/DISPLAY SUBSYS**

W
/DISPLAY OLDS

X
Time stamp, **/DISPLAY POOL DSECTS**

Y
/DISPLAY MODIFY

Z
/DISPLAY HSSP

FID chars 2-3
Defines type of output line:

00-49
Data line

50-69
Message line

70-89
Heading line

90-98
Reserved

99
Time stamp line

The response to the **/DISPLAY POOL** command does not include a FID. You can get similar information about a buffer pool using a DL/I STAT call, although the STAT call returns no Fast Path information.

Related reference
/DISPLAY commands (Commands)

Getting messages from an AO exit routine or a terminal

Your Automated Operator (AO) application receives control whenever a message with the appropriate transaction code is queued for processing. Your AO application retrieves messages from an AO exit routine or a terminal using the GU call.

Cross-reference of commands and command response messages

When an IMS command executes properly, the program does not receive message DFS058 confirming that the command completed successfully. The commands and their response messages are listed for when an IMS command does not execute properly.

The following table associates IMS commands used in an AO application and the resulting DFS response messages. Refer to this topic when writing your AO application to know what error messages to expect and be able to plan for them in your application.

Most of the messages found in the following table are error messages that the AO application receives when an IMS command executes improperly.

Exception: This is true for all commands except the **/DISPLAY**, **/RDISPLAY**, and the **/RMxxxxxx** commands. When **/DISPLAY**, **/RDISPLAY**, or **/RMxxxxxx** is successfully completed, the AO application receives the data it requested to be displayed rather than message DFS058.

Table 47. IMS commands and DFS response messages

Command	DFS response message
/ACTIVATE	107 127 158 163 181 182 2103

Table 47. IMS commands and DFS response messages (continued)

Command	DFS response message
/ALLOCATE	107 108 121 127 158 163 182 216 1950 1952 1953 2038 3110
/ASSIGN	107 119 124 127 131 138 139 147 150 151 152 157 158 161 163 164 182 201 210 211 212 213 214 232 243 289 794 795 2034 2035 2036 2104 2105 3102 3103 3104 3105 3108 3115 3632 3636 3637 3638
/BROADCAST	099 100 102 113 115 127 147 158 163 182 2105 3633
/CHANGE	107 108 121 127 130 163 181 182 216 696 1951 1953 2038 2105 2231 2232 3458 3619 3630 3632 3633 3639 3805 3811 3812 3817 3818 3819 3826 3827 3832 3833
/CHECKPOINT	107 127 130 140 141 142 163 182 2038 2717
/CLSDST/OPNDST	107 121 127 128 130 140 147 163 181 182 2037 2081 2103 2108 2109 2477 3101 3105 3106 3107 3108 3110 3111 3112 3113 3114 3116 3630 3633 3634 3639 3862
/COMPT	107 127 163 182 2103 2104 2254 2255 2256 2257 2261 3106 3108
/DBDUMP	107 127 130 132 140 141 142 160 163 174 175 182 2038 2717 3316 3319
/DBRECOVERY	107 121 127 130 132 140 141 142 158 160 163 174 175 178 182 2038 2537 2717 3316 3319 3327
/DELETE	102 106 107 113 114 115 116 117 127 128 130 146 150 158 163 2103 2104 2105 3654
/DEQUEUE	107 113 114 115 121 127 128 130 150 151 152 158 163 182 189 216 236 237 238 239 240 1191 1192 1952 1953 2038 2103 2104 2153 2183 2185 3104 3106 3107 3108 3110 3117 3824
All **DISPLAY**s	074 107 127 130 163 182 1924 2038 2093
/DISPLAY APPC	127 163 182 1953
/DISPLAY ASSIGNMENT SYSID MSNAME MSPLINK LINK	147 147 147 147
/DISPLAY AREA	116 132
/DISPLAY ASSIGNMENT NODE LTERM LINE-PTE USER	182 2103 3106 3630 3639 114 113 115 150 3108
/DISPLAY CCTL	182 216
/DISPLAY CONVERSATION	099 113 115 163 182 216 2103 3108
/DISPLAY DATABASE	116 132 3653
/DISPLAY DBD	182
/DISPLAY DESCRIPTOR	121 127 163 182 1953
/DISPLAY HSB	182 3813
/DISPLAY LINE-PTE	113 115 150 182 216
/DISPLAY LINK	147 2038
/DISPLAY LTERM	114 216 2038
/DISPLAY LUNAME	107 121 127 130 158 163 182 1953 1954 3110

Table 47. IMS commands and DFS response messages (continued)

Command	DFS response message
/DISPLAY MASTER	
/DISPLAY MODIFY	163 182 237 3431 3443 3451 3452 3455 3460 3461
/DISPLAY MSNAME	114 216 2038
/DISPLAY NODE	182 216 2103 3106 3114 3630 3639 3831
/DISPLAY OASN SUBSYS	147 182 216 3601
/DISPLAY OLDS	182 216 2038 3828
/DISPLAY PROGRAM	117
/DISPLAY PSB	147 182
/DISPLAY Q	127 147 156 163 170 216 2038
/DISPLAY RTCODE	147
/DISPLAY SHUTDOWN STATUS	134 198
/DISPLAY SUBSYS	147 182 216 3601
/DISPLAY SYSID TRAN	147
/DISPLAY TRANSACTION	146 216
/DISPLAY USER	3108 3653
/END	113 115 150 152 158 181 182 189 285 2103 2104 3106 3108 3635 3654
/EXCLUSIVE	113 115 150 152 158 181 182 189 241 2103 2104 3106 3108 3635 3654
/EXIT	107 180 181 182 183 184 189 576 577 2103 2104 2105 3106 3108 3635 3655
/FORMAT	070 182 244 3824
/IDLE	107 127 130 134 147 163 181 216 2105 2109 3633
/LOCK	107 114 116 117 127 146 158 163 182 2038 3250
/LOG	
/LOOPTEST	113 115 143 150 152 181 195 196 203
/MONITOR	099 107 113 115 130 140 150 163 182 216
/MSASSIGN	107 127 147 163 164 182 2038 2151 2152 2153 2154 2155 2156 2157 2233 2240 2244 2252 2259 2260 3200 3214
/PSTOP	099 107 113 114 115 117 130 140 146 147 150 156 163 181 182 216 232 2152 2270 2272 2273 2297 3204 3630 3633 3824
/PURGE	099 107 108 113 114 115 117 127 130 140 146 150 156 163 182 216 1953 2038 3630 3633 3824
/QUIESCE	107 130 140 158 163 181 182 216 2103 3106 3630 3632 3633 3824
/RDISPLAY	
/RM	181 182 2038 3322
/RSTART	099 107 113 115 130 140 147 150 163 181 182 216 2103 2152 3104 3106 3108 3204 3630 3633 3639 3824
/SECURE	107 127 163 182 1953
/SMCOPY	165 181 182
/SSR	182 696 2038 3601 3609 3610 3618

Table 47. IMS commands and DFS response messages (continued)

Command	DFS response message
/START	099 107 108 113 114 115 117 127 130 132 140 146 150 156 158 163 170 175 176 178 179 182 216 232 696 1953 2010 2020 2021 2022 2023 2024 2025 2038 2103 2105 2475 3104 3106 3108 3110 3315 3316 3319 3601 3604 3609 3630 3633 3639 3798 3799 3805 3806 3807 3808 3809 3816 3817 3824 3829 3843
/STOP	099 107 108 113 114 115 117 127 130 132 140 146 150 156 158 163 170 175 176 178 179 182 216 232 696 1953 2010 2020 2021 2022 2023 2024 2025 2038 2103 2105 2109 2475 3104 3106 3108 3110 3315 3316 3319 3601 3604 3630 3633 3639 3805 3806 3808 3824 3825 3829
/SWITCH	107 127 140 163 696 2038 3810 3813 3814 3824
/TEST	113 115 143 150 152 158 181 182 189 196 282 283 284 2103 2104 2158 3106 3108 3635 3654
/TRACE	107 108 113 130 147 150 158 163 182 216 237 279 280 281 314 775 1953 2028 2029 2030 2031 2032 2038 2093 2103 2104 2155 2460 3106 3108 3110 3630 3633 3639 3813
/UNLOCK	107 114 116 117 127 146 158 163 182 2038 3250 3813
/VUNLOAD	107 130 140 163 182 216

Sample AO application (UETRANS)

A sample AO application is available from the IMS library (IMS.ADFSSRC, member name UETRANS). The sample shows use of AOI by an AO application and an associated AO exit routine. The AO application is written in PL/I for the Optimizing Compiler.

The sample AO application:

- Accepts only messages originating from the exit routine (indicated by a CG PCB status code).
- Upon receipt of a DFS994 message, looks for the character string EMERGENCY. If found, it issues the command **/RSTART LINE ALL**. Otherwise, it issues the following commands:

```
/RSTART LINE ALL
/START DATABASE ALL
/START TRAN ALL
```

- Upon receipt of a terminal error message, it issues the command **/RDISPLAY** to determine on which line and PTERM is the master terminal. If the error occurred on that line, it issues a **/DISPLAY LINE ALL** command to determine the first available started line, and reassigns the master terminal to that line and PTERM.
- Upon receipt of an **/ASSIGN** command, if the response is a DFS119 message, reissues the command until it is successful. If the response is a DFS138 message, it restarts the line with an **/RSTART** command and reissues the **/ASSIGN** command. It reissues the **/ASSIGN** command a maximum of 20 times; if the command is not successfully completed after the 20th attempt, an appropriate message is issued.

The associated AO exit routine is assumed to:

- Pass all DFS994 (checkpoint identifier) messages to the AO application.
- Pass all terminal error messages to the AO application. The line number of the terminal in error is placed in the last 20 bytes of the buffer. Message numbers passed are DFS001, DFS025, DFS026, DFS027, DFS029, DFS072, DFS250, DFS251, DFS253, DFS260, DFS269, DFS973, DFS998.
- Pass all **/ASSIGN** commands that receive a DFS119 or DFS138 message to the AO application. These messages indicate that either the LTERM is busy or the line is not started and cannot be reassigned.
- Ignore all other messages.

Chapter 11. Type-2 automated operator (AO) application program (GMSG, ICMD, and RCMD calls)

You can write type-2 automated operator (AO) application programs that will retrieve messages from an AOIE type exit routine (GMSG call), issue a subset of IMS operator commands (ICMD call), or retrieve command responses for commands issued on the ICMD call (RCMD calls).

Related concepts

"Type-1 automated operator (AO) application program (GU, GN, CMD, and GCMD calls)" on page 275
You can write type-1 automated operator (AO) application programs that will retrieve messages from the AO exit routine DFSAOUE0 (GU and GN calls), issue a subset of IMS operator commands (CMD call), or retrieve command responses for the commands that are issued on the CMD call (GCMD call).

"Tools for automated operations" on page 254
You can use the Automated Operator Interface (AOI), the REXX SPOC API, and the time-controlled operations (TCO) to help you automate the IMS subsystem.

"IMS Automated Operator Interface (AOI)" on page 254
The IMS Automated Operator Interface (AOI) lets an application program, using DL/I calls, issue most IMS operator commands and receive command responses.

About the type-2 AO application program (GMSG, ICMD, and RCMD calls)

The automated operator (AO) application can help you monitor and control IMS activities. You can use it separately or with a type-2 automated operator exit (DFSAOE00 or another AOIE type user exit).

This AO application can:

- Issue DL/I calls to retrieve messages from an AOIE type exit routine (GMSG call).
- Issue a subset of IMS operator commands (ICMD call).
- Retrieve responses to those commands from the AOIE type exit routine (RCMD call).

Retrieving messages (GMSG call)

When an AOIE type exit routine is used to route messages to an AO application, the IMS message queues are not used. Instead, applications that cannot access the message queues or that do not want to use the message queues retrieve the messages from the AOIE type exit routine.

The AO application retrieves messages using the GMSG call. GMSG associates an AOI token name with a message. The application passes an 8-byte AOI token to IMS, and IMS returns to the application a message associated with the AOI token. By using different AOI tokens, the AOIE type exit routine can direct messages to different AO applications.

The GMSG call, specifying an 8-byte token, retrieves the first segment of the message associated with the token. Subsequent GMSG calls, with no token specified, retrieve the second through n segments of a multi-segment message. If an application needs all segments of a multi-segment message, it must issue GMSG until all segments are retrieved before issuing another GMSG call with a token specified. IMS discards all remaining message segments from the previous GMSG call when a new GMSG call specifying an AOI token is issued from the same AO application.

An application can specify a wait on the GMSG call. If there are no messages for the specified AOI token, the application is put into a wait state until a message is available. The decision to wait is specified by the application, unlike a WFI (wait for input) transaction where the wait is specified in the transaction definition.

Defining the AOI token

You define the AOI token value. You also decide how the AO application gets the AOI token value. Some possible methods for doing these follow:

- The token could be a hard-coded value understood by both the AOIE type exit routine and the AO application.
- A naming convention could be established that relates the name of an AOI token to an AO application. For example, an AO application with the name ABCAOI02 might use AOI token AOI02nnn.
- The AO application could issue an ICMD '/LOG data ' call with the AOI token in the data. The application would then issue a GMSG call with WAITAOI specified for the AOI token named in the LOG call.

  The AOIE type exit routine will see the ICMD and can parse the data in the LOG command to get the AOI token name and any other information in the data. The AOIE type exit routine inserts messages to the AOI token name; the messages are then returned to your AO application when the GMSG call is issued.
- The AOIE type exit routine could create a dynamic AOI token when it processes an IMS message. Then the AOIE type exit routine could issue an SVC 34 to start a BMP. The START command can override the APARM parameter on the EXEC statement. The APARM parameter is a way of passing information to the AO application. To retrieve the value specified on the APARM parameter, the AO application can issue the DL/I INQY call.
- The AOIE type exit routine could store an AOI token in an area that both the AOIE type exit routine and the AO application have access to. The AO application, when it starts, could get the AOI token from this area.

Related reference

DL/I calls for IMS TM system services (Application Programming APIs)

Issuing commands and retrieving command responses (ICMD and RCMD calls)

ICMD issues an IMS command and retrieves the first command response segment if one exists. RCMD retrieves subsequent command response segments. When the response to an ICMD call results in a multi-segment message, you might want to retrieve all message segments.

If you need all message segments, you must issue RCMDs until all segments are retrieved before issuing another ICMD call. IMS discards all remaining message segments from the previous ICMD call when a new ICMD is issued from the same AO application.

Restriction: All messages queued to an AO application by an AOIE type exit routine remain in the IMS subsystem to which they are queued. Only AO applications in the local subsystem can issue a DL/I GMSG call to access the queued messages. Because these messages are not written to a share-queue structure, applications on other IMS subsystems cannot access them.

Supported application program environments

The types of automated operator (AO) applications that can issue GMSG, ICMD, and RCMD are listed. In addition, these calls are supported from a CPI-C driven program.

Table 48. GMSG, ICMD, and RCMD call support by application region type

	IMS environment		
Application region type	**DBCTL**	**DB/DC**	**DCCTL**
DRA thread	Yes	Yes	N/A
BMP (non-message-driven)	Yes	Yes	Yes
BMP (message-driven)	N/A	Yes	Yes
MPP	N/A	Yes	Yes

Table 48. GMSG, ICMD, and RCMD call support by application region type (continued)

Application region type	IMS environment		
	DBCTL	**DB/DC**	**DCCTL**
IFP	N/A	Yes	Yes

AO application security

Security consists of controlling which applications can issue the Issue Command (ICMD) call and which commands they can specify using ICMD.

ICMD security is implemented using RACF (or equivalent), the Command Authorization Exit Routine (DFSCCMD0), or both. To specify one or both methods of securing ICMD, use the AOIS parameter in the DBC, DCC, and IMS procedures.

You need to use the following RACF commands to implement security for AO applications:

ADDGROUP
ADDUSER
PERMIT
RDEFINE

Related concepts

Security for AO application programs (System Administration)

Related reference

Parameter descriptions for IMS procedures (System Definition)

Commands used with AO applications (ICMD)

The syntax and synonyms for commands are the same whether the command is entered using an ICMD call or from a terminal. All messages returned in response to an ICMD call or a command entered from the terminal are the same. To receive responses to these commands, use the RCMD call.

Format of commands

The general format of your I/O work area on an ICMD call is:

`LLZZ/`*verb* `KEYWORD1 P1 KEYWORD2 P2, P3.` *Comments*

LL
Two-byte field containing the length of the command text, including **LLZZ**.

ZZ
Two-byte field reserved for IMS.

/ or CRC
Indicates an IMS command follows. CRC (command recognition character) rather than a slash (/) is used in the DBCTL environment.

verb
The command you issued.

KEYWORD*x*
Keywords that apply to the command you issued.

P*x*
Parameters for the keywords you specified.

. (period)
End of the command.

The length of a command is limited by the size of the I/O area; the size is specified in the IOASIZE parameter in the PSBGEN macro during PCB generation. LL is the length of the command text. The size of the I/O area is the length of the actual command text, plus 4 bytes for LLZZ. The minimum size of the I/O work area is 132 bytes.

The fifth byte must be a slash (/) (or CRC for DBCTL), and the verb must follow immediately. The /BROADCAST and /LOOPTEST commands must have a period between the command segment and text segment, and must be preceded by an LLZZ field that includes the size of the text. Comments can be added by placing a period (.) after the last parameter.

When issuing the /SSR command, do not code an end-of-message indicator (period). If a period is used, it is considered part of the text.

Responses to commands

The execution of a command with an ICMD call might result in a response that comprises one or more segments. The response is similar to that received by a terminal operator upon entering the same command. The maximum size of the response returned by IMS to the AO application is 132 bytes (including the 4-byte LLZZ field). DFS messages that are the initial response to the command are time-stamped even if the NOTIMESTP keyword is coded on the COMM macro.

No response segments

This condition is indicated when AIB return and reason codes are zero and the length of the message returned is zero. These indicators tell you execution of the command was successful or is in progress and would have resulted in one of the following command responses had the command been entered from a terminal:

```
DFS058 XXX COMMAND COMPLETED (No EXCEPT phrase)
DFS058 XXX COMMAND IN PROGRESS.
```

where XXX is the command issued by the automated operator.

When DFS058 COMMAND COMPLETED is issued, no exceptional conditions are reported.

When DFS058 COMMAND IN PROGRESS is issued, one or more additional messages are issued to indicate whether the command was successful. These messages are sent to the master terminal rather than to AOI. Commands such as **/DBDDUMP**, **/DBRECOVERY**, **/START DATABASE**, **/STOP DATABASE**, and **/SSR** can result in message DFS058, followed by one or more asynchronous messages. Asynchronous messages are not returned as responses to the AO application.

Response segments

Response segments are identical to the response to a command entered by the operator, with preceding LLZZ and carriage control (optional) and trailing carriage controls (optional). Command responses not retrieved by an RCMD call are discarded on the next ICMD call.

Format identifiers for the /DISPLAY command

Output from the **/DISPLAY** command includes a format identifier (FID) on each output segment. **/DISPLAY** commands generally return multiple lines with different formats to the AO application. Your AO application uses the FID to determine what type of line it is processing. A DFS message might be produced with no FID when there is a syntax error.

When a **/DISPLAY** command is entered with an ICMD call, a command response segment is always produced. Normally, the response is a multi-segment message. Each segment contains EBCDIC and line control characters. Each segment varies in length with a maximum of 132 but usually less than 80 characters.

The FID is 3 bytes of EBCDIC characters. It appears at the beginning of each line, after the LLZZ and any optional carriage control character. The format of the FID is:

FID character 1
Specific **/DISPLAY** command class (A-Z)

FID characters 2-3
Type of output line

The following are the values you can receive in FID characters 1-3 when issuing **/DISPLAY** using an ICMD.

FID char 1
Defines specific **/DISPLAY** command:

A
/DISPLAY ACTIVE, /DIS AOITOKEN, /DISPLAY APPC

B
/DISPLAY PSB

C
/DISPLAY CONVERSATION

D
/DISPLAY AREA, /DISPLAY DATABASE, /DISPLAY DESCRIPTOR

E
/DISPLAY MSNAME

F
/DISPLAY LTERM

G
/DISPLAY SHUTDOWN STATUS

H
/DISPLAY ASMT LINE/PTERM/LTERM/NODE/USER, /DISPLAY HSB

I
/DISPLAY ASMT LINK/MSPLINK/SYSID/MSNAME

J
/DISPLAY DBD

K
/DISPLAY LINK

L
/DISPLAY LINE, /DISPLAY LUNAME, /DISPLAY PTERM

M
/DISPLAY MASTER (/RDISPLAY)

N
/DISPLAY NODE

P
/DISPLAY PROGRAM

Q
/DISPLAY Q

R
/DISPLAY RTCODE

S
/DISPLAY STATUS

T
/DISPLAY TRANSACTION, /DISPLAY TIMEOVER, /DISPLAY SYSID TRANSACTION

U
/DISPLAY USER, /DISPLAY TRACE

V
/DISPLAY CCTL, /DISPLAY OASN SUBSYS, /DISPLAY SUBSYS

W
/DISPLAY OLDS

X
Time stamp, **/DISPLAY POOL DSECTS**

Y
/DISPLAY MODIFY

Z
/DISPLAY HSSP

FID chars 2-3
Defines type of output line:

00-49
Data line

50-69
Message line

70-89
Heading line

90-98
Reserved

99
Time stamp line

The response to the **/DISPLAY POOL** command does not include a FID. You can get similar information about a buffer pool using a DL/I STAT call, although the STAT call returns no Fast Path information.

Restart and recovery considerations

Messages directed to an automated operator (AO) application cannot be recovered at restart because the automated operator interface (AOI) does not use the IMS message queues in this environment.

When a recoverable command is issued by the ICMD call, log record X'02' is generated when the command completes. If a failure occurs, the command is not reprocessed at restart time. If a failure occurs after a command is completed but before the command response is returned, the command response is lost. Your AO application could issue the appropriate **/DISPLAY** command, if applicable, to determine if the ICMD call executed successfully (as is required with asynchronous commands).

Related concepts

"Type-1 automated operator (AO) application program (GU, GN, CMD, and GCMD calls)" on page 275
You can write type-1 automated operator (AO) application programs that will retrieve messages from the AO exit routine DFSAOUE0 (GU and GN calls), issue a subset of IMS operator commands (CMD call), or retrieve command responses for the commands that are issued on the CMD call (GCMD call).

"AOI and the IMS environments" on page 258
You can use one of two Automated Operator Interface (AOI) functions, depending on which IMS environment you are operating in and what you want to do. The AOI function for the IMS DB/DC and DCCTL environments is called *type-1 AOI*. The AOI function for the DB/DC, DCCTL, and DBCTL environments is called *type-2 AOI*.

Sample AO application (DFSAOPGM)

A sample AO application is available from the IMS library `IMS.ADFSSMPL`, member name DFSAOPGM.

Chapter 12. What to do if the IMPORT DEFN SOURCE(CATALOG) command or DDL automatic import times out in a managed ACBs environment

If the IMPORT DEFN SOURCE(CATALOG) command or DDL automatic import fails before commit phase 2, the IMS import command master aborts the online change.

If the IMPORT DEFN SOURCE(CATALOG) command or DDL automatic import fails after a commit phase 2 begins and the directory updates are committed, the IMS import command master cannot abort the online change.

You should run an IMPORT DEFN SOURCE(CATALOG) command after the problem is resolved to finish the import and get all of the IMSs out of an online change state.

An import from the catalog that is initiated by the IMPORT DEFN SOURCE(CATALOG) command or DDL automatic import can take some time when there are many ACBs in the staging directory. IBM recommends that the IMPORT DEFN SOURCE(CATALOG) timeout value be made longer to avoid timing out without the results.

If the import times out, you can determine whether the import succeeded or failed by using one or more of the following methods:

- Issue the QUERY MEMBER TYPE(IMS) SHOW(STATUS) command to see whether any IMSs remain in an online change state and if so, whether the online change is committed. The online change is committed if any IMS shows a status of import commit phase 2 or greater: IMPPHC*n*, where IMP means IMPORT, PH means PHASE, C means COMMIT, and *n* represents the phase of 2 or greater.
- Issue the QRY IMSPLEX SHOW(STATUS) command to see whether any IMSs, SCIs, or RMs are down in the IMSplex. Timeouts can be caused by an IMS address space involved in the import failing.
- Code the OM output exit to receive the IMPORT DEFN SOURCE(CATALOG) unsolicited output that was sent after the command timed out by viewing the x'70xx' log records. For example, if the x'7004' COMMIT MEMBER OLC and x'7005' COMMIT MEMBER COMPLETE OLC log records are included, the IMPORT was successful.
- If the IMSplex includes Resource Manager (RM), look for the following RM global process messages:
  - CSL2200I CLIENT client INITIATED PROCESS DFSOLC FOR IMSPLEX *imsplexname*
  - CSL2201I CLIENT client TERMINATED PROCESS DFSOLC FOR IMSPLEX *imsplexname*
- Look for the following import messages on the system console log that indicate when the import was started, if it completed, or if it failed:
  - DFS4896I IMPORT DEFN SOURCE(CATALOG) PROCESSED ONE OR MORE RESOURCES, COMMAND TOKEN=*command_token*.
  - DFS4897I IMPORT DEFN SOURCE(CATALOG) COMPLETED, COMMAND TOKEN=*command_token*.
  - DFS4898E IMPORT DEFN SOURCE(CATALOG) FAILED, COMMAND TOKEN=*command_token*.
  - DFS7433E IMPORT PHASE phase FAILED BECAUSE RM PROCESS STEP SENT TO IMS *imsid* TIMED OUT, COMMAND TOKEN=*command_token*. This message identifies what phase of online change failed. If the phase is CMT2, CMT3, or CMT4, the directory updates are committed and cannot be aborted.
  - DFS7445A IMPORT TIMED OUT AFTER DIRECTORY UPDATES COMMITTED, CHECK IF IMPORT NEEDS TO BE RETRIED. This message indicates that the import timed out after the directory updates were committed.

If the import failed after commit phase 2 began and the directory updates were committed, one or more IMSs are stuck in an online change state. You can resolve this by doing the following tasks:

- Address the reason for the failure; for example, start SCI, RM, or other components that failed.

- Issue the IMPORT DEFN SOURCE(CATALOG) command to finish the import and remove the IMSs out of an online change state. The IMPORT command should be routed to the IMS that was the import command master of the previous IMPORT DEFN SOURCE(CATALOG) command or DDL automatic import that timed out because it knows the state of the import. This is the IMS that issued the DFS7445A message.

Related reference

IMPORT DEFN SOURCE(CATALOG) command (Commands)

Related information

CSL2201I (Messages and Codes)
CSL2200I (Messages and Codes)
DFS4896I (Messages and Codes)
DFS4897I (Messages and Codes)
DFS4898E (Messages and Codes)
DFS7433E (Messages and Codes)
DFS7445A (Messages and Codes)

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_.

Trademarks

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

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

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

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

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

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

Terms and conditions for product documentation

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

Applicability

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

Personal use

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

Commercial use

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

Rights

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

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

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

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

IBM Online Privacy Statement

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

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

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

To learn more, see IBM Privacy Statement.

Bibliography

This bibliography lists all of the publications in the IMS 15.4 library.

Title	Acronym
IMS Version 15.4 Application Programming	APG
IMS Version 15.4 Application Programming APIs	APR
IMS Version 15.4 Commands, Volume 1: IMS Commands A-M	CR1
IMS Version 15.4 Commands, Volume 2: IMS Commands N-V	CR2
IMS Version 15.4 Commands, Volume 3: IMS Component and z/OS Commands	CR3
IMS Version 15.4 Communications and Connections	CCG
IMS Version 15.4 Database Administration	DAG
IMS Version 15.4 Database Utilities	DUR
IMS Version 15.4 Diagnosis	DGR
IMS Version 15.4 Exit Routines	ERR
IMS Version 15.4 Installation	INS
IMS Version 15.4 Licensed Program Specifications	LPS
IMS Version 15.4 Messages and Codes, Volume 1: DFSMessages	MC1
IMS Version 15.4 Messages and Codes, Volume 2: Non-DFS Messages	MC2
IMS Version 15.4 Messages and Codes, Volume 3: IMSAbend Codes	MC3
IMS Version 15.4 Messages and Codes, Volume 4: IMSComponent Codes	MC4
IMS Version 15.4 Operations and Automation	OAG
IMS Version 15.4 Release Planning	RPG
IMS Version 15.4 System Administration	SAG
IMS Version 15.4 System Definition	SDG
IMS Version 15.4 System Programming APIs	SPR
IMS Version 15.4 System Utilities	SUR

Index

Special Characters

Numerics

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

Z

Product Number: 5635-A06
5655-DS5
5655-TM4