IBM IMS Connect Extensions for z/OS
Version 2 Release 4

# *User's Guide*

SC19-4364-01

IBM IMS Connect Extensions for z/OS
Version 2 Release 4

# *User's Guide*

SC19-4364-01

**Note:**
Before using this information and the product it supports, read the "Notices" topic at the end of this information.

**Second Edition (September 2016)**

This edition applies to Version 2 Release 4 of IBM IMS Connect Extensions for z/OS (product number 5655-S56) and to all subsequent releases and modifications until otherwise indicated in new editions.

# Contents

##

#

##

##

##

##

##

##

# About this information

IBM® IMS™ Connect Extensions for z/OS® (also referred to as IMS Connect Extensions) is a tool that enhances the operation of IMS Connect.

These topics are designed to help IMS Connect administrators perform the following tasks:

- Plan for the installation of IMS Connect Extensions
- Install and operate IMS Connect Extensions
- Customize your IMS Connect Extensions environment
- Diagnose and recover from IMS Connect Extensions problems

Instructions that are specific to the IMS Connect Extensions Operations Console are contained in the online help for that component.

Specific changes since the previous edition of this book are indicated by a vertical bar (|) to the left of a change. Editorial changes that have no technical significance are not noted.

To use these topics, you should have a working knowledge of the following products:

- The z/OS operating system
- IMS Connect for z/OS
- ISPF
- SMP/E (only required if you are installing IMS Connect Extensions)

Always check the IMS Tools Product Documentation page for the most current version of this information:

http://www.ibm.com/software/data/db2imstools/imstools-library.html

# Part 1. Getting started

These topics provide an overview of the features and benefits of IMS Connect Extensions, describe how to install and configure it, and introduce the ISPF dialog interface and Operations Console graphical user interface (GUI).

**Topics:**

# Chapter 1. Overview

IBM IMS Connect Extensions for z/OS (referred to as IMS Connect Extensions) is a tool that enhances the operation of IMS Connect. IMS Connect, a function of IMS, is the premier pathway for accessing IMS applications and databases via TCP/IP.

IMS Connect Extensions extends IMS Connect by providing the following features:

**Monitoring and recording of IMS Connect activity**

IMS Connect Extensions provides a detailed audit of activity, giving you the information you need to analyze performance, throughput, resource availability, and security. You can also use this information to debug clients and new applications.

**Single point of control for multiple IMS Connect systems**

Centralized management of all your IMS Connect systems, including monitoring and control of OTMA and Open Database workloads, MSC physical links, and remote IMS Connect systems, from an ISPF Operations dialog or Operations Console GUI client.

**Enhanced transaction management**

Dynamic management of TCP/IP transactions, allowing you to define rules to automatically distribute workloads and reroute messages when IMS system failures occur. Ability to assign selected OTMA routing rules to a routing plan and dynamically switch between routing plans. Automatic response to changes in the IMS environment such as dynamically added datastores and flood conditions.

**Enhanced Open Database management**

Dynamic management of TCP/IP DRDA® requests, allowing you to define rules to redistribute workload based on capacity or by relative machine running costs. Ability to assign selected ODBM routing rules to a routing plan and dynamically switch between routing plans.

**Socket management**

Controls the number of input messages for a persistent session, allowing automatic distribution of persistent session workloads in a sysplex environment.

**Enhanced security**

Control access to OTMA transactions or DRDA requests based on the client IP address and IMS Connect port number, and to IMS Connect instances, via a system authorization facility (SAF) security class.

**Improved client services**

Additional features for IMS Connect clients such as enhanced information in error messages, password change facility, and extended message translation.

These features enable you to:

- Improve the availability, reliability, and performance of IMS Connect
- Speed and simplify problem determination
- Make your systems more transparent so that they are easier to audit and manage

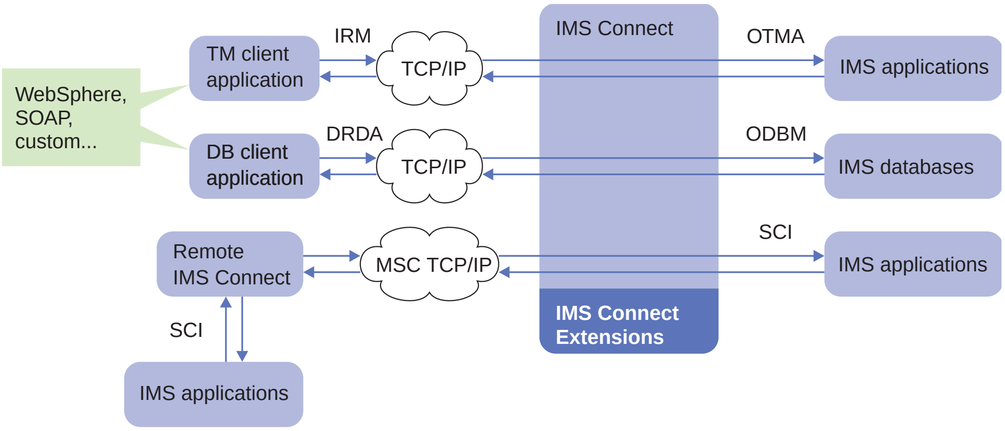


*Figure 1. IMS Connect with IMS Connect Extensions*

IMS Connect Extensions consists of components that run with IMS Connect, journal data sets that record IMS Connect activity, and an ISPF dialog-based client and Operations Console GUI client for managing IMS Connect systems and their IMS Connect Extensions features.

# What's new in IMS Connect Extensions

This topic summarizes the technical changes for this edition.

New and changed information is indicated by a vertical bar (|) to the left of a change. Editorial changes that have no technical significance are not noted.

## Changes for Version 2 Release 4

This topic lists the technical changes that were introduced in or incorporated into IMS Connect Extensions V2.4.

### SC19-4364-01

This edition describes new functions introduced by PTFs for the following APARs: PI65651, PI63839, PI59534, PI43276, PI26433.

- Purge ACEE cache entries based on ENF 71 notification of changes to a user's security profile.
- Automatic detection of mixed-case passwords setting when an IMS Connect system starts up.
- New control options and keywords supported the CEXCTLIN data set:
  - MESSAGE allows specified messages to be written to the WTO destination or not to be written to CEXPRINT.
  - SECURITY PWCASE determines how mixed-case passwords are handled.
  - SECURITY VALIDATE_TRUSTED determines whether IP validation occurs when a message has the “trusted user” flag set.
- New WARNRC0 option on Archive Manager utility sets the return code to zero after a non-critical warning.

### SC19-4364-00

- Host command environment for REXX supersedes the batch command utility:

  In V2.3 you could use the CEXDFCMD utility to connect to an IMS Connect system and submit a sequence of IMS Connect Extensions commands in batch.

In V2.4 the new CEX host command environment enables IMS Connect Extensions commands to be embedded in REXX programs, which allows more flexible automation of IMS Connect operations.

- Programs can take advantage of REXX features such as variables, conditional logic, and integration with other host environments.
- Programs can connect to multiple IMS Connects.
- Programs can be submitted interactively as well as in batch.

The CEXDFCMD batch command utility is still supported, so existing batch jobs will continue to work.

The CEX host command environment has equivalents for the CEXDFCMD batch commands and also supports the following new host commands and options:

- ADD command adds an entry to the in-memory datastore table for a datastore that is defined to IMS Connect but was not defined to IMS Connect Extensions at startup.
- DRAIN command safely terminates IMS Connect sessions by marking them for closure at the next inbound message.
- OPTION command specifies run-time options MSGLVL, TIMEZONE, and CONID that affect subsequent host commands.
- QUERY command supports additional query types ACEE_CACHE, PLAN, and SESSIONS.
- SET PLAN command activates or deactivates the OTMA routing plan or ODBM routing plan.
- UPDATE command allows certain settings, such as datastore and alias capacity weights, event collection level, and session message limit options, to be changed.
- WTO command issues a Write To Operator message to the console.

- Ability to logically group both OTMA routing rules or ODBM routing rules as belonging to a routing plan, and to switch dynamically from one routing plan to another.
- The pre-routing user exit for rules-based routing provides a customizable service that can be used to prevent messages being routed to specific datastores based on the availability or health of a specific transaction on the IMS system.
- The session DRAIN command allows safe closure of specific sessions without interrupting in-flight messages.
- Alternate transaction code support provides a way to route transactions using something other than the primary transaction code.
- ACEE cache statistics can be displayed from the Operations - System View and System Overview ISPF dialogs.
- Ability to assign datastores and Open Database targets a relative processing capacity of zero, making them ineligible as routing candidates.

## IMS Connect Extensions Operations Console changes

- Enhancements to Status Monitor on the IMS Connects tab:
  - New **Clear ACEE Cache** and **ACEE Cache Statistics** context options.
  - New **Update** context option allows updating of routing plans, event collection levels, and session message limit options.
  - New fields available including: Start Time; OTMA Routing Plan; ODBM Routing Plan; Event Collection Level; Message Limit; Message Limit Threshold.
- Enhancements to Status Monitor on the Datastores tab:

  - New **Update Capacity Weight** context option.
  - New fields are available, including: Member; TMember; XCF Group.
- Enhancements to Sessions:
  - New **Drain Eligible Sessions** context option allows safe closure of specific sessions without interrupting in-flight messages.
  - New fields are available, including: Socket Type; In IMS Conversation; Commit Mode; Synch Level; Resume TPIPE; Alternate Transaction Used; Alternate Transaction Length; Alternate Transaction Offset; Alternate Transaction Code.
- Usability improvements in the List Filters option enhance filtering of tabular data:
  - New Contains operator allows filtering where any field matches the specified value.
  - Manage/Define List Filters toolbar button provides shortcuts to activate or deactivate a filter and to manage conditions.
- New "Manage list layout" option allows a particular arrangement of tabular data in each view to be saved and reused.
- The Find function is enhanced to allow the search to be in any field or limited to a selected field.
- New Properties context menu option displays a detailed view of the properties of the selected item.

## Changes for Version 2 Release 3

This topic lists the technical changes that were introduced in or incorporated into IMS Connect Extensions V2.3.

### SC19-3632-01

This edition describes new functions introduced by PTFs for the following APARs: PM76242, PM77360, PM69452, PM64619, PM50767, PM96836, PI05160, and PI08666.

- Ability to qualify rules-based routing by transaction code. That is, you can create optional rules that route specified transaction codes to different target datastores.
- New Drain/Resume feature allows the OTMA routing status of a datastore to be suspended temporarily when an IMS system is stopped.
- New Datastore Monitor shows the status of a datastore with respect to IMS, IMS Connect, and IMS Connect Extensions routing.
- Batch command utility QUERY PENDING_RESPONSES feature returns the number of messages being processed for the datastore in IMS. This allows an automated script to drain and then safely shut down an IMS system.
- The Session Message Limit option allows you to specify the maximum number of input messages for a persistent session. This allows the balance of sessions across IMS Connect systems to be maintained in environments where a session distribution mechanism is used.
- IMS Connect Extensions support for zIIP offload.
- Support for IMS V13. To use IMS Connect Extensions V2R3 with IMS V13, APARs PM76242 and PM88073 are required.
- Support for dynamically added IMS Connect ports (introduced in IMS V13).
- Support for dynamically added IMS Connect datastores (introduced in IMS V13).
- OTMA Global Flood Warning support helps protect IMS systems from experiencing an abend due to a message flood condition.

- New keywords are supported by the CEXROUTE control option:
  - The INELIGIBLEIF keyword is used to control how IMS Connect Extensions responds to a Global Flood Warning condition.
  - The RBR_FAILURE keyword specifies how to route an input message that has no valid destination.
  - The RBR_NODEST and RBR_NOALIAS keywords specify routing behavior when no matching routing rule is defined.
  - The SECURITY CACHED_USER_MAX keyword specifies the maximum number of cached users.
- New ICON_CONTROL PORT control option specifies a dedicated IMS Connect port which is to be used for routing IMS Type-1 commands.
- Enhancements to the batch command utility:
  - New ROUTE command enables automation of datastore drain and restore.
  - New SHELL command runs a command on a specified target system or datastore.
  - REFRESH command now supports security class rules (SAF)
- Enhancements to the archive journal cleanup utility (CLEAN):
  - The NAME keyword now accepts a comma-separated list of HWS systems.
  - New DELERROR parameter allows deletion of entries that have an invalid time stamp.
- New IVP programs CEXCSE06 to CEXCSE09 use the HWSSMPL0 exit.

## SC19-3632-00

- Support for monitoring of IMS Connect to IMS Connect TCP/IP MSC communications:
  - New object types MSC and RICON (remote ICON) are supported in the Status Monitor. New session type MSC is supported in the Active Sessions display.
  - Ability to Stop and Resume communications on a selected MSC physical link or with a remote IMS Connect instance.
- A new EVENTLOGGING control option optimizes performance by buffering events before they are written to the active journal.
- The IRM_Timer value is now displayed on the Active Sessions dialog for OTMA or MSC sessions.
- Active Sessions FORM prefill option automatically selects fields corresponding to the session type (OTMA | ODBM | MSC).
- New ODBM security exit CEXAUTH0 provides the same access control options as for OTMA workloads.
- A new installation verification program is provided that produces a DRDA workload.
- The command shell now supports IMS V12 IMS Connect WTOR commands and IMS Connect z/OS commands.
- The Print utility PRINT command supports new keywords STATIME and STOTIME. The READ62 command supports new keywords IPV4ADR and IPV6ADR.
- Ability to validate IMS Connect access based on the client IP address and IMS Connect port.
- Ability to dynamically override the Transaction Expiration and Client ID Cancellation options specified in the transaction definition.

- The batch definition EXTRACT utility has been enhanced to produce a standard configuration file that can be distributed to Operations Console users.

Operations Console client:

- The Operations Console is now a plug-in to the IBM Tools Base Connection Server framework, enabling it to be installed along with other supported IBM Tools plug-ins.
- Import/export facility allows a standard Operations Console configuration file to be distributed to other users
- Active sessions summary allows grouping of sessions by selected fields
- Support for saving multiple filters in Active Sessions and the Status Monitor
- Selective highlighting of items that match certain filter conditions in both Active Sessions and the Status Monitor
- Autorefresh of the Active Sessions display

## Changes for Version 2 Release 2

This topic lists the technical changes that were introduced in or incorporated into IMS Connect Extensions V2.2.

### SC19-2817-02

This edition describes new functions introduced by the PTF for APAR PM15459.

- Open Database Manager support enables routing and monitoring of DRDA requests. The Status Monitor and Active Sessions dialogs have been enhanced to display information about ODBMs and aliases.
- Control options data set: New ODBMDEFAULTS and SET keywords are supported in the CEXCTLIN DD statement.

### SC19-2817-01

This edition describes new functions introduced by PTFs for the APARs PM04021, PM04643, and PM12498.

- Support for IMS Type-1 commands has been added to the ISPF command shell and the Operations Console.
- Authorized clients can now submit user data for logging to the journal.
- Control options data set: A new optional CEXCTLIN DD statement enables selected options to be provided when IMS Connect Extensions restarts. The supported keywords are CEXTRACE and CEXROUTE.
- **Warning override** profile setting enables you to bypass host name and archive JCL warnings during definition maintenance. SET OPTION=(WARNING=IGNORE) is the equivalent for the definition maintenance and definition takeup utilities.
- The Status Monitor now supports the `RESet FORM` primary command.
- New chapter "Tracing message flows" describes IMS Connect Extensions tracing.

### SC19-2817-00

- The Operations Console provides a graphical user interface for accessing the operational management features of IMS Connect Extensions from distributed platforms. You can view activity on IMS Connect systems, submit commands, and list active sessions across the sysplex.

- Rules-based routing introduces the ability to route messages based on the datastore ID that a message specifies in the DESTID field. You can include a fallback list in case none of the target datastores are available.
- All IMS Connect Extensions routing features now use the IMS V10 OTMA message flood detection features to enhance selection of candidate datastores.
- Active Sessions enhancement: display extended TCP/IP statistics.
- Status Monitor enhancements: display the timeout (Keep Alive) value for IMS Connect ports; display OTMA super member for individual datastores.
- The command shell now supports new IMS Connect commands in IMS V11
- Event collection for new IMS V10 and IMS V11 workloads: includes information about ODBM access, Port Exit activity, XML messages, and message-related activity.
- Centrally administer transaction timeout, transaction expiration, and duplicate client cancellation policy by modifying definitions for individual transactions.
- Persistent IMS Connect Extensions tracing across system restarts.
- “Verify user” service exit function.
- New batch **REFRESH** command to refresh definitions.
- Mixed-case password support for IMS Connect Extensions security validation.
- A new ISPF-based Definitions Setup dialog lets you create IMS Connect Extensions definitions from an IMS Connect configuration file.

**Terminology changes:** In previous versions the term “console” was used to refer to an agent running in the IMS Connect address space that listens on a TCP/IP port for connections from IMS Connect Extensions ISPF dialog clients. From V2.2 this is called a console listener to distinguish it from the IMS Connect Extensions Operations Console, the Eclipse-based GUI client.

## Features and benefits

IMS Connect Extensions provides a set of enhancements to the IMS Connect function of IMS. Collectively, these enhancements reduce the cost of providing TCP/IP access to IMS applications while improving the overall quality of service and increasing the security and auditability of IMS.

**Diagnose and fix TCP/IP problems**

IMS Connect Extensions helps you to administer and operate IMS Connect by providing the information you need to diagnose problems in real-time, and then, to fix problems, allowing you to change more IMS Connect options dynamically.

You can use the ISPF Operations dialog or GUI Operations Console to:

- View instrumentation for all active IMS Connect instances.
- View information on active sessions, highlighting delays and inactivity and TCP/IP statistics.
- View details of port status and utilization for systems or sysplexes, including dynamically added ports.
- Monitor datastores across all of your systems for OTMA flood conditions.

Combined with IBM IMS Problem Investigator for z/OS, IMS Connect Extensions enables rapid problem determination, including the ability to view important state event information such as datastore and OTMA transaction member (tmember) availability, OTMA failures and timeouts, exit failures, and session errors.

**Optimize TCP/IP performance**

IMS Connect Extensions helps you better manage performance, accurately plan capacity, and dynamically manage workloads.

IMS Connect Extensions collects internal event data from IMS Connect that you can use with IMS Performance Analyzer for z/OS. Using these products together, you can produce detailed performance reports with useful information including:

- Flow control data. For example: ACK, NAK, and OTMA.
- Internal and external transit times and latencies for IMS Connect transactions.
- Message activities. For example: OTMA, READ, and XMIT exits, read and write socket utilization, and security and commit confirm elapsed times.
- Network activity. For example: port and socket utilization, amount of data processed and accepted, and read and write socket request counts.
- IMS information. For example: resume tpipe requests, including information on each command type.

IMS Connect Extensions helps you manage IMS Connect workloads by providing flexible control of input message traffic. It can balance workloads between eligible datastores and reroute all messages to an alternate datastore when a given datastore is unavailable or in a flood condition. IMS Connect Extensions can also reject input messages or generate warnings when a user-defined message rate is exceeded. Sets of OTMA routing rules or ODBM routing rules can be assigned to different routing plans, and IMS Connect Extensions can be switched dynamically from one routing plan to another.

The Session Message Limit option allows you to automatically respond to the failure of an IMS Connect system by balancing the number of TCP/IP sessions directed to each remaining live IMS Connect system.

**Manage all your IMS Connect systems from a single user interface**

IMS Connect Extensions provides a management interface that:

- Centralizes all operations. Operators can work a Group view that organizes systems by sysplex, location, business unit, or any other grouping that you choose.
- Provides an easy-to-use sysplex view of your IMS Connect instances, including remote IMS Connect systems over an MSC link.
- Integrates monitoring with command and control capabilities.

IMS Connect Extensions includes both an ISPF Operations dialog and a GUI. The Operations Console is an Eclipse-based PC application that provides a graphical interface to perform IMS Connect Extensions operations. It allows you to monitor and control IMS Connect systems and their components and active sessions from a distributed workstation.

You can perform tasks such as monitoring IMS Connect throughput, monitoring datastore health and activity, analyzing and controlling client sessions, issuing IMS Connect WTOR commands and IMS Connect z/OS commands, IMS Type-1 commands, and IMS Connect Extensions commands, starting and stopping traces, or viewing logs.

Operations functions are secured with flexible access control rules, allowing you to tailor which commands users can issue based on their role in your organization.

**Maximize the availability and reliability of IMS Connect**

IMS Connect Extensions helps you meet service-level objectives by maximizing IMS and IMS Connect availability with these features:

- Add, delete, or reload some IMS Connect message exits without interrupting IMS Connect.
- Dynamically add datastore capacity and adjust workload balancing profiles.
- Control access to IMS Connect instances via a system authorization facility (SAF) security class.
- Control access to an OTMA transaction or DRDA request based on the client IP address and IMS Connect port number.
- Refresh IMS Connect SAF security profiles dynamically.
- Temporarily suspend routing of messages to a datastore, allowing it to be 'drained' of in-progress transactions. This can reduce the likelihood of transactions failing or being rejected when IMS is shut down.
- Use interactive and batch commands to seamlessly change between routing plans and adjust distribution of workload according to the capacity of each datastore or ODBM target. These can be used to respond to differing periods of demand or planned maintenance outages.
- Use batch QUERY commands to automate responses to changing datastore conditions such as processing status and number of active connections.

**Record and report IMS Connect activity**

IMS Connect Extensions records information about the transactions IMS Connect processes. Records are collected throughout the transaction. These records provide information on performance, response time, and throughput. For example:

- Performance and response time information for IMS, IMS Connect, and user message exits.
- Availability information for datastores and ports.
- Throughput information for different programming models; for example, conversational, non-conversational, and Send Only.
- Resource availability.

IMS Performance Analyzer provides a comprehensive set of reports from the IMS Connect performance and accounting data collected by IMS Connect Extensions.

## IMS Connect Extensions components and architecture

IMS Connect Extensions consists of several components. Some components run in the IMS Connect address space and are initiated by IMS Connect when it starts.

### Components

This topic describes the main components of IMS Connect Extensions.

**ISPF dialog client**

The ISPF dialog client connects via TCP/IP to one or more IMS Connect Extensions console listeners, providing centralized monitoring and control of IMS Connect systems across your enterprise. You can also configure the IMS Connect Extensions server component through the ISPF dialog. The configuration settings are stored in a VSAM repository.

**Operations Console GUI client**

An Eclipse-based PC application that provides a graphical interface to perform IMS Connect Extensions operations. It allows you to monitor and control IMS Connect systems and their components and active sessions from a distributed workstation.

The GUI client is also referred to as the Operations Console in this information.

**Console listener**

An agent running in the IMS Connect address space that listens on a TCP/IP port for connections from IMS Connect Extensions clients. It allows clients to access information about an IMS Connect system and issue commands to that system. A console listener provides information about a single IMS Connect system; a single client can connect to multiple console listeners.

**Repository**

A VSAM key-sequenced data set (KSDS) that contains configuration data for IMS Connect Extensions. You can manage the repository using the ISPF dialog client or the batch definition utilities.

**Exit manager**

When IMS Connect starts, it initializes the IMS Connect Extensions user exit manager. The user exit manager loads user exits on behalf of IMS Connect and dynamically configures event collection, real-time monitoring, message translation, and workload management for the exits. The exit manager handles both IBM-supplied and custom user message exits.

**Event collector**

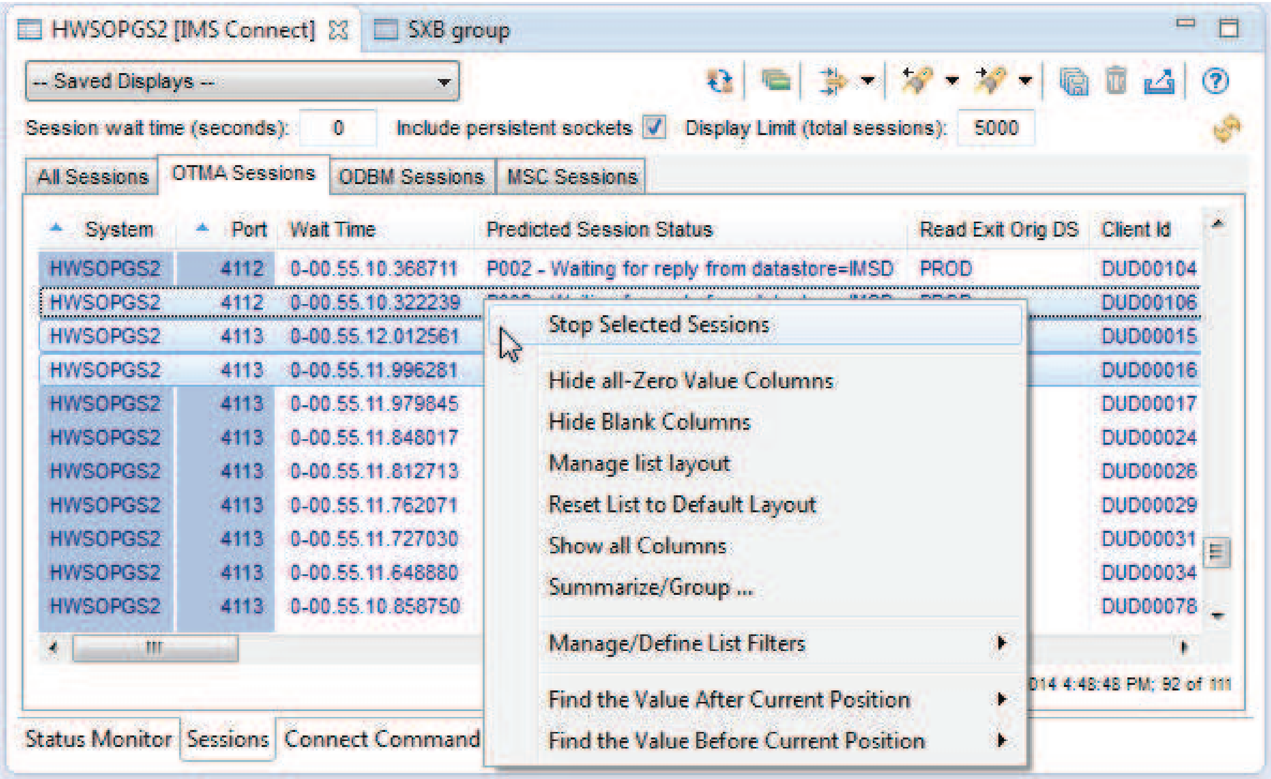
The event collector collects IMS Connect events and data about the input and output to every function call to every exit, and including data sent to and returned by custom user exits. Event records are collected in journals (see “Active journals, archive journals, and the Archive Manager” later in this section). The event collector is initialized by the exit manager.

**Status monitor agent**

Provides real-time statistics on message processing activity for a system, its exits, datastores, ODBMs, aliases, and ports, including datastores and ports that are added dynamically in IMS. Status monitor statistics are available through the ISPF Operations dialog or Operations Console.

**Active session agent**

Provides real-time tracking of active TCP/IP and LOCAL port sessions. Active session information is available through the ISPF Operations dialog, Operations Console, or publisher API (see “Publisher interface” in this section). Through the ISPF Operations dialog and Operations Console you can view details of active sessions and stop hung sessions.



Session wait time (seconds): 0 Include persistent sockets ☑ Display Limit (total sessions): 5000

| System | Port | Wait Time | Predicted Session Status | Read Exit Orig DS | Client Id |
|---|---|---|---|---|---|
| HWSOPGS2 | 4112 | 0-00.55.10.368711 | P002 - Waiting for reply from datastore=IMSD | PROD | DUD00104 |
| HWSOPGS2 | 4112 | 0-00.55.10.322239 | | | DUD00106 |
| HWSOPGS2 | 4113 | 0-00.55.12.012561 | | | DUD00015 |
| HWSOPGS2 | 4113 | 0-00.55.11.996281 | | | DUD00016 |
| HWSOPGS2 | 4113 | 0-00.55.11.979845 | | | DUD00017 |
| HWSOPGS2 | 4113 | 0-00.55.11.848017 | | | DUD00024 |
| HWSOPGS2 | 4113 | 0-00.55.11.812713 | | | DUD00026 |
| HWSOPGS2 | 4113 | 0-00.55.11.762071 | | | DUD00029 |
| HWSOPGS2 | 4113 | 0-00.55.11.727030 | | | DUD00031 |
| HWSOPGS2 | 4113 | 0-00.55.11.648880 | | | DUD00034 |
| HWSOPGS2 | 4113 | 0-00.55.10.858750 | | | DUD00078 |

014 4:48:48 PM; 92 of 111

*Figure 2. Sessions tab in Operations Console*

**Publisher interface**

Provides to monitoring applications real-time data about event records and active sessions running in an IMS Connect region, as well as configuration information about the IMS Connect system. The publisher interface supports the IMS Connect monitoring function of OMEGAMON® for IMS on z/OS. It is implemented by the CEXPAPI macro.

**Active journals, archive journals, and the Archive Manager**

Active journals contain IMS Connect event data recorded by IMS Connect Extensions. They are written to directly by the event collector. Archive journals are sequential data sets or GDGs, on DASD or tape, containing data copied from active journals. They are created by the Archive Manager.

IMS Connect Extensions journals can be analyzed by other IMS tools, such as IMS Problem Investigator and IMS Performance Analyzer to provide debugging, performance analysis, and auditing information.

**Batch utilities and host command environment for REXX**

Batch utilities provide additional features such as definition maintenance, reporting and log formatting, and journal archiving.

The CEX host command environment for REXX enables IMS Connect Extensions commands to be embedded in REXX programs, which allows more flexible automation of IMS Connect operations.

The following figure shows these components:

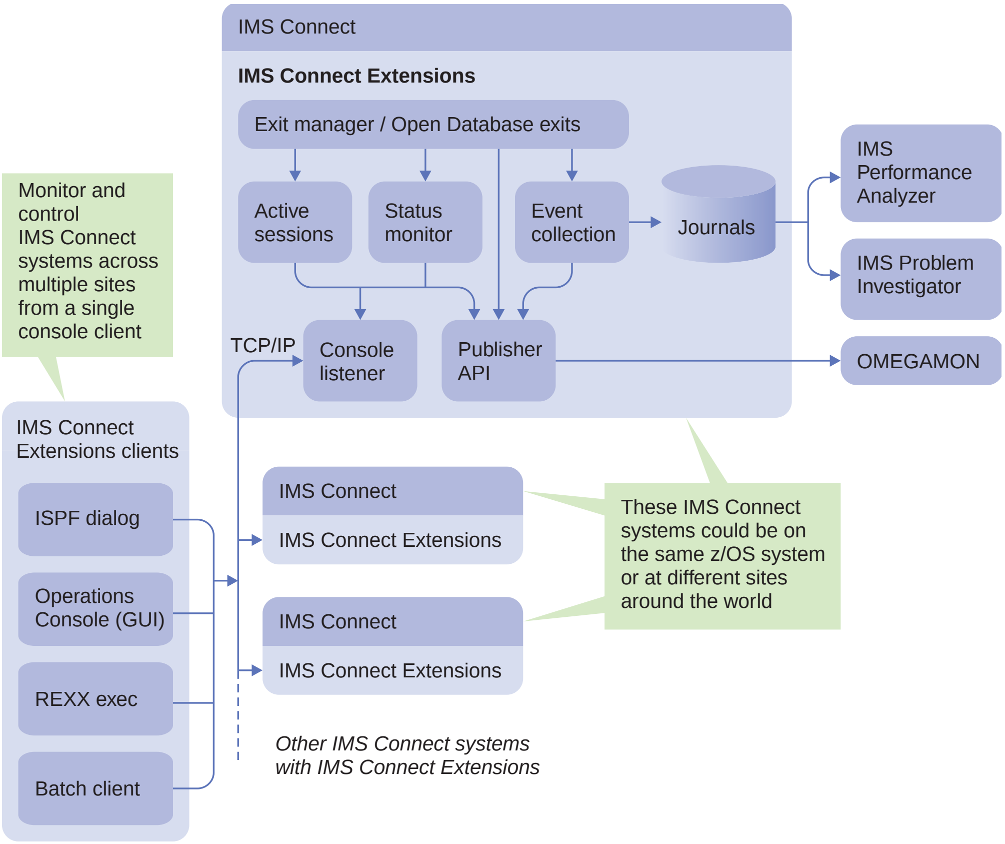

IMS Connect
IMS Connect Extensions
Exit manager / Open Database exits
Active sessions
Status monitor
Event collection
Journals
IMS Performance Analyzer
IMS Problem Investigator
OMEGAMON
TCP/IP
Console listener
Publisher API
Monitor and control IMS Connect systems across multiple sites from a single console client
IMS Connect Extensions clients
ISPF dialog
Operations Console (GUI)
REXX exec
Batch client
IMS Connect
IMS Connect Extensions
IMS Connect
IMS Connect Extensions
These IMS Connect systems could be on the same z/OS system or at different sites around the world
Other IMS Connect systems with IMS Connect Extensions

*Figure 3. IMS Connect Extensions components*

## Architecture

IMS Connect Extensions uses a loosely coupled component-based architecture.

The following topics describe the architecture of the core components.

### Exit manager architecture

The IMS Connect Extensions exit manager is initialized when IMS Connect starts. It loads and manages both IBM-supplied exits and custom exits.

IMS Connect exit routines fall into two general categories: user message exits and function-specific exits. User message exit routines, which are used only with clients connecting to IMS TM, manage messages to and from IMS Connect TCP/IP clients. These exits perform operations such as translation between IRM and OTMA protocols and between ASCII and EBCDIC character code sets. They include:

HWSSMPL0 and HWSSMPL1 (for user-written IMS Connect client applications)

HWSJAVA0 (for the IMS TM Resource Adapter)

HWSSOAP1 (for the IMS Enterprise Suite SOAP Gateway)

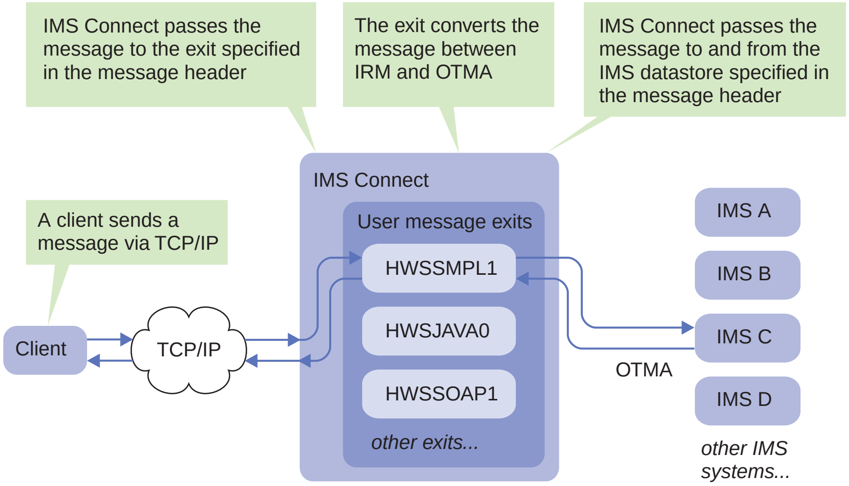
Function-specific exit routines provide general functionality to IMS Connect. These include:

HWSROUT0 (IMS Connect DB Routing user exit routine)

HWSAUTH0 (IMS Connect DB Security user exit routine)

As shown in Figure 4, the use of assembly language message exits ensures that IMS Connect has both high performance and flexibility. However, this performance and flexibility comes as a cost:

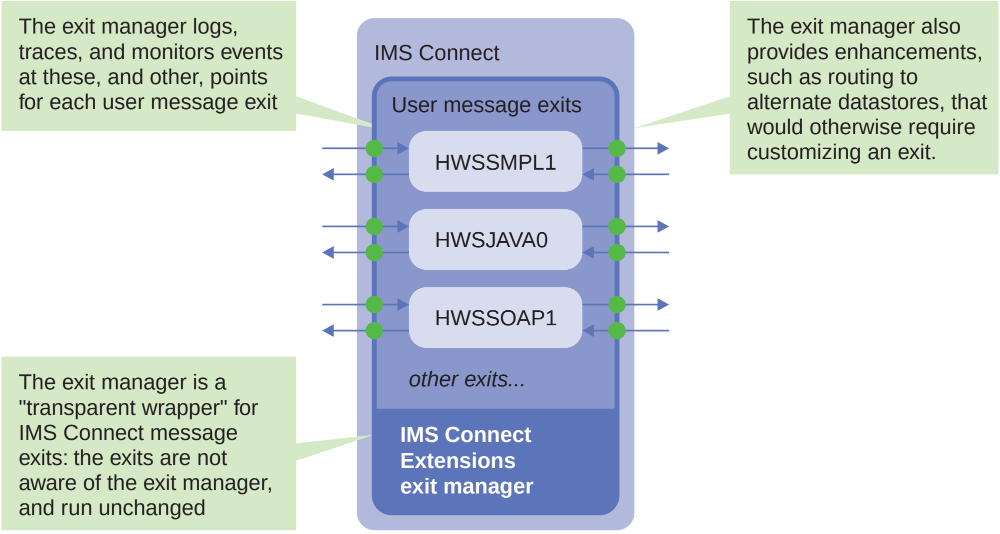
- Customization is difficult.
- The internal workings of the exit are hidden: there is no record of what messages were processed or what was done to them.



IMS Connect passes the message to the exit specified in the message header
The exit converts the message between IRM and OTMA
IMS Connect passes the message to and from the IMS datastore specified in the message header
IMS Connect
A client sends a message via TCP/IP
User message exits
HWSSMPL1
HWSJAVA0
HWSSOAP1
other exits...
Client
TCP/IP
OTMA
IMS A
IMS B
IMS C
IMS D
other IMS systems...

*Figure 4. The role of user exits in message processing without IMS Connect Extensions*

As shown in Figure 5, the IMS Connect Extensions exit manager wraps around both IBM-supplied user message exits and custom exits, extending their capabilities with features such as routing, monitoring, enhanced event collection, and tracing. For Open Database workloads you can replace the IBM-supplied exit routines with more capable IMS Connect Extensions equivalents.



The exit manager logs, traces, and monitors events at these, and other, points for each user message exit
IMS Connect
User message exits
HWSSMPL1
HWSJAVA0
HWSSOAP1
other exits...
IMS Connect Extensions exit manager
The exit manager also provides enhancements, such as routing to alternate datastores, that would otherwise require customizing an exit.
The exit manager is a "transparent wrapper" for IMS Connect message exits: the exits are not aware of the exit manager, and run unchanged

*Figure 5. The exit manager: a transparent wrapper for IMS Connect exits*

This approach has the following benefits:

- IMS Connect Extensions has a complete picture of all activity occurring within the exits, which you can monitor through the Status Monitor and Active Sessions display or view offline in IMS Connect Extensions journals.

- The exit manager can provide additional functions that normally would have to be written in an exit.

You do not need to modify IMS Connect user message exits to work with IMS Connect Extensions. The exit manager is transparent to the exits, which run unchanged.

## Event collection architecture

To create a comprehensive picture of activity within IMS Connect, IMS Connect Extensions collects information directly from IMS Connect and through the exit manager.

As the following figure shows, information about activity in IMS Connect is written to journals, while the activity that occurs within OTMA and IMS is recorded in the IMS logs. Other IBM IMS tools can combine the data to provide an end-to-end picture of activity for a given transaction.

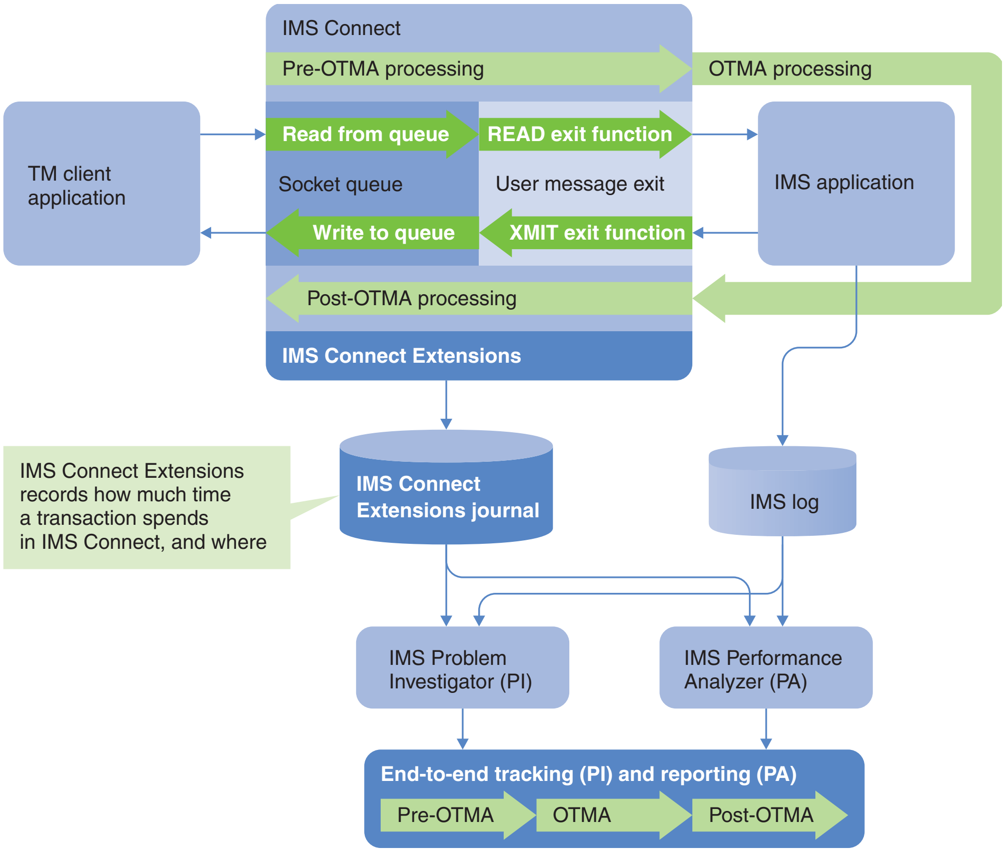


*Figure 6. Event collection architecture: OTMA processing*

The following figure shows the equivalent architecture for ODBM processing.

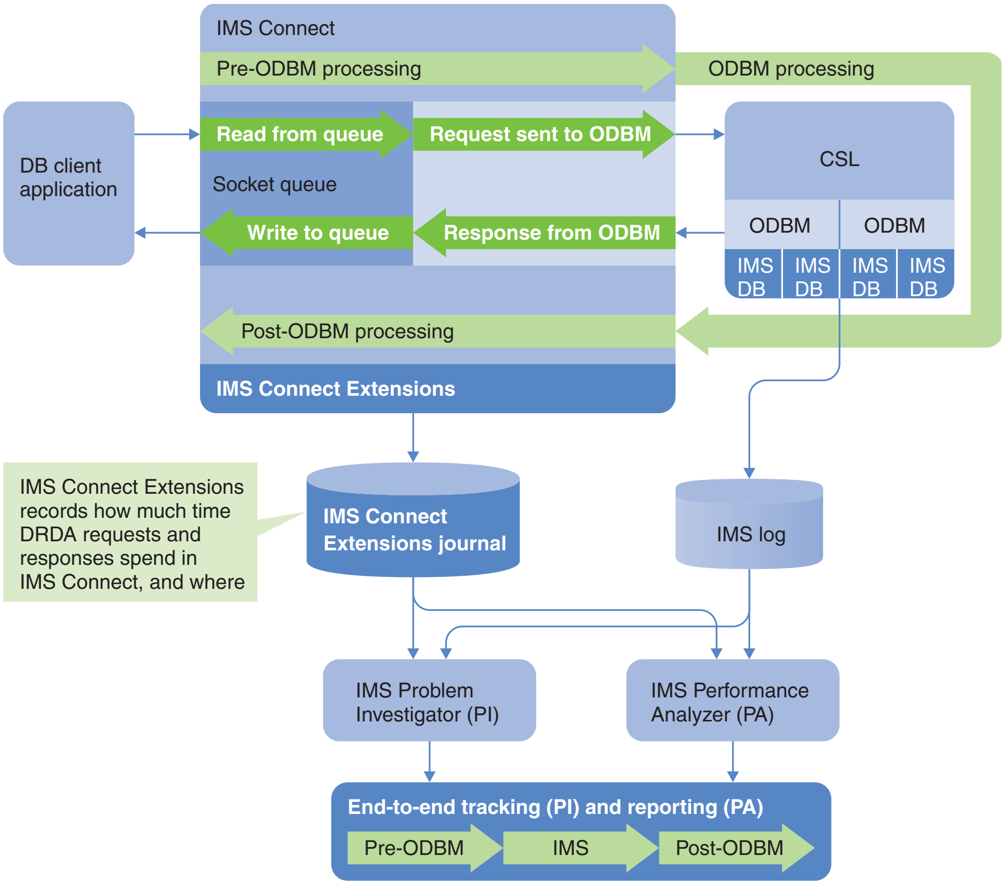


*Figure 7. Event collection architecture: ODBM processing*

**Related concepts**:

Chapter 4, “Event collection,” on page 55
IMS Connect Extensions continuously collects events as incoming messages or DRDA requests are processed or when other changes occur, such as when IMS Connect starts up and shuts down or when a datastore goes offline.

Chapter 5, “Reporting IMS Connect activity,” on page 71
The following topics provide information on how you can use IMS Connect Extensions with IMS PA and IMS PI to report IMS Connect activity.

## Repository architecture

Each instance of IMS Connect with IMS Connect Extensions refers to a repository, which contains configuration data for the IMS Connect Extensions components that start with IMS Connect.

For example, the repository determines the console port and the event collection level.

As shown in the following figure, one repository can contain configuration data for multiple IMS Connect systems. If you have multiple sites, you can copy the same repository to those sites.

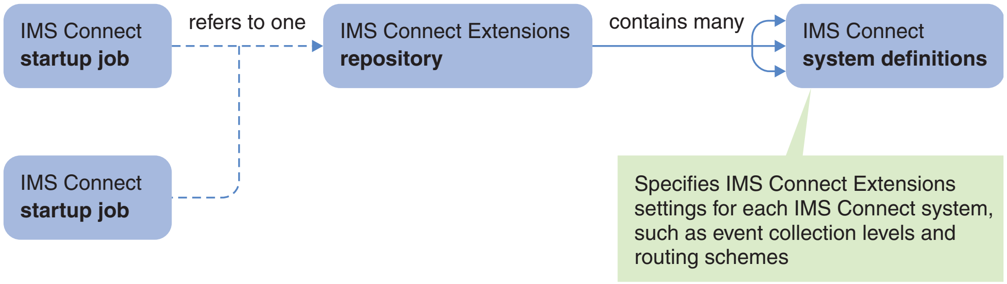

IMS Connect startup job
refers to one
IMS Connect Extensions repository
contains many
IMS Connect system definitions
IMS Connect startup job
Specifies IMS Connect Extensions settings for each IMS Connect system, such as event collection levels and routing schemes

*Figure 8. IMS Connect Extensions repository and IMS Connect configuration*

As shown in Figure 9, an IMS Connect Extensions ISPF dialog can maintain definitions in a repository, and also monitor and control IMS Connect systems by connecting via TCP/IP to one or more console listeners.

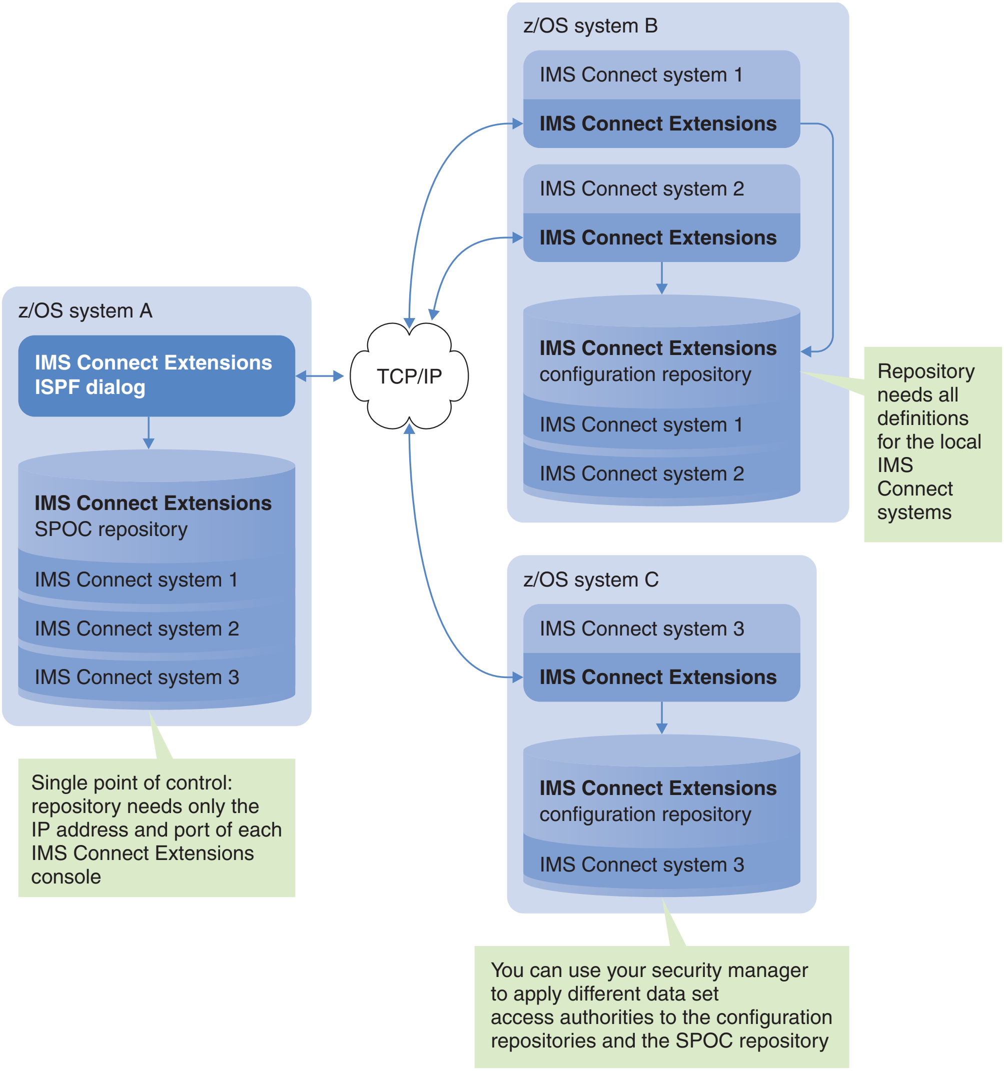

z/OS system B
IMS Connect system 1
IMS Connect Extensions
IMS Connect system 2
IMS Connect Extensions
IMS Connect Extensions configuration repository
IMS Connect system 1
IMS Connect system 2
Repository needs all definitions for the local IMS Connect systems
z/OS system A
IMS Connect Extensions ISPF dialog
TCP/IP
IMS Connect Extensions SPOC repository
IMS Connect system 1
IMS Connect system 2
IMS Connect system 3
Single point of control: repository needs only the IP address and port of each IMS Connect Extensions console
z/OS system C
IMS Connect system 3
IMS Connect Extensions
IMS Connect Extensions configuration repository
IMS Connect system 3
You can use your security manager to apply different data set access authorities to the configuration repositories and the SPOC repository

*Figure 9. IMS Connect Extensions repositories*

To monitor and control an IMS Connect system, the IMS Connect Extensions ISPF dialog does not need to refer to a repository containing a complete definition of that system. All the ISPF dialog needs is a system definition with the IP address and port number of the relevant IMS Connect Extensions console listener.

You could have a dedicated SPOC (single point of control) repository that contains IMS Connect system definitions with only these details, and no other configuration settings. Only the repository referred to by the IMS Connect startup job needs to contain the complete configuration settings for that IMS Connect system.

This single point of control via TCP/IP does not include editing definitions in remote repositories. To edit definitions in a repository, the ISPF dialog needs direct access to the repository data set. When direct data set access is not available, you can centralize the administration of IMS Connect Extensions configuration by using the batch definition utilities.

**Related concepts**:

Chapter 19, “Administering definitions with the ISPF dialog,” on page 187
This section describes how to specify IMS Connect Extensions definitions using the ISPF dialog.

Chapter 22, “Definition utilities,” on page 269
IMS Connect Extensions provides utilities to administer its definitions.

## IMS Connect Extensions utilities

IMS Connect Extensions includes several utilities that help you to analyze logs, manage IMS Connect, and manage IMS Connect Extensions.

**CEX host command environment for REXX**

The CEX host command environment for REXX enables IMS Connect Extensions commands to be embedded in REXX programs, which allows more flexible automation of IMS Connect operations. You can use this to perform tasks such as suspending or resuming routing to a datastore, switching the active journal, initiating tracing, refreshing and updating definitions, and running commands on a target system or datastore.

- Programs can take advantage of REXX features such as variables, conditional logic, and integration with other host environments.
- Programs can connect to multiple IMS Connects.
- Programs can be submitted interactively as well as in batch.

**Definition utilities**

These utilities import, export, and modify IMS Connect Extensions definitions in batch. You can use these utilities to:

- Create an initial configuration for IMS Connect Extensions
- Export definitions to various global sites, or create an import file in CSV format that can be used to distribute a standard configuration to Operations Console users
- Transform IMS Connect Extensions definitions from test systems to production systems

**Archiving utilities**

These utilities allow you to perform manual archiving and clean up archive journals.

**Reporting utilities**

These utilities format and analyze IMS Connect event data:

**Print utility**

Provides basic formatting and analysis of IMS Connect Extensions journals.

**Active session utility**

Analyzes an IMS Connect Extensions journal and then identifies and reports on TCP/IP sessions that have not ended.

**RECORDER trace utilities**

These convert RECORDER trace records produced by IMS Connect into IMS Connect Extensions format and can also print the data from these records.

**Installation verification programs**

The installation verification programs run client sessions against IMS Connect, directly from the mainframe. There are separate utilities based on OTMA and Open Database workloads. These utilities verify the installation of IMS Connect and IMS Connect Extensions, and the OTMA utilities allow you to perform basic stress testing.

**Client services exit**

The client services exit is an additional user exit supplied with IMS Connect Extensions. The exit runs like other IBM sample exits included with IMS Connect. Clients can specify the exit and then receive additional services that provide password change, user verification, diagnostic information, and logging of user data.

## Related IBM tools

IMS Connect Extensions combines with other IMS tools to enable you to report on and analyze IMS Connect throughput and performance and to diagnose problems.

**IBM IMS Performance Analyzer for z/OS**

IMS Performance Analyzer integrates information collected by IMS Connect Extensions with information in the IMS logs to provide an overall picture of transactional throughput and performance. IMS PA can automatically select the relevant IMS Connect Extensions archive journals and merge them with the relevant IMS logs for a given reporting period. It can generate a transaction summary called a transaction index to make problem and pattern analysis easier.

**IBM IMS Problem Investigator for z/OS**

IMS Problem Investigator is an intelligent log analysis tool that can help you use the IMS logs for tasks such as debugging, performance tuning, tracing, and creating audit trails. IMS PI merges information in the IMS logs with information collected by IMS Connect Extensions, allowing you to dissect individual transactions as they progress from TCP/IP to OTMA to IMS and back, or individual DRDA requests as they progress from TCP/IP via ODBM to IMS and back. Like IMS PA, IMS PI automatically selects IMS Connect Extensions archive journals and IMS logs.

**OMEGAMON for IMS on z/OS**

IMS Connect Extensions enables you to gain a real-time graphical view of TCP/IP sessions and performance throughput using OMEGAMON.

**Related concepts**:

"Reporting and analysis with IMS Performance Analyzer" on page 71
In-depth analysis and reporting of IMS Connect event records is provided by the IBM IMS Performance Analyzer.

## IMS Connect Extensions terminology

This topic defines some terms that occur frequently in IMS Connect Extensions information.

**Event collection**
The process of recording IMS Connect activity. This includes collecting information about IMS Connect operations, the state of clients connecting to IMS Connect, and details of the transactions processed through IMS Connect.

**IMS Connect client**
A program that starts a session with IMS Connect in order to access an IMS application or database. An IMS Connect client typically connects via TCP/IP, in the case of an OTMA client using its own implementation of the IRM protocol, or in the case of an Open database client using DRDA. Clients can also connect via middleware such as WebSphere® or the IMS SOAP gateway.

In the case of IMS-to-IMS MSC and OTMA TCP/IP communications, IMS Connect can become the client of another IMS Connect system.

**IMS Connect Extensions client**
A program that starts a session with IMS Connect Extensions to perform task automation, monitoring, configuration, or control functions. The ISPF Operations dialog, GUI operations console, and jobs that invoke the IMS Connect Extensions host command environment for REXX are called console clients because they connect to IMS Connect Extensions console listeners.

**IMS Connect Extensions definitions**
You configure and customize IMS Connect Extensions by creating definitions that reflect your real-world IMS topology. For example, for each IMS Connect system, you create a definition that specifies the features of IMS Connect Extensions you want that system to use. You can define routing rules to be used for routing IRM messages or DRDA requests. For each user exit, you can specify custom message translation or additional debugging information.

IMS Connect Extensions also has definitions to help manage and organize your IMS components into groups that reflect your business requirements. Such topologies include your sysplex configuration, environment (for example, development environment versus production environment), or system function (for example, accounting, inventory, or payroll).

**IMS Connect system**
The IMS Connect function of IMS runs in a separate address space from IMS itself. Each such instance of IMS Connect is referred to in this information as an "IMS Connect system". Unless otherwise specified, an IMS Connect system includes the IMS Connect Extensions components that run with it.

**OTMA routing**
The process of dynamically changing the target datastore for a transaction. IMS Connect Extensions can generate a list of candidate datastores based on the transaction type, transaction code, and datastore specified in the message; it can then select one of those candidates to route the message to. You can use routing to automate failover for clients and response to OTMA flood conditions, and for workload balancing, improved parallelism (creating multiple TCBs from a single DESTID), and additional security.

**ODBM routing**

The process of dynamically changing the ODBM target for a DRDA request. IMS Connect Extensions can generate a list of candidate ODBM names and aliases based on the alias or PSB name specified in the message; it can then select one of those candidates to route the request to.

**Workload balancing**

The process of using routing to automatically balance transaction workloads across multiple datastores, and to balance DRDA requests across multiple databases. IMS Connect Extensions can select a datastore or ODBM target from the list of candidates based on either equal probabilities or weighted probabilities.

## Application management solutions

IBM solutions help IT organizations maximize their investment in DB2® and IMS databases while staying on top of some of today's toughest IT challenges. Application Management solutions can help maximize the productivity and profitability of your DB2 and IMS databases.

IMS Connect Extensions is one of several IMS Tools products that help application developers to achieve their goal of delivering well written applications that maximize performance.

The following additional IMS Tools products also provide Application Management solutions:

- IMS Batch Terminal Simulator
- IMS Batch Backout Manager
- IMS Program Restart Facility

IMS Connect Extensions improves the availability, reliability, and performance of IMS Connect, enabling dynamic management of TCP/IP transactions and an enhanced level of monitoring and control.

IMS Connect Extensions complements IMS Performance Analyzer for z/OS and IMS Problem Investigator for z/OS, making the end-to-end analysis of IMS Connect problems and performance quicker and easier than ever before. With Tivoli® OMEGAMON XE for IMS on z/OS it enables you to view graphical real-time reports of TCP/IP activity.

The result is improved productivity for problem analysts, more efficient TCP/IP application performance, improved IMS resource utilization, and simpler deployment of WebSphere and SOAP clients.

## Service updates and support information

Service updates and support information for this product, including software fix packs, PTFs, Frequently Asked Question (FAQs), technical notes, troubleshooting information, and downloads, are available from the Web.

To find service updates and support information, see the following web page:

http://www.ibm.com/support/entry/portal/Overview/Software/Information_Management/IMS_Tools

## IMS Connect Extensions documentation and updates

IMS Tools information is available at multiple places on the web. You can receive updates to IMS Tools information automatically by registering with the IBM My Support service.

### IMS Connect Extensions information on the web

The IMS Tools Product publications web page provides current product documentation that you can view, print, and download. To locate publications with the most up-to-date information, refer to the following web page:

http://www.ibm.com/software/data/db2imstools/imstools-library.html

You can also access documentation for many IMS Tools from the Information Management Software for z/OS Solutions Information Center:

http://publib.boulder.ibm.com/infocenter/imzic

IBM Redbooks® publications that cover IMS Tools are available from the following web page:

http://www.redbooks.ibm.com

The Data Management Tools Solutions website shows how IBM solutions can help IT organizations maximize their investment in IMS databases while staying ahead of today's top data management challenges:

http://www.ibm.com/software/data/db2imstools/solutions/index.html

### Receiving documentation updates automatically

To automatically receive automated emails that notify you when new technote documents are released, when existing product documentation is updated, and when new product documentation is available, you can register with the IBM My Notifications service. You can customize the service so that you receive information about only those IBM products that you specify.

To register with the My Notifications service:

1. Go to http://www.ibm.com/support/mysupport
2. Enter your IBM ID and password, or create one by clicking **register now**.
3. When the My Notifications page is displayed, click **Subscribe** to select those products that you want to receive information updates about. The IMS Tools option is located under **Software** > **Information Management**.
4. Click **Continue** to specify the types of updates that you want to receive.
5. Click **Submit** to save your profile.

## Accessibility features

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

The major accessibility features in IMS Connect Extensions enable users to:

- Use assistive technologies such as screen readers and screen magnifier software. Consult the assistive technology documentation for specific information when using it to access z/OS interfaces.
- Customize display attributes such as color, contrast, and font size.
- Operate specific or equivalent features by using only the keyboard. Refer to the following publications for information about accessing ISPF interfaces:
  - *z/OS ISPF User's Guide, Volume 1*
  - *z/OS TSO/E Primer*
  - *z/OS TSO/E User's Guide*

These guides describe how to use ISPF, including the use of keyboard shortcuts or function keys (PF keys), include the default settings for the PF keys, and explain how to modify their functions.

# Chapter 2. Installing and customizing IMS Connect Extensions

After you SMP/E install IMS Connect Extensions, there are a number of steps you perform to install it in IMS Connect and then customize it for your environment and workloads.

The initial configuration steps include taking up system and datastore definitions from the IMS Connect configuration member using the takeup utility, authorizing the load libraries and `MODIFY` command, and modifying the IMS Connect startup job. You must complete these tasks before running IMS Connect Extensions. Installation verification procedures are provided to help confirm that the basic configuration is correct.

Optional steps include customizing control options, workload routing rules, event collection, and message translation, and installing IMS Connect Extensions Operations Console on distributed workstations.

**Important:** Before you can customize IMS Connect Extensions, it must be installed successfully. Instructions for installing IMS Connect Extensions are provided in the program directory.

## Customization overview

These topics describe information you might need to prepare and steps you might need to perform to install and customize IMS Connect Extensions to run under IMS Connect.

Steps are required unless noted otherwise.

| Step | Description | Instructions |
|---|---|---|
| 1 | Determine the type of customization that is appropriate for your environment and gather required information. | “Planning your customization” on page 28 |
| 2 | Review the default control option settings to see whether they best suit your requirements. | “Reviewing control options” on page 30 |
| 3 | Start the ISPF interface and configure the definitions repository and load libraries.<br>Use the Definitions Setup utility to create a starting set of definitions in the IMS Connect Extensions repository based on the IMS Connect configuration member. | “Creating an initial configuration for IMS Connect Extensions” on page 31 |
| 4 | APF-authorize the IMS Connect Extensions load libraries and the MVS™ `MODIFY` command. | “APF-authorizing load libraries and the MODIFY command” on page 34 |
| 5 | Modify the IMS Connect startup job to include IMS Connect Extensions libraries and data sets. | “Modifying and submitting the IMS Connect startup job” on page 35 |
| 6 | Verify that IMS Connect Extensions is installed in IMS Connect and is accessible from the Operations dialog. | “Viewing the IMS Connect system in IMS Connect Extensions” on page 36 |
| 7 | If you are upgrading from an earlier version of IMS Connect Extensions, see if you need to perform any additional steps. (Optional) | “Upgrading to IMS Connect Extensions V2.4” on page 43 |
| 8 | Install Operations Console plug-in on distributed workstations. | “Install IMS Connect Extensions Operations Console” on page 43 |

### Configuring routing and security

This table provides links to information on customizing IMS Connect Extensions to route OTMA or Open Database workloads and to enhance IMS Connect security.

If you are likely to use an IMS Connect instance for Open Database workloads, then install the Open Database routing and security exits. This will best prepare your system to take advantage of IMS Connect Extensions features.

| Feature/workload | Optional or workload-dependent customization | Instructions |
|---|---|---|
| OTMA routing | Verify that communications are configured to allow monitoring and routing of IMS request messages through TCP/IP connections. | "Running IVP jobs to generate sample OTMA workload" on page 37 |
| OTMA routing | Configure OTMA routing rules and datastores; customize routing of IRM messages. | Chapter 14, "Routing OTMA messages and balancing workloads," on page 139 |
| Required for Open Database routing | Configure the IMS Connect BPE environment to use the CEXROUT0 exit. | "Configuring the ODBMROUT exit member" on page 30 |
| Open Database; security (optional) | Configure the IMS Connect BPE environment to use the CEXAUTH0 exit to perform user authentication for connections to IMS DB. | "Configuring the Open Database security exit" on page 31 |
| Open Database routing | Verify that communications are configured to allow monitoring and routing of DRDA requests through TCP/IP connections. | "Running IVP jobs to generate sample Open Database workload" on page 39 |
| Open Database routing | Configure ODBM routing rules and targets; customize routing of DRDA requests. | Chapter 15, "Routing Open Database requests," on page 167 |
| IMS Connect security | Review and configure IMS Connect Extensions enhancements to IMS Connect security. | Chapter 16, "Enhancing IMS Connect security," on page 173 |

### Additional customization

This table provides links to information on other common steps you might want to perform to customize IMS Connect Extensions for your environment and workloads.

| Feature/workload | Optional or workload-dependent customization | Instructions |
|---|---|---|
| Event collection | Customize event collection and journaling. For example, control what events are recorded and create the archive JCL member. | "Configuring event collection and journals: overview" on page 61 |
| All | Automate IMS Connect operations tasks. | Chapter 11, "Automating operations tasks using REXX," on page 123 |
| OTMA | Change how IMS Connect translates messages received from clients between character sets. | Chapter 18, "Customizing IMS Connect message translation," on page 183 |
| All; security | Customize access controls to IMS Connect z/OS and WTOR commands and IMS Connect Extensions commands. | "Controlling access to commands in your external security manager" on page 130 |
| All | Customize the ISPF dialog. | "Advanced ISPF dialog configuration" on page 41 |

## Hardware and software prerequisites

This section lists the requirements for installing and running IMS Connect Extensions and IMS Connect Extensions Operations Console.

### Installation requirements

To install IMS Connect Extensions you require the following:

- 5694-A01 z/OS, V1.13, 5650-ZOS z/OS, V2.1, or later
- 5655-G44 SMP/E for z/OS, V3.5 or later

### Hardware requirements

IMS Connect Extensions for z/OS operates on any hardware configuration that supports the required version of IMS.

### Software requirements

IMS Connect Extensions is designed to operate with any supported IMS version.

For IMS Connect Extensions Operations Console to operate, IMS Connect Extensions 2.3.0 or later must be installed on all IMS Connect systems that you want to connect to.

IMS Connect Extensions Operations Console is an Eclipse plug-in. It is designed to operate on Windows platforms supported by z/OS Explorer Version 2.1 or later.

## IMS Connect Extensions product libraries

The components of the IMS Connect Extensions dialog are delivered in a set of product libraries.

The following figure illustrates which libraries are used by each IMS Connect Extensions component.

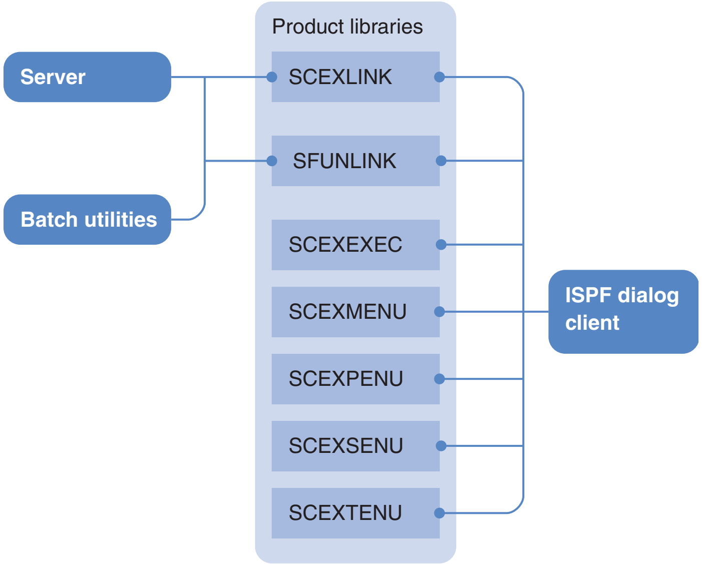


*Figure 10. Product libraries used by each IMS Connect Extensions component*

**Note:** IMS Connect Extensions programs must either be located in libraries specified in a STEPLIB DD card or in a link-listed data set.

### Load module libraries

**SCEXLINK**
: Application load module library for the IMS Connect Extensions server components and ISPF client. This library contains components that run in the same address space as IMS Connect so it must be an authorized (APF) library.

**SFUNLINK**
: Functional support load module library for the IMS Connect Extensions server components and ISPF client. The Functional Support Library contains common function that is used in a number of IMS Tools. This library contains components that run in the same address space as IMS Connect so it must be an authorized (APF) library.

**Note:** Common Services Library (CSL) provides equivalent services to those provided by the Functional Support Library. If you already have CSL or if you need to install it, IMS Connect Extensions will run with it instead of Functional Support Library. That is, IMS Connect Extensions will work with either CSL110 or FUN240.

### Libraries used only by the ISPF dialog client

**SCEXEXEC**
: REXX EXECs

**SCEXLINK**
: Executable load modules

**SCEXMENU**
: ISPF messages

**SCEXPENU**
: ISPF panels

**SCEXTENU**
: ISPF input tables

**SCEXSENU**
: ISPF skeletons

### Sample library

**SCEXSAMP**
: Includes skeleton JCL for journal archiving, IVP programs, sample REXX execs, and automated configuration scripts.

## Planning your customization

To simplify customization, determine the type of customization that is appropriate for your workload and environment, and gather the information that you will need during the customization process.

Print the following tables for each IMS Connect system you intend to enable with IMS Connect Extensions, and write the values for your installation in the “Enter your value” column.

### Product libraries

These values are used to specify DD statements relating to IMS Connect Extensions in the IMS Connect startup job.

| Item | Enter your value | Installation parameter/DD statement |
|---|---|---|
| IMS Connect Extensions load libraries (SMP/E-controlled) | (Default: CEX.V2R4M0.SCEXLINK) | |
| Functional support load libraries (SMP/E-controlled) | (Default: CEX.V2R4M0.SFUNLINK) | |
| Name of the target definitions repository you will be creating | | *your.repos* in //CEXREPOS DD |
| Name of the control options data set you will be creating (optional) | | *your.options.ds* in //CEXCTLIN DD |

## Client connections

These values are used to establish connections from the ISPF Operations dialog clients, Operations Console clients, REXX execs, and batch clients to the IMS Connect Extensions console listener.

| Item | Enter your value | Installation parameter name |
|---|---|---|
| Host name or IP address of the server IMS Connect is running on | | |
| Unallocated TCP/IP port for the IMS Connect Extensions console listener | | |
| For installation verification: any ports that IMS Connect is listening on which you want to test using the IVP utilities:<br>• OTMA workload: IMS Connect message ports<br>• Open Database workload: DRDA ports<br>If you are using security you also will need to supply an authorized user ID and password | | |

## IMS Connect startup job

These values are used to modify the IMS Connect startup job to also start IMS Connect Extensions.

| Item | Enter your value | Installation parameter name |
|---|---|---|
| Name of PROCLIB data set containing the IMS Connect configuration member (HWSCFG) | | |
| PDS member name of the configuration member HWSCFG | | |
| IMS Connect startup job and write-access to that job | | |
| HLQ to be used to create the archive and active journal data sets (optional) | | |
| PDS and member name of the Archive JCL skeleton (optional) | | |

## Reviewing control options

The control input data set enables selected configuration options to be provided when IMS Connect Extensions starts up. Review the default settings to see whether they best suit your requirements.

Control options can be used to tune for performance:

- ZIIP_OFFLOAD determines if event collection processing is to be offloaded to available ZIIP processors. To enable ZIIP_OFFLOAD, specify:
  ```
  ZIIP_OFFLOAD OPTION=Y
  ```
- The EVENTLOGGING option can be changed to optimize performance. For example, to buffer event records before they are written to the journal, specify:
  ```
  EVENTLOGGING WRITE=BUFFER
  ```
- CEXTRACE sets the size of the trace buffer. For XML workloads you might need to set this value higher to see the complete payload. For example, to increase the trace buffer size to 5120 bytes (5 * 1024), specify:
  ```
  CEXTRACE BUFSIZE=5
  ```

The SECURITY control option affects how user credentials are processed and cached when IMS Connect Extensions security is activated. For example, the PWCASE keyword determines how mixed-case passwords are handled. To obtain the mixed-case password setting from the external security manager, specify:

```
SECURITY PWCASE=AUTOMATIC
```

Other control options, such as CEXROUTE, ODBMDEFAULTS, and AUTOADD_DATASTORE, affect routing behavior.

The SCEXSAMP library includes a sample member named CEXCTL01. You can use this file to explore the effect of customizing different configuration options.

For details and usage information for all of the available control options, see Chapter 32, "Control input data set options," on page 443.

## Configuring the ODBMROUT exit member

If you want to use IMS Connect Extensions to route DRDA request traffic, specify CEXROUT0 as the ODBMROUT exit member in the BPE configuration member.

### About this task

The IMS Connect Extensions routing exit CEXROUT0 provides advanced features to control the routing of incoming requests from clients that access IMS DB. CEXROUT0 replaces the IMS Connect user-written DB Routing Exit routine (HWSROUT0). If you want to use IMS Connect Extensions to route DRDA request traffic, CEXROUT0 must be added to the BPE exit list and specified as the ODBMROUT exit member. The exit member is specified in the ODBMROUT EXITDEF statement in the PROCLIB member pointed to by the EXITMBR statement in the BPECFG member.

### Procedure

1. Specify the name of the exit list member in the BPE configuration parameter member (BPECFG*xx*):
   ```
   EXITMBR=(HWSEXITN,HWS)
   ```
2. Include the following statement in the exit list member (HWSEXITN):

```
EXITDEF(TYPE=ODBMROUT,EXITS=(CEXROUT0),ABLIM=8,COMP=HWS)
```

### Results

CEXROUT0 will execute as the ODBM routing exit for IMS Connect.

**Related tasks**:

"Implementing ODBM routing" on page 169
Implementing ODBM routing involves analyzing your DRDA request traffic, configuring routing plans, routing rules, targets, and PSB name lists, and then activating ODBM routing on required systems.

## Configuring the Open Database security exit

CEXAUTH0 is the IMS Connect Extensions ODBM security exit. When installed it enables for Open Database workloads the same access control options as for OTMA workloads.

### About this task

CEXAUTH0 replaces HWSAUTH0, the IMS Connect DB security user exit routine.

### Procedure

If you require connections to IMS DB to be authenticated by IMS Connect Extensions, follow these steps to add CEXAUTH0 to the BPE exit list:

1. Specify the name of the exit list member in the BPE configuration parameter member (BPECFGxx):
   ```
   EXITMBR=(HWSEXITN,HWS)
   ```
2. Include the following statement in the exit list member (HWSEXITN):
   ```
   EXITDEF(TYPE=ODBMAUTH,EXITS=(CEXAUTH0),ABLIM=8,COMP=HWS)
   ```
3. Ensure that the ODACCESS statement is specified in the IMS Connect configuration member (HWSCFGxx).

### Results

CEXAUTH0 will execute as the ODBM security exit for IMS Connect.

### What to do next

To enable validation on an IMS Connect system, open the system definition, select **Activate Security** and **Activate validation**, and specify **Validation type**.

**Related concepts**:

"Validating an incoming OTMA transaction or DRDA request" on page 174
If security validation is activated, IMS Connect Extensions will check whether the user ID associated with an incoming message or DRDA request is authorized.

## Creating an initial configuration for IMS Connect Extensions

Use the Definitions Setup dialog to take up definitions from the IMS Connect configuration member into a repository. This provides you a basis from which you can start customizing IMS Connect Extensions.

## Before you begin

- Follow the installation instructions in the Program Directory. For details of the IMS Connect Extensions product libraries, see "IMS Connect Extensions product libraries" on page 27.
- If journal archiving is required, the Archive JCL member must already exist.

**Note:** The examples and terminology in this topic assume you are using RACF®. However, you can use any SAF-conforming security server.

## Procedure

1. Go to the ISPF command shell panel (ISPF primary menu option 6), and enter: `EX '`*`cexpre`*`.SCEXEXEC(CEXOREXX)' '`*`cexpre`*` ENU'` where *cexpre* is the high-level qualifier for the IMS Connect Extensions product libraries.

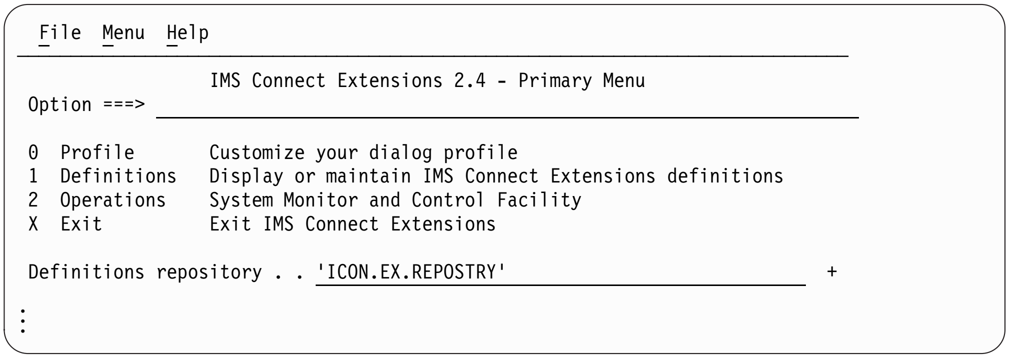
   For example:

   ```
   EX 'CEX.V2R4M0.SCEXEXEC(CEXOREXX)' 'CEX.V2R4M0 ENU'
   ```

   You can either allow the IMS Connect Extensions libraries to be set up dynamically each time you start the IMS Connect Extensions dialog (as done in the previous example), or you can add the libraries statically to the relevant ISPF library concatenations. Then you can optionally add IMS Connect Extensions to an ISPF menu. See "Advanced ISPF dialog configuration" on page 41 for details.

   Dynamic setup is the simplest and quickest approach.

2. On the IMS Connect Extensions primary menu, enter the name of the Definitions repository. Use the information prepared in "Product libraries" on page 28.



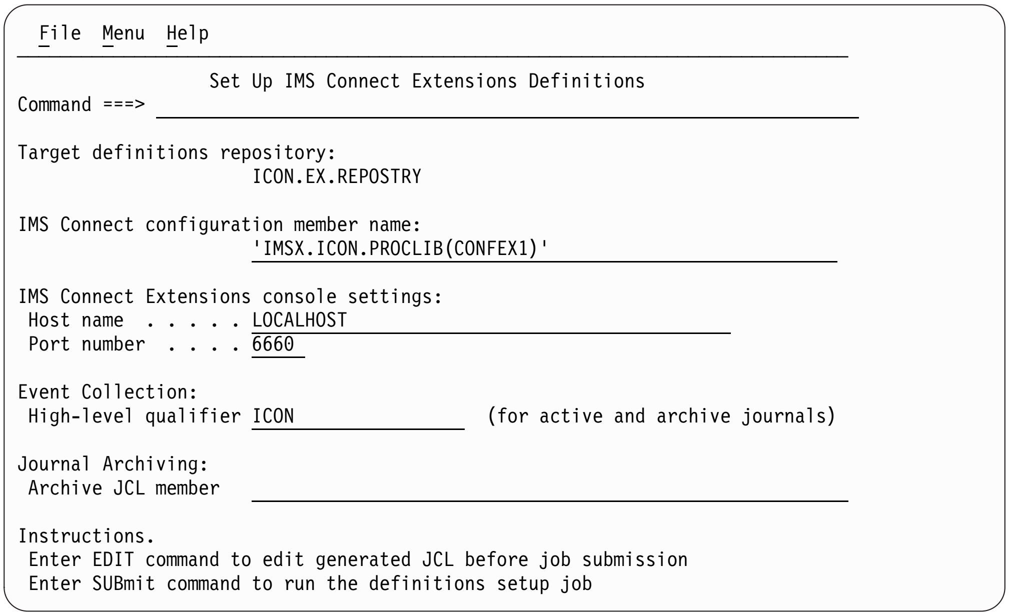
```
 File  Menu  Help
──────────────────────────────────────────────────────────────────────────────
                 IMS Connect Extensions 2.4 - Primary Menu
Option ===> ____________________________________________________________

0  Profile       Customize your dialog profile
1  Definitions   Display or maintain IMS Connect Extensions definitions
2  Operations    System Monitor and Control Facility
X  Exit          Exit IMS Connect Extensions

Definitions repository . . 'ICON.EX.REPOSTRY'______________________________  +
⋮
```

*Figure 11. Specify the Definitions repository*

   **Note:** If the repository does not exist and you try to access options for any definitions, IMS Connect Extensions will automatically prompt you to define a repository.

3. Select option 0 **Profile**. Enter the IMS Connect Extensions load library and functional support library using the information prepared in "Product libraries" on page 28. Review the other profile settings such as the time zone and warning override.
4. Select option 1.S **Definitions Setup**.

   Complete the fields in the dialog using the information you prepared in "IMS Connect startup job" on page 29 and the host name and console listener port in "Client connections" on page 29.



```
  File  Menu  Help
------------------------------------------------------------------------------
                  Set Up IMS Connect Extensions Definitions
Command ===> __________________________________________________________________

Target definitions repository:
                      ICON.EX.REPOSTRY

IMS Connect configuration member name:
                      'IMSX.ICON.PROCLIB(CONFEX1)'

IMS Connect Extensions console settings:
 Host name  . . . . . LOCALHOST
 Port number  . . . . 6660

Event Collection:
 High-level qualifier ICON                (for active and archive journals)

Journal Archiving:
 Archive JCL member

Instructions.
 Enter EDIT command to edit generated JCL before job submission
 Enter SUBmit command to run the definitions setup job
```

*Figure 12. Specify the Definitions Setup fields*

5. Enter the **EDIT** command to view or tailor the generated JCL before job submission, then enter the **SUB** command to submit the job. The JCL performs the following steps:
   - Creates an IMS Connect Extensions definitions repository using IDCAMS (DEFINE step).
   - Runs the IMS Connect Extensions definition takeup utility (TAKEUP step) to read the IMS Connect configuration member and create definition maintenance utility commands.
   - Run these definition maintenance utility commands to create definitions in the repository.
6. Look in the CEXPRINT output from step 5 for this message:

```
CEX5501I Updates completed successfully
```

   This message confirms that IMS Connect Extensions has configured definitions for the IMS Connect system, its datastores, and its exits. Do not proceed until you have seen this message.

   If you do not see this message or if there is no CEXPRINT output for this step:
   - Review the messages in the JES message log.
   - Review the instructions and resubmit. Common errors include:
     - Insufficient permissions to create the IMS Connect Extensions repository.
     - Using the wrong high-level qualifiers of the IMS Connect Extensions load libraries.
7. Select option 1.1 **System Definitions**. Look for a system whose name is equal to the value of the ID parameter from the IMS Connect configuration member.

## Results

The setup job has created an example configuration based on your current environment and on the settings you specified in step 5. This configuration includes providing access to advanced features such as commands and monitoring.

### What to do next

Later, you can create additional systems or modify settings for this system using the ISPF dialog. The TAKEUP step can be run multiple times for different configuration members and targeting the same or a different repository.

**Tip:** From the system definition panel, you can use the command:

```
NEW hwsid2 MODEL hwsid1
```

where *hwsid2* is the value of the ID parameter for the new IMS Connect system and *hwsid1* is the name of an existing system.

You only need to change the value for the IMS Connect Extensions console port, as this must be an unassigned TCP/IP port. However, if your systems use different or additional exits you will also need to define them.

## APF-authorizing load libraries and the MODIFY command

This topic explains how to APF-authorize the IMS Connect Extensions load libraries and the MVS **`MODIFY`** command. These steps enable IMS Connect Extensions to work with IMS Connect and to issue commands to the IMS Connect address space.

### About this task

IMS Connect Extensions runs in the same address space as IMS Connect. Therefore, the IMS Connect Extensions programs must be executed from an authorized (APF) library.

The command shell and some dialog line actions use the MVS **`MODIFY`** command to issue commands to IMS Connect. Therefore, the address space that is running IMS Connect must be authorized to issue the **`MODIFY`** command. If this is not done, attempting to issue commands to IMS Connect might result in a message `IEE345I MODIFY AUTHORITY INVALID` being displayed.

### Procedure

1. Include the following load libraries as part of your authorized list:
   - SCEXLINK
   - SFUNLINK
2. In your SAF-compliant external security manager, grant the MVS MODIFY command access to the address space. For example, if you are using RACF as your security manager, perform the following procedure:
   a. Define a resource profile in the OPERCMDS class with one of the following names for the IMS Connect address space:

      For a job: `MVS.MODIFY.JOB.`*`jobname`*

      For a started task: `MVS.MODIFY.STC.`*`mbrname`*`.`*`jobname`*

   b. Grant UPDATE authority to the users or groups that are authorized to run this IMS Connect address space.

   For a complete description of this process, see the section about Planning Console Security in *MVS Planning Operations*.

# Modifying and submitting the IMS Connect startup job

This topic explains how to modify the IMS Connect startup job to start IMS Connect with IMS Connect Extensions.

## About this task

The following figure shows an example JCL to start IMS Connect with IMS Connect Extensions.

```
//HWS       PROC  RGN=4096K,SOUT=A,
//             BPECFG=BPECFGHT,
//             HWSCFG=HWSCFG00
//*
//**********************************************************************
//*  BRING UP IMS CONNECT WITH EXTENSIONS                              *
//**********************************************************************
//STEP1    EXEC PGM=HWSHWS00,REGION=&RGN,TIME=1440,
//             PARM='BPECFG=&BPECFG,HWSCFG=&HWSCFG'
//STEPLIB  DD   DSN=CEX.V2R4M0.SCEXLINK,DISP=SHR, 1
//         DD   DSN=FUN.V2R4M0.SFUNLINK,DISP=SHR,
//         DD   DSN=SDFSRESL,DISP=SHR
//              existing libraries
//              ...
//PROCLIB  DD   DSN=USER.PROCLIB,DISP=SHR
//SYSPRINT DD   SYSOUT=&SOUT
//SYSUDUMP DD   SYSOUT=&SOUT
//HWSRCORD DD   DSN=HWSRCDR,DISP=SHR
//CEXREPOS DD   DSN=your.repos,DISP=SHR 2
//CEXCTLIN DD   DSN=your.options.ds,DISP=SHR 3
//CEXPRINT DD   SYSOUT=&SOUT 4
```

*Figure 13. Startup JCL for IMS Connect with IMS Connect Extensions*

## Procedure

1. Modify the IMS Connect startup job to include the IMS Connect Extensions load libraries, definitions repository, and optional control input data set.
   a. Put the IMS Connect Extensions product load library and IMS Connect Extensions functional support load library in front of the IMS Connect load library ( 1 ). See "Load module libraries" on page 28.

      **Important:** IMS Connect Extensions uses its own version of HWSTECL0. If you have modified HWSTECL0 and are now using IMS Connect Extensions, you must place your HWSTECL0 load libraries after the IMS Connect Extensions product and functional support load libraries.
   b. Add the CEXREPOS DD card with the name of your IMS Connect Extensions repository ( 2 ). The repository stores IMS Connect Extensions definitions and other configuration data.
   c. Optionally, add the CEXCTLIN DD card with the IMS Connect Extensions control input data set ( 3 ). The control input data set enables selected options to be provided when IMS Connect Extensions restarts.
   d. The CEXPRINT DD is automatically allocated and does not have to be added to the JCL. It defines the output file containing all IMS Connect Extensions messages. It is usually defined as SYSOUT=*. The DCB attributes for this data set are RECFM=VBA and LRECL=133. ( 4 ).
2. Submit the IMS Connect startup job.

3. Look in CEXPRINT (ddname defined in the IMS Connect startup job) for this message:

```
CEX5407I IMS Connect Extensions initialization complete
```

   This message confirms that the IMS Connect Extensions server components have successfully started. Do not proceed until you have started IMS Connect and seen this message.

   If you do not see this message:

   - Review the messages in the JES message log and in CEXPRINT.
   - Review the customization steps starting at “Creating an initial configuration for IMS Connect Extensions” on page 31.

## Verify the installation of IMS Connect and IMS Connect Extensions

This topic explains several ways to verify your installation.

At this stage, you should have at least one IMS Connect system with IMS Connect Extensions running. To verify the installation, you can view this system and its activity through the IMS Connect Extensions ISPF dialog. You can then run a series of installation verification programs and REXX commands. The IVP jobs execute IMS Connect client sessions that are relevant to your workload (OTMA or Open Database) and to the exits that are defined, and report on the output of those sessions.

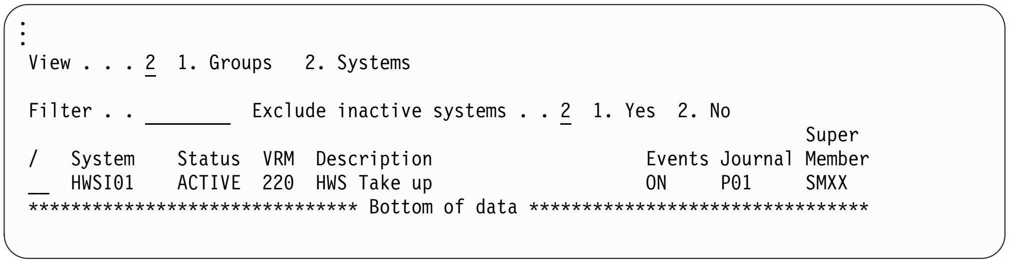
### Viewing the IMS Connect system in IMS Connect Extensions

From the IMS Connect Extensions ISPF dialog you can monitor and control all running IMS Connect systems in your definitions repository.

#### Procedure

Verify that the IMS Connect system you started earlier is running by following these steps:

1. From the primary menu of the IMS Connect Extensions ISPF dialog select option 2 **Operations**.
2. In the View option on the Operations dialog, select **2 Systems**.



```
⋮
View . . . 2  1. Groups   2. Systems

Filter . . ________  Exclude inactive systems . . 2  1. Yes  2. No
                                                                         Super
/   System    Status  VRM  Description                    Events Journal Member
__  HWSI01    ACTIVE  220  HWS Take up                    ON     P01     SMXX
******************************* Bottom of data ********************************
```

*Figure 14. Select the systems view in the Operations dialog*

3. Confirm that the status for your IMS Connect system is ACTIVE. If your system is not active, see “Connection failure between a client and a console listener” on page 367.

### Results

From the Operations dialog you can perform most IMS Connect Extensions functions:

- Enter **/** next to the system to view a list of available actions. Actions include viewing message logs or starting traces.
- Enter **SM** or **S** next to a system to view details for its ports, datastores, ODBMs, aliases, and exits. Additional actions are available for these objects. For example you can stop and start them. You can enter **SM** next to a group to view components for all systems in that group.
- Enter **AS** next to a system to view active TCP/IP sessions on that system. You can view details of long running sessions and optionally stop them. You can also enter **AS** next to a group and view all sessions running anywhere within that group.
- Enter **SH** next to the system to enter the command shell. From the command shell you can execute IMS Connect commands and see the results. Enter Help for a list of supported commands. For example enter /VIEWHWS to view the current status of IMS Connect systems.

## Running IVP jobs to generate sample OTMA workload

OTMA installation verification program (IVP) jobs run client sessions against IMS Connect. Their successful execution can be used to verify the configuration of IMS Connect and to explore IMS Connect Extensions features using sample OTMA workloads.

### Before you begin

In the IMS Connect configuration member, ensure the following:

- Either security validation is off or you have a valid RACF user name and password.
- Any user message exit routine that you use (HWSSMPL0 or HWSSMPL1) is defined in the exit list member. Each of the IMS Connect Extensions CEXIVP samples uses one of these exits.
- A valid datastore definition exists that points to an IMS system supporting any transactions generated by the IVP programs. For example, the CEXIVP programs follow the IMS IVP in generating PART transactions.

### About this task

Ideally you will have test clients which you can run to generate workloads that are representative of your transaction types and IMS environment. If not, you can use sample clients that are provided in the SCEXSAMP library.

The supplied CEXIVP test cases send an OTMA transaction to IMS and receive the response from IMS Connect.

- Each IVP program uses a different programming model:
  - CEXIVP1 performs a Commit Mode 1 SYNCLEVEL None transaction.
  - CEXIVP2 performs a Commit Mode 1 SYNCLEVEL Confirm transaction.
  - CEXIVP3 performs a Commit Mode 1 SYNCLEVEL None transaction using persistent sockets.
  - CEXIVP4 performs a Commit Mode 1 SYNCLEVEL Confirm transaction using persistent sockets.

These IVP samples use the PART transaction and the IBM-supplied exit HWSSMPL1. The IVP programs CEXIVP6 to CEXIVP9 perform the same set of tests using the IBM-supplied exit HWSSMPL0.

- The RACF PASSWORD parameter is required in the following cases:
  - If IMS Connect security is enabled (RACF=Y is specified in the IMS Connect configuration file).
  - If IMS Connect Extensions security is enabled in the system definition.

  Otherwise, the PASSWORD parameter is optional.

## Procedure

1. Edit and submit the member CEXIVPT in the SCEXSAMP library. This job runs the REXX sample program CEXRXC11 to activate OTMA tracing. Its purpose is to write additional event records to the log and show that journaling is configured correctly.
2. Submit IVP jobs to generate OTMA transactions that are appropriate to your environment. If you have your own test clients you can run these, otherwise:
   - If sample exit HWSSMPL1 is loaded and active, edit the sample members CEXIVP1-4 and follow the instructions in each.
   - If sample exit HWSSMPL0 is loaded and active, edit the sample members CEXIVP6-9 and follow the instructions in each.

   After you have tailored each sample, submit it.
3. For the CEXIVP jobs, look in the program output, which is sent to the MSGOUT ddname defined in the job, for similar messages to these:

   ```
   CEX5527I Start of formatted data
            Part..........     AN960C10; Desc.......... WASHER
            Proc Code......          74; Inv Code.......       2
            Make Dept......       12-00; Plan Rev Num...
            Make Time......          63; Comm Code......      14
   *CSMOKY*
   CEX5528I End of formatted data
   CEX5523I CSMOKY received, state information; ACK=N, CONV=N, ASYNC=N
   ```

   These messages confirm that the IVP program received an expected response.

   MSGOUT contains general information about the execution of the test program. If you do not see the preceding output in MSGOUT, look in the data set referred to by the CEXPRINT ddname. CEXPRINT contains error messages and diagnostic information to help you correct problems with the execution of the IVP.

   If CEXPRINT does not contain sufficient information to diagnose the error, perform the following actions:
   - Review the JES message log.
   - Confirm that the following is true about your environment:
     - The IMS Connect system has loaded the applicable exit (HWSSMPL0 or HWSSMPL1).
     - Communication with the datastores is established.
4. Edit and submit the member CEXIVPS in the SCEXSAMP library. This job runs the REXX sample program CEXRXC01 to switch the active journal. Its purpose is to show that journal archiving is configured correctly.

## What to do next

1. An additional IVP program, CEXIVP5, validates the installation of the IMS Connect Extensions client services exit CEXSVC01. It is not part of the general

installation verification. To use CEXIVP5 you must first install the client services exit. See Chapter 13, “Providing IMS Connect clients with additional services,” on page 137 for details.
2. To create your own IVP programs, for example to test systems with security activated, or to use IP addresses instead of host names, see Chapter 25, “Installation verification programs (IVP),” on page 339.
3. Browse the journals and generate reports using IMS PI and IMS PA if they are installed, or else run the Print utility.

**Related reference**:

“TRACE” on page 264
Use the **TRACE** host command to start or stop tracing. You can specify filter conditions to control which additional event records are written to the Active Journal data set.

“SWITCH” on page 264
The **SWITCH** command switches the active journal for an IMS Connect system.

## Running IVP jobs to generate sample Open Database workload

ODBM installation verification program (IVP) jobs run Open Database client sessions against IMS Connect. Their successful execution can be used to verify the configuration of IMS Connect and to explore IMS Connect Extensions features using sample Open Database workloads.

### Before you begin

In the IMS Connect configuration member, ensure the following:

- Either security validation is off or you have a valid RACF user name and password.
- CEXROUT0 is configured as the ODBMROUT exit member.

### About this task

Ideally you will have test clients which you can run to generate workloads that are representative of your request types and IMS environment. If not, you can use sample clients that are provided in the SCEXSAMP library.

The supplied CEXIVP test cases send DRDA requests to an IMS sample database and receive the response from IMS Connect.

- CEXIVPO1 uses the RETRIEVE function to retrieve the contents of the phone book segment 'A11111111' from the IMS IVP Phone Book database (DFSIVP1).
- CEXIVPO2 sets up an SSA and uses a PCB to retrieve all 'DEALER ' segments from the IMS IVP Dealership database (AUTPSB11).

The purpose of these jobs is to verify that IMS Connect Extensions has been installed successfully and the communication path through IMS Connect and ODBM is available.

The RACF PASSWORD parameter is required in the following cases:

- If IMS Connect security is enabled (RACF=Y is specified in the IMS Connect configuration file).
- If IMS Connect Extensions security is enabled in the system definition.

Otherwise, the PASSWORD parameter is optional.

## Procedure

1. Submit IVP jobs to generate ODBM transactions that are appropriate to your environment.

   If you have your own test clients you can run these. Otherwise, edit the members CEXIVPO1 and CEXIVPO2 in the SCEXSAMP library. Follow the instructions that are contained in each member. After you have tailored each sample, submit it.

2. For the CEXIVP jobs, look in the program output, which is sent to the SYSTSPRT ddname defined in the job. For example, the output from CEXIVPO1 should contain a series of `#Reply` blocks that echo the contents of the phone book segment 'A11111111', like this:

   ```
   000001 # ----------------------------------------------------------------------
   000002 # IVP             : CEXIVPO1
   000003 #
   000004 # Function        : DRDA sequence to retrieve the contents of the
   000005 #                   phone book segment 'A11111111'.
   000006 # ----------------------------------------------------------------------
   000007
   000008 #Socket being initialised
   000009
   000010   Client : 172.17.69.25
   000011   Server : FTS1
   000012   Family : IPV4
   000013   Port   : 48855
   000014   Socket : 1
   000015
   000016 #Request (Write Socket)
   000017
   000018  'Header'     FDSSLen: +54   FDDMID: D0   FormatID: 41   FRQSCRR: 0001
   000019  'Objects'
   000020

   ⋮
   000180  'Header'     FDSSLen: +104  FDDMID: D0   FormatID: 03   FRQSCRR: 0001
   000181  'Objects'
   000182     Length: +98   Codepoint: 241B  QRYDTA-Query Answer Set Data
   000183       +0000  '00000000 28000000 00000000 00000000'  |                |
   000184       +0010  '000000C9 E5D7C4C2 F14040F0 F14040C1'  |   IVPDB1  01  A|
   000185       +0020  'F1F1F1F1 F1F1F100 0000000A D3C1E2E3'  |1111111     LAST|
   000186       +0030  'F1404040 4040D3C1 E2E3F140 40404040'  |1     LAST1     |
   000187       +0040  'C6C9D9E2 E3F14040 4040F860 F1F1F160'  |FIRST1    8-111-|
   000188       +0050  'F1F1F1F1 C4F0F161 D9F0F140 4040    '  |1111D01/R01   |
   000189

   ⋮
   ```

   This output confirms that the IVP program received an expected response.

3. Edit and submit the member CEXIVPS in the SCEXSAMP library. This job runs the REXX sample program CEXRXC01 to switch the active journal. Its purpose is to show that journal archiving is configured correctly.

## What to do next

1. To create your own IVP programs, for example to test systems with security activated, or to use IP addresses instead of host names, see “IVP reference: Open Database workloads” on page 342.
2. Browse the journals and generate reports using IMS PI and IMS PA if they are installed, or else run the Print utility.

**Related tasks**:

“Configuring the ODBMROUT exit member” on page 30
If you want to use IMS Connect Extensions to route DRDA request traffic, specify

CEXROUT0 as the ODBMROUT exit member in the BPE configuration member.

**Related reference**:

"SWITCH" on page 264
The **SWITCH** command switches the active journal for an IMS Connect system.

# Advanced ISPF dialog configuration

The topics in this section explain how to perform advanced customization of the ISPF dialog.

## CEXOREXX

The IMS Connect Extensions initialization module CEXOREXX accepts four parameters.

**prefix** The data set prefix for IMS Connect Extensions data sets. For example, `CEX.V2R4M0`. Alternatively, specify NODYNAM to tell IMS Connect Extensions to use the existing allocation settings.

**lang** Identifies the national language. The default is ENU (U.S. English). Currently, IMS Connect Extensions only supports U.S. English.

**low level qualifiers**
Optional. Overrides the default low level qualifiers for the six IMS Connect Extensions data sets. All six qualifiers must be specified in the correct order, enclosed in brackets and separated by commas. For example: (EXEC,LINKLIB,MSG,PNL,TBL,SKL)

## Configuring libraries to be invoked dynamically

This topic explains how to enable the IMS Connect Extensions libraries to be dynamically set up when the IMS Connect Extensions dialog is invoked. Note that dynamic setup requires that the supplied library names are retained.

### Procedure

1. On the TSO command processor panel, enter the following command:

   ```
   EX 'prefix.SCEXEXEC(CEXOREXX)' 'prefix lang'
   ```

   For example:

   ```
   EX 'CEX.V2R4M0.SCEXEXEC(CEXOREXX)' 'CEX.V2R4M0 ENU'
   ```

   If the qualifier for your IMS Connect Extensions installation data sets is not CEX.V2R4M0, then alter the command accordingly.
2. To add IMS Connect Extensions to an ISPF menu, set &ZSEL to the following statement:

   ```
   CMD(EX 'prefix.SCEXEXEC(CEXOREXX)' 'prefix lang') NOCHECK
   ```

   NOCHECK is specified to support entry of concatenated commands via the direct option (trail). Also specify on the calling panel:

   ```
   &ZTRAIL=.TRAIL
   ```

## Configuring libraries to be invoked statically

This topic explains how to install the IMS Connect Extensions libraries statically within your ISPF library setup.

### Procedure

1. Include the library *prefix*.SCEXEXEC in your SYSEXEC or SYSPROC concatenation. This library contains the required EXECs. It is allocated with fixed-block 80 record format during installation.

   Put these libraries in the SYSEXEC concatenation. However, if you want to put them in SYSPROC, it must have a record length of 80 bytes.

   Ensure that all libraries contained in your concatenations are either in the same format (F, FB, V, VB) and have the same block size, or are in order of decreasing block sizes. Otherwise, you may experience problems using the IMS Connect Extensions panels.
2. Add the remaining libraries to your ISPF library setup:
   - Include the panel library *prefix*.SCEXPENU in the ISPPLIB concatenation.
   - Include the skeleton library *prefix*.SCEXSENU in the ISPSLIB concatenation.
   - Include the link/load module library *prefix*.SCEXLINK in the ISPLLIB concatenation.
   - Include the table library *prefix*.SCEXTENU in the ISPTLIB concatenation.
   - Include the message library *prefix*.SCEXMENU in the ISPMLIB concatenation.
3. On the TSO command processor panel, enter the following command:
   ```
   %CEXOREXX NODYNAM lang
   ```
4. To add IMS Connect Extensions to an ISPF menu, set &ZSEL to:
   ```
   CMD(%CEXOREXX NODYNAM lang) NOCHECK
   ```

## Overriding the data set low level qualifiers

You can override the default IMS Connect Extensions data set low level qualifiers.

The default IMS Connect Extensions data set low level qualifiers are listed under "Product libraries" on page 28. You can override these by specifying the desired qualifiers as the last parameter in the ISPF menu &ZSEL setting. All six qualifiers must be specified in the correct order, enclosed in brackets and separated by commas. For example:

```
CMD(EX ''prefix.SCEXEXEC(CEXOREXX)'' ''prefix lang (EXEC,LNK,MSG,PNL,TBL,SKL)'')
```

IMS Connect Extensions will then use the following libraries:

**prefix.EXEC**
REXX EXECs

**prefix.LNK**
Executable load modules

**prefix.MSG**
ISPF messages

**prefix.PNL**
ISPF panels

**prefix.TBL**
ISPF input tables

**prefix.SKL**
ISPF skeletons

# Upgrading to IMS Connect Extensions V2.4

Some migration actions might be required if you are upgrading from an earlier version of IMS Connect Extensions.

### Upgrading from IMS Connect Extensions V2.1, V2.2, or V2.3

- APAR PM15459 introduced new data fields in the Active Sessions form. Users who have customized the Active Sessions dialog must recreate their customized layout to allow for these new fields. See “Customizing the Active Sessions panel” on page 117.
- IMS Connect Extensions V2.1 introduced the ability to issue IMS Connect commands from IMS Connect Extensions, as well as issue the existing IMS Connect Extensions commands. For security reasons, all commands are disabled when you upgrade. To re-enable commands, edit the system definition in IMS Connect Extensions and select **Activate commands**; you must then restart the IMS Connect system.
- The Operations Console cannot monitor a system running IMS Connect Extensions V2.1 or earlier.

# Install IMS Connect Extensions Operations Console

These topics describe how to install and start the IMS Connect Extensions Operations Console. You can use the Operations Console to monitor and control IMS Connect systems and their components and active sessions from a distributed workstation.

IMS Connect Extensions Operations Console is Built on Eclipse. It is a PC application that provides a graphical interface to perform IMS Connect Extensions operations. The Operations Console is installed as a plug-in to IBM Explorer for z/OS (z/OS Explorer).

**Tip:** z/OS Explorer can be installed on a network drive so that it can be run by multiple users from a network folder, but the configuration area and workspaces are stored on users' local workstations. See the Eclipse website for documentation that describes the steps that are required to set up a multiuser installation.

**Related reference**:

“Hardware and software prerequisites” on page 26
This section lists the requirements for installing and running IMS Connect Extensions and IMS Connect Extensions Operations Console.

## Obtaining the required packages

The Operations Console plug-in can be found in the IMS Connect Extensions SCEXGENU library. The SMP/E installation data set includes a member named CEXEPR, which is a zip file that adds the IMS Connect Extensions functionality to the z/OS Explorer shell.

### About this task

To install the Operations Console, you will need to transfer the following members from the SMP/E installation data set to a PC workstation:

**CEXEPR**
A zip file that provides the IMS Connect Extensions functionality for the environment.

**CEXREAD**

A text file that provides the fix history, prerequisites, and installation instructions for the Operations Console.

**Note:** Ensure that you install the Connection Server client components from the IMS Connect Extensions SCEXGENU library. The SCEXGUIW library, which was allocated and used by previous releases of this product, is no longer used in this release and can be deleted after you delete the previous release from your system.

### Procedure

1. Using FTP in binary mode, transfer the CEXEPR member from the SMP/E installation data set to your workstation. Rename CEXEPR to cexepr.zip.
2. Using FTP in ASCII mode, transfer the CEXREAD member from the SMP/E installation data set to your workstation. Rename CEXREAD to cexread.txt.

## Installing the Operations Console

Follow the instructions in cexread.txt to install the Operations Console. First install z/OS Explorer, if it is not already installed, and then install the required components for IMS Connect Extensions (cexepr.zip).

## Starting the Operations Console

This topic explains how to run the IMS Connect Extensions Operations Console on a distributed workstation.

### About this task

**Workspace:** The workspace folder is used to save details of your customizations to the workbench layout and working sets. You can specify a location anywhere on a local or network drive. The default location will depend on your Windows version. The default folder name is `workspace`.

**Tip:** It can be very useful to set up multiple workspaces. Each workspace can represent a different configuration, for example PRODCONFIG1, PRODCONFIG2, and TEST, that can be chosen when the Operations Console is started.

**Default Connection Profile:** The Default Connection Profile specifies a set of TSO credentials that is used to connect to any server that does not specify a connection profile explicitly.

**Configuration file:** A configuration file is a text file in CSV (comma-separated values) format. Each entry contains information about an IMS Connect system or group. Operations Console users can use configuration files to import a common set of system and group definitions instead of having to define them individually. Configuration files can either be exported from another instance of the Operations Console or created by the batch definition extract utility for the IMS Connect system you wish to define.

### Procedure

To start the IMS Connect Extensions Operations Console interface:

1. Double-click the z/OS Explorer shortcut. You are prompted to select a workspace folder.
2. If you are not already viewing the IMS Connect Extensions perspective in z/OS Explorer, switch to it now:

a. In the Welcome page, click the **Workbench** icon.
b. Select **Window** > **Open Perspective** > **Other** > **IMS Connect Extensions**.

## Results

When you first open the IMS Connect Extensions perspective it loads with a blank navigation view on the left panel, using the settings in your current workspace.

If this is a new workspace, you are prompted to specify a TSO user ID and password for the Default Connection Profile.

An item labeled All Sources is listed in the Navigation view. This is a container for all your IMS Connect systems and groups, as well as data sources contributed by other plug-ins.

Any systems that you define will be displayed here, each with a status icon next to it.

- A green circle indicates that the console could successfully connect to the system.
- A red square indicates that the connection attempt was unsuccessful.

  To diagnose an unsuccessful connection attempt, refer to “Connection failure between a client and a console listener” on page 367.

## What to do next

You can perform the following tasks to configure Operations Console objects and preferences:

- Add other connection profiles:

  Click the **View** menu ( ▽ ) in the Navigation toolbar and then click **Preferences > Connection Profiles**.
- Import a configuration file containing system and group definitions:

  Click the **View** menu ( ▽ ) in the Navigation toolbar and click **Import IMS Connect Extensions Configuration**.
- Add systems individually:

  Click ☐ in the Navigation view and then click **IMS Connect**.

  For a description of the fields, press F1 then click the **New system** topic. If you are unsure of what values to enter, contact the administrator who is responsible for IMS Connect Extensions at your site.
- Add groups individually:

  Click ☐ in the Navigation view and then click **Group**.
- Add systems to a group:

  Right-click a group in the Navigation view and then click **Edit**.

Procedures for using the Operations Console are explained in the online help. To access the help system, click **Help** > **Help Contents**. To resolve problems connecting to a system, refer to the help topic **Troubleshooting connection failures**.

# Updating the Operations Console plug-in

To install an updated version of the IMS Connect Extensions Operations Console plug-in, download and replace the .zip file containing the original plug-in.

## Before you begin

If the previous version of the Operations Console plug-in ran under z/OS Explorer or the Tools Base Connection Server client framework V1.4.0.04 or later, you can back up your old workspace files and restore them to your new z/OS Explorer workspace if necessary. Copy the contents of:

```
old workspace folder\.metadata\.plugins\com.fundi.framework.eclipse
```

to:

```
new workspace folder\.metadata\.plugins\com.fundi.framework.eclipse
```

## Procedure

To perform an update:

1. Start z/OS Explorer.
2. Click **Help** > **Check for updates**. If you have copied the new files to the same location that you previously used to install the plug-in, then Eclipse will identify the components that can be updated.

   **Note:** If Eclipse does not identify that an update is available, follow the steps in "If Eclipse does not find the updated plug-in".
3. The plug-in appears in the list of available software. Select the checkbox next to the plug-in, and then click **Next**.
4. Follow the on-screen instructions to complete the installation. If you receive a security warning about unsigned content, click **OK**.

**If Eclipse does not find the updated plug-in:**

Eclipse has an issue that can cause it to overlook an updated .zip file. Perform the following steps to identify to Eclipse that an updated plug-in is available:

5. Click **Help > Install New Software**.
6. Click the link labeled Available Software Sites.
7. In the Available Software Sites window, select and highlight the CEX plug-in, and then click **Reload**. Eclipse displays a message saying that the plug-in is available.
8. Click **OK** to dismiss the message.
9. In the Available Software Sites window, click Reload to reload the CEX plug-in a second time. Eclipse redisplays a message saying that the plug-in is available.
10. Click **OK** to dismiss the message.
11. Click **Cancel** to exit the Available Software Sites window. You can now return to the main procedure above.

# Chapter 3. ISPF dialog and Operations Console

IMS Connect Extensions provides centralized management and control of all your IMS Connect systems, from both the ISPF operations dialog and from the Eclipse-based Operations Console.

You can perform tasks such as:

- Monitoring IMS Connect throughput
- Analyzing and controlling client sessions
- Issuing IMS Connect WTOR commands and IMS Connect z/OS commands, IMS Type-1 commands, and IMS Connect Extensions commands
- Starting and stopping traces
- Viewing logs

## The ISPF dialog

The IMS Connect Extensions dialog has been designed to follow CUA and established ISPF conventions.

- Possible actions are presented in action bar pull-down menus; those available from the File and Edit pull-down menus can also be requested from the command line.
- A menu or selection list item can be selected either by positioning the cursor over it (point-and-shoot) or by specifying its corresponding number, and then pressing Enter.
- For many entry fields you can select from a list of available choices by positioning the cursor on the field and pressing the Prompt function key (F4). A + (plus sign) to the right of the field or column heading indicates that Prompt is available.
- Shortcut navigation to the primary IMS Connect Extensions functions is available. For example, to invoke Definitions, you can select option 1 from the IMS Connect Extensions primary menu, or enter =1 on the command line from anywhere in the IMS Connect Extensions dialog.
- Context-sensitive help is available for each panel and input field.

## ISPF environment considerations

The IMS Connect Extensions dialog is an ISPF application adhering to CUA conventions.

### Function key settings

IMS Connect Extensions uses standard conventions for function keys (for example, F1=Help, F3=Exit, F4=Prompt, F7=Backward, F8=Forward, F10=Left, F11=Right, F12=Cancel), and displays the settings at the bottom of each panel. However, ISPF facilities accessed using the **KEYS** and **KEYLIST** commands enable users to assign alternative functions to the keys. Also, the ISPF command **PFSHOW ON/OFF** allows display of key settings to be turned on and off. The IMS Connect Extensions default settings for the function keys can be displayed using the **KEYSHELP** command, also available from **Help** in the action bar.

### Prompt (F4)

Prompt is available on various data entry fields throughout the IMS Connect Extensions dialog to help you specify valid values. Eligible prompt input fields are suffixed with a plus sign (+).

To use this facility, position the cursor in the input field and press the Prompt function key (F4). A list of available values is displayed from which you can select one or more as instructed.

### Point-and-Shoot fields

IMS Connect Extensions employs point-and-shoot fields. To ensure point-and-shoot works in IMS Connect Extensions, use the `SETTINGS` command and select **Tab to point-and-shoot fields**.

### Panel size and scrolling

IMS Connect Extensions dialog panels are optimized for 32 lines, but accommodate 24 lines using scrolling with the Backward function key (F7) and the Forward function key (F8). Help windows have a maximum of 24 lines.

### Displaying messages

IMS Connect Extensions uses both long and short messages. Short messages display at the top right, on the same line as the panel title. Long messages are designed to display in a pop-up window. However, long messages of less than the panel width can be customized to display just after or before the command line rather than in a window if **Long message in pop-up** is not selected on the ISPF Settings panel accessed using the `SETTINGS` command.

Messages displayed in a window can be moved to another location on the panel:

1. Position the cursor on the top or bottom border of the message window, and press Enter.
2. Position the cursor at the location on the panel to which you wish to move the message, then press Enter.

## Online Help

IMS Connect Extensions help is context-sensitive. That is, the information displayed is appropriate to the position of the cursor when you request Help.

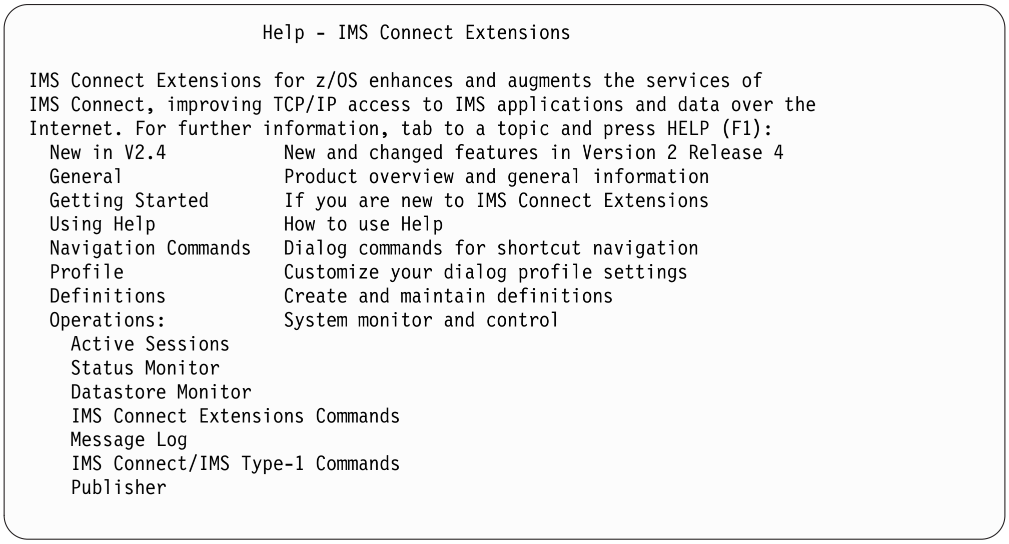
The default key to request Help is the Help function key (F1).

The IMS Connect Extensions dialog offers the following types of online Help:

- Help options in the action bar of every panel:
  1. Extended Help
  2. Commands Help
  3. Keys Help
- Extended Help is available from the command line of every panel. Position the cursor on the command line and press the Help function key (F1) or enter the `HELP` command. The line actions and primary commands that apply to the current panel are listed in the Extended Help.
- Field Help is available on every input field. Position the cursor in the field and press the Help function key (F1).

- Message Help: When a short message is displayed at the top right, press the Help function key (F1) to see the long message in a pop-up window. This is applicable if, in ISPF Settings, you have selected **Long message in pop-up**.

Example: The following Help window is displayed when you press the Help function key (F1) or enter **HELP** in the command line of the Primary Option Menu, or select **Extended Help** from **Help** in the action bar.



```
                      Help - IMS Connect Extensions

IMS Connect Extensions for z/OS enhances and augments the services of
IMS Connect, improving TCP/IP access to IMS applications and data over the
Internet. For further information, tab to a topic and press HELP (F1):
  New in V2.4          New and changed features in Version 2 Release 4
  General              Product overview and general information
  Getting Started      If you are new to IMS Connect Extensions
  Using Help           How to use Help
  Navigation Commands  Dialog commands for shortcut navigation
  Profile              Customize your dialog profile settings
  Definitions          Create and maintain definitions
  Operations:          System monitor and control
    Active Sessions
    Status Monitor
    Datastore Monitor
    IMS Connect Extensions Commands
    Message Log
    IMS Connect/IMS Type-1 Commands
    Publisher
```

*Figure 15. Example: Online Help*

To use the online Help efficiently, the Color and Intensity attributes for Normal Text, Emphasized Text and Reference Phrase should have different settings. Enter the ISPF command **CUAATTR** to check these.

The default settings are appropriate:

- Normal Text: Green, Low intensity
- Emphasized Text: Turquoise, High intensity
- Reference Phrase: White, High intensity

To navigate inside Help windows:

- Reference Phrases are used to indicate that more information is available on a topic. Press the Tab key to position the cursor on a Reference Phrase (typically white, highlighted) then press the Help function key (F1). A pop-up Help window displays additional information on the topic.
- When **More: +** is displayed in the top right corner, press the Forward function key (F8) or Enter for next page.
- When **More: -** is displayed in the top right corner, press the Backward function key (F7) for previous page.
- To close the Help window and return to the underlying window, press the Cancel function key (F12).
- To close Help and return to the dialog panel, press the Exit function key (F3).

To resize the Help window by removing or restoring the border, press the Resize function key (F6).

To display the default function keys for Help windows as described here, press the KeysHelp function key (F4).

# Operations Console

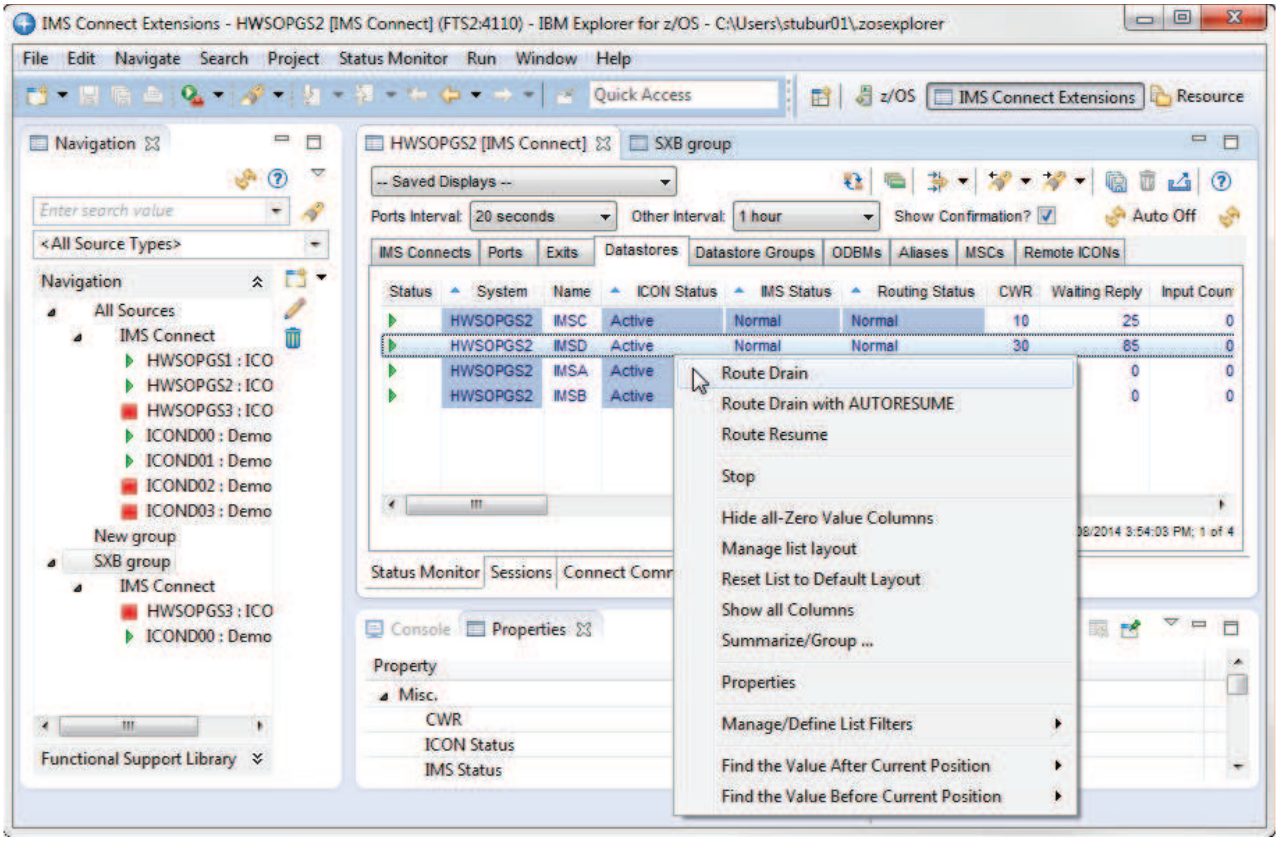
IMS Connect Extensions Operations Console is an Eclipse-based PC application that provides a graphical interface to perform IMS Connect Extensions operations from a distributed workstation.

## Eclipse environment considerations

IMS Connect Extensions Operations Console is installed as a plug-in to z/OS Explorer and adheres to Eclipse conventions.

Figure 16 shows an example of the console layout:

- To the left: the Navigation view, which can be used to add or select data sources such as systems and groups
- In the middle: a tabbed area for working in one or more editors
- At the bottom: a Console view for displaying messages



IMS Connect Extensions - HWSOPGS2 [IMS Connect] (FTS2:4110) - IBM Explorer for z/OS - C:\Users\stubur01\.zosexplorer

-- Saved Displays --

Ports Interval: 20 seconds  Other Interval: 1 hour  Show Confirmation? ☑  Auto Off

| Status | System | Name | ICON Status | IMS Status | Routing Status | CWR | Waiting Reply | Input Coun |
|---|---|---|---|---|---|---|---|---|
| ▶ | HWSOPGS2 | IMSC | Active | Normal | Normal | 10 | 25 | 0 |
| ▶ | HWSOPGS2 | IMSD | Active | Normal | Normal | 30 | 85 | 0 |
| ▶ | HWSOPGS2 | IMSA | Active | | | | 0 | 0 |
| ▶ | HWSOPGS2 | IMSB | Active | | | | 0 | 0 |

*Figure 16. Operations Console sample screen*

The interface is highly configurable. For example, you can filter and highlight list data. The Manage List Layout window allows a particular arrangement of tabular data in each view to be saved and reused. You can open a detailed view of the attributes of a selected item by right-clicking on it and selecting **Properties**.

Panels within the perspective can be moved by dragging them to a new location. Panels can be resized by dragging the divider, which is the border between it and the next panel.

Configuration settings that are changed during a session are saved to the user's specified workspace folder. To revert the current perspective to the default version, select **Window** > **Reset perspective**.

### Context menus

Right-clicking almost anywhere in the console displays a context menu showing the available options for the current panel or selected item.

### Displaying messages

IMS Connect Extensions messages display in the Console view.

## Online Help

Online Help is available throughout the Operations Console. Pressing the Help function key (F1) or clicking the **Help** button displays a Help panel to the right of the console.

The Help panel contains context-sensitive information and links appropriate to the current view or editor. It also includes several options to help you find a particular topic:

**Contents**
View the Table of Contents.

**Search**
List topics containing a search string or expression.

**Related Topics**
List topics relating to a selected area of the Operations Console.

**Bookmarks**
List favorite topics that you have previously bookmarked.

**Index** List index entries. Each entry is a link to the related topic.

To display the full help system in a separate window, select **Help Contents** from the **Help** menu.

To display keyboard shortcuts, press **Ctrl-Shift-L** or select **Key Assist** from the **Help** menu.

# Part 2. Collecting and reporting IMS Connect events

These topics introduce event collection, explain how to configure and manage journal data sets, and describe how to work with other IMS Tools to analyze and report on IMS Connect activity.

**Topics:**

# Chapter 4. Event collection

IMS Connect Extensions continuously collects events as incoming messages or DRDA requests are processed or when other changes occur, such as when IMS Connect starts up and shuts down or when a datastore goes offline.

IMS Connect publishes information regarding these events using the HWSTECL0 exit. IMS Connect Extensions captures the event details from this exit and writes event records to the active journal. IMS Connect Extensions can augment these event records with additional data and can also produce additional events (tracing events are one example).

The Event **Collection level** specified in the system definition (value 0 to 4) determines which events are actually recorded in the journal. The types of reports you want to view will determine which collection level to specify.

When tracing is active, additional tracing records are written by IMS Connect Extensions to the active journal. For message-related workloads, the tracing level determines whether the tracing records also include client application data. A unique log record prefix in the range X'A0' - X'FF' is used to identify IMS Connect event records created by IMS Connect Extensions.

**Related concepts**:

"Event collection architecture" on page 16
To create a comprehensive picture of activity within IMS Connect, IMS Connect Extensions collects information directly from IMS Connect and through the exit manager.

Chapter 6, "Tracing message flows," on page 85
Tracing events are additional event records that are produced by IMS Connect Extensions when tracing is activated. You can query and set the status of tracing through the ISPF Operations dialog.

**Related reference**:

"SET TRACING" on page 239
The SET TRACING command causes IMS Connect Extensions to trace based on filtering criteria.

## Active journals and archive journals

Event records are first written to active journal data sets and subsequently copied by the Archive Manager task to archive journal data sets.

The active journal consists of a series of up to 32 BSAM data sets that are used in rotation. When the first data set is full a second data set is used. This continues until the last data set is full, when IMS Connect reverts to using the first data set.

Archive journal data sets are dynamically allocated by the Archive Manager using an archive JCL skeleton. You can create the skeleton using one of the sample skeletons that is provided in the SCEXSAMP library.

The Archive Manager runs whenever an active journal data set is ready to be archived. This happens when:

- The active journal data set is full and IMS Connect Extensions has switched to the next data set in the rotation.
- IMS Connect Extensions is shut down and any remaining event records in active journal data sets need copying.
- IMS Connect Extensions restarts and for some reason there are outstanding event records to be archived.
- The journal is switched prior to becoming full. You can force a switch manually by:
  - Using the `JOURNAL SWITCH` dialog command
  - Using the `SWITCH` host command in a REXX exec
  - Using the `Switch Journal` action in the Operations Console

You can also perform archiving manually by submitting a batch job.

Note that if all of the active data sets are full, event records might be discarded or overwritten depending on the setting of the Journal full option.

It's possible to use only active journal data sets. If archiving is not required, clear the **Archive manager** field and specify a larger number of data sets in the Active Journal Data Set definition.

## Types of event record

The number and type of event records collected varies depending on the collection level specified for the IMS Connect system and the tracing level.

**Connect status event**

A Connect status event identifies a change in the status of your IMS Connect system. For example, a resource becoming available or unavailable, or a socket becoming accepted for input by a port task, or a journal switch. Connect status events are typically not related to the processing of input messages, but can affect their processing.

Connect status event records are identified by a constant Event Key, EVNT.

**Workload-related event**

These records cover message-related events, Open Database-related events, and IMS-to-IMS TCP/IP communications. IMS Connect Extensions uses a STCK token Event Key to associate workload-related event records with each other. This allows event records to be identified and reported in the sequence they occur.

Message-related event records: For non-persistent sockets, each incoming message is assigned a unique Event Key and every event associated with the processing of the message has the same Event Key. For persistent sockets, all messages and all their associated events for the duration of the socket are assigned the same Event Key.

Open Database requests and responses and associated calls to ODBM: All events for the duration of the socket are assigned the same Event Key.

IMS-to-IMS TCP/IP communications: All events for the duration of the link are assigned the same Event Key.

**RECORDER trace event**

Recorder trace data produced by IMS Connect is converted to event record format by the IMS Connect Extensions Recorder trace utilities.

**IMS Connect trace event**

IMS Connect Extensions trace records. These records include IRM, CSM, RSM, OTMA, XML, and DRDA structures. Trace records can optionally include application data. Which records are written can be made conditional on their matching specified criteria.

Event record 172 (X'AC') is a special type of event record that is used to represent a set of events resulting from IMS Connect Extensions command being issued. A record is produced for each significant action that is initiated by this command or is issued by another process in response to or on behalf of this command.

For example, events recorded for the Drain/Resume feature include the issuing of the ROUTE DRAIN and ROUTE RESUME commands as well as actions taken on behalf of the ROUTE command such as AUTORESUME processing. If the ROUTE DRAIN or ROUTE RESTORE command targets a routing list, one “request” record and one “response” record is created for the routing list, as well as one “both” record for each datastore in the routing list.

**User data logging event**

User data event records contain data submitted by clients for logging to the journal. User data can be segmented over multiple 1024-byte records. The client must have a valid user ID and password to use this service.

IMS Connect Extensions uses *framing events* to clearly identify all events for a transaction or iteration of a conversational transaction.

**Start of frame**

The Read Prepare (X'3C') event is used as the start of frame event record. This record starts a collection of event records related by a key token.

**End of frame**

IMS Connect generates this Trigger (X'48') event at the end of a multiple event process. It may be as a result of a deallocate request or other condition that represents the end of the process. The trigger event contains an indication of why the trigger event was generated.

**Related concepts**:

Chapter 31, “Event records recorded by IMS Connect Extensions,” on page 435
Event records are collected continuously as messages are processed by IMS Connect. An event record consists of an event number and data associated with the event.

“RECORDER trace utilities” on page 331
Two IMS Connect Extensions utilities can be used to process an IMS Connect-managed Recorder Trace data set (HWSRCDR). This data set stores information about the messages that are processed by IMS Connect.

# Sample event flows

## Sample OTMA event flow: Sync Level None

The typical event flow of an incoming message with Sync Level NONE is:

```
3C Prepare Read Socket              <== Incoming message from client
49 Read Socket
3D Message Exit called for READ
3E Message Exit return for READ
41 Message sent to OTMA             <== Sent to OTMA for processing
42 Message received from OTMA
3D Message Exit called for XMIT
3E Message Exit return for XMIT
4A Write Socket                     <== Response sent back to client
0C Begin Close Socket               <== Non-persistent Socket is closed
0D End Close Socket
48 Trigger event CLOS               <== Connect has finished processing message
```

*Figure 17. Sample event flow: Sync Level None*

## Sample OTMA event flow: Sync Level Confirm

The typical event flow of an incoming message with Sync Level CONFIRM is:

```
3C Prepare Read Socket              <== Incoming message from client
49 Read Socket
3D Message Exit called for READ
3E Message Exit return for READ
41 Message sent to OTMA             <== Sent to OTMA for processing
42 Message received from OTMA
3D Message Exit called for XMIT
3E Message Exit return for XMIT
4A Write Socket                     <== Response sent back to client
49 Read Socket                      <== ACK received from Client
3D Message Exit called for READ
3E Message Exit return for READ
41 Message sent to OTMA             <== ACK sent to OTMA
42 Message received from OTMA
46 De-allocate Session
3D Message Exit called for XMIT
3E Message Exit return for XMIT
4A Write Socket                     <== De-alloc Response sent back to client
0C Begin Close Socket               <== Non-persistent Socket is closed
0D End Close Socket
48 Trigger event CLOS               <== Connect has finished processing message
```

*Figure 18. Sample event flow: Sync Level Confirm*

## Sample ODBM event flow

```
3C Prepare READ Socket                               <== Incoming message from client
49 READ Socket
5B DRDA 1041 EXCSAT-Exchange Server Attributes       <== Request for server attributes
49 READ Socket                                           (including security mechanism)
49 READ Socket
5B DRDA 106D ACCSEC-Access Security
5C DRDA 1443 EXCSATRD-Server Attributes Reply Data
4A WRITE Socket
49 READ Socket
49 READ Socket
5B DRDA 106E SECCHK-Security Check                   <== Request Security check
63 ODBM Security Exit called                         <== Open Database Security Exit called
64 ODBM Security Exit returned
5C DRDA 1219 SECCHKRM-Security Check Reply Message   <== Reply from Security Check
4A WRITE Socket
49 READ Socket
49 READ Socket
5B DRDA 2001 ACCRDB-Access RDB                       <== Request to allocate PSB
5D ODBM begin Allocate PSB (APSB) Program=AUTPSB11
61 ODBM Routing Exit called                          <== Open Database Routing Exit called
62 ODBM Routing Exit returned
AA ODBM Trace: Message sent to ODBM
69 Message sent to ODBM
AA ODBM Trace: Message received from ODBM
6A Message received from ODBM
5E ODBM end Allocate PSB (APSB) Program=AUTPSB11     <== PSB allocated
5C DRDA 2201 ACCRDBRM-Access RDB Reply Message
4A WRITE Socket
48 Trigger Event for ODBMMSG
3C Prepare READ Socket
49 READ Socket                                       <== Request data
5B DRDA 200C OPNQRY-Open Query
49 READ Socket
49 READ Socket
5B DRDA CC05 DLIFUNC-DL/I function
49 READ Socket
49 READ Socket
5B DRDA CC01 INAIB-AIB data
49 READ Socket
49 READ Socket
5B DRDA CC04 RTRVFLD-Field client wants to retrieve data
49 READ Socket
49 READ Socket
5B DRDA CC04 RTRVFLD-Field client wants to retrieve data
49 READ Socket
49 READ Socket
5B DRDA CC06 SSALIST-List of segment search argument
AA ODBM Trace: Message sent to ODBM                  <== Request sent to ODBM
69 Message sent to ODBM
AA ODBM Trace: Message received from ODBM
6A Message received from ODBM                        <== Reply contains requested data
5C DRDA 2205 OPNQRYRM-Open Query Complete
4A WRITE Socket
48 Trigger Event for ODBMMSG
...
3C Prepare READ Socket
49 READ Socket
5B DRDA C801 DEALLOCDB-Deallocate PSB                <== Request to Deallocate PSB
5F ODBM begin Deallocate PSB (DPSB)
AA ODBM Trace: Message sent to ODBM
69 Message sent to ODBM
AA ODBM Trace: Message received from ODBM
6A Message received from ODBM
60 ODBM end Deallocate PSB (DPSB)
5C DRDA CA01 DEALLOCDBRM-Name of deallocated PSB     <== Reply to Deallocate PSB
4A WRITE Socket
48 Trigger Event for ODBMMSG
3C Prepare READ Socket
47 Session Error
0C Begin CLOSE Socket
0D End CLOSE Socket                                  <== Client disconnected
```

*Figure 19. Sample ODBM event flow*

## Sample MSC event flow

Example of the flow of events on a local and a remote IMS Connect system connected by a MSC TCP/IP-type link.

```
7D ICON to ICON start of session
78 MSC message received from MSC Msgtype=REQUEST     <== Notification of transaction
A3 Event Collection OTMA Trace                           coming from IMS
3D Message Exit called for XMIT
A6 Event Recording EXIT Output Message Trace
3E Message Exit returned from XMIT
79 MSC message sent to remote ICON Msgtype=REQUEST   <== Transaction passed to remote
                                                         IMS Connect via TCP/IP
```

*Figure 20. Sample MSC event flow: local IMS Connect system*

```
3C Prepare READ Socket
49 READ Socket
A4 Event Collection IRM Trace
3D Message Exit called for READ
A3 Event Collection OTMA Trace
3E Message Exit returned from READ
7D ICON to ICON start of session
7A MSC message received from remote ICON Msgtype=REQUEST
7B MSC message sent to MSC Msgtype=REQUEST              <== Transaction passed to IMS
78 MSC message received from MSC Msgtype=REQRESP        <== Response from IMS
A3 Event Collection OTMA Trace
3D Message Exit called for XMIT
A6 Event Recording EXIT Output Message Trace
3E Message Exit returned from XMIT
79 MSC message sent to remote ICON Msgtype=REQRESP      <== Response sent to local
7E ICON to ICON end of session                              IMS Connect via TCP/IP
```

*Figure 21. Sample MSC event flow: remote IMS Connect system*

```
3C Prepare READ Socket
49 READ Socket
A4 Event Collection IRM Trace
3D Message Exit called for READ
A3 Event Collection OTMA Trace
3E Message Exit returned from READ
7A MSC message received from remote ICON Msgtype=REQRESP     <== Response passed to
7B MSC message sent to MSC Msgtype=REQRESP                       local IMS
7E ICON to ICON end of session
78 MSC message received from MSC Msgtype=REQRESP
A3 Event Collection OTMA Trace
3D Message Exit called for XMIT
A6 Event Recording EXIT Output Message Trace
```

*Figure 22. Sample MSC event flow: local IMS Connect system*

## Configuring event collection and journals: overview

To configure event collection on a system you must activate event collection in the system definition and enter details of the active and archive journal data sets.

| Step | Description | Reference and instructions |
|---|---|---|
| 1 | Determine a journal archiving scheme that is appropriate for your environment and gather required information.<br>• Event collection level<br>• Number of active journals in the pool and their size<br>• A base name for active journal data sets<br>• When to take an archive copy of the active journal and how long to retain it<br>• A base name for archive journal data sets<br>**Important:** Ensure that either the user running IMS Connect or the IMS Connect task has RACF ALTER authorization for the active and archive journal data sets. | Collection level<br>"Estimating the size of active journals"<br>"Active Journal" on page 197<br>"Naming archive journal data sets" on page 62 |
| 2 | Create an archive JCL skeleton based on one of the samples. | "Creating an archive JCL skeleton" on page 63 |
| 3 | In the system definition:<br>• **Activate Event Collection** and set the collection level.<br>• Enter / (slash) next to **Active Data Set**. | "Defining IMS Connect systems" on page 189 |
| 4 | In the Active Journal Data Set panel:<br>• Specify the base name and attributes of Active Journal data sets.<br>• Select **Archive manager** to activate journal archiving.<br>• Specify the name of the archive JCL skeleton member. | "Active Journal" on page 197 |
| 5 | In the Archive Journal Data Set Template panel:<br>• Specify a pattern for naming Archive Journal data sets and other Archive Journal properties. | "Archive Journal" on page 200 |
| 6 | To implement the new event collection and journal settings, either refresh the system definition or restart the IMS Connect system.<br>IMS Connect Extensions will collect events at the specified level and automatically allocate active journal data sets. | "Changing definitions dynamically with REFRESH commands" on page 234<br>Or:<br>REFRESH host command. |

## Estimating the size of active journals

This topic provides some assistance when determining Active Journal data set sizes.

The following table shows the amount of data produced for the different collection levels when processing 10 transactions per second for 24 hours (total 864,000 transactions). These numbers are indicative only.

*Table 1. Active Journal data set size estimates (based on 3390 devices)*

| Collection level | Tracks/day | Cyls/day | MB/day |
|---|---|---|---|
| 0 | 1 | < 1 | < 1 |
| 1 | 5184 | 346 | 277 |
| 2 | 13738 | 916 | 734 |
| 3 | 22291 | 1486 | 1190 |
| 4 | 24106 | 1607 | 1287 |

The number of Active Journal data sets and their size depends on two factors:

1. How long you want to wait before accessing the data and running reports.
2. The amount of processing required by the Archive Manager to copy the files to the Archive Journal data sets.

You can define a small number of large data sets or a large number of smaller data sets.

## Naming archive journal data sets

You can use symbols representing date and time stamps and the IMS Connect system ID to generate a unique archive data set name.

You can use any of the symbols in Table 2, but the symbols you choose must generate a unique data set name.

These symbols will generate unique names:

- &FIRST
- &JFIRST
- &DATE with &TIME
- &JDATE with &TIME

*Table 2. Symbols available for the archive data set name*

| Name | Description | Result format | Example result |
|---|---|---|---|
| &DATE | The date the archive journal was created. | D*YYMMDD* | `D160115` **1** |
| &FIRST | The date and time of the earliest event in the archive journal. | D*YYMMDD*.T*HHMMSS* | `D160115.T153025` **2** |
| &ID | The HWS (system) ID of the IMS Connect system. | 1 - 8 characters | `HWS1` |
| &JDATE | The Julian date when the archive journal was created. | J*YYDDD* | `J16015` **1** |
| &JFIRST | The date and time of the earliest event in the archive journal. The date is in Julian format. | J*YYDDD*.T*HHMMSS* | `J16015.T153025` **2** |
| &TIME | The time the archive journal was created. | T*HHMMSS* | `T153025` |

**Notes:**

**1**
Example result is for an archive journal data set created on January 15, 2016 GMT.

**2**
Example result is for an archive journal data set where the earliest event is time stamped on January 15, 2016 at 15:30:25 GMT.

### Specifying symbols

- End each symbol with a period (`.`).

When you use a symbol as the last qualifier in a data set name, the period is optional.

- If a period follows a symbol, type two consecutive periods (`..`).

  For example, to obtain a data set name ending with a date and time stamp, such as `'USER.ARCHIVE.D151221.T115520'`, type the following pattern:

  ```
  'USER.ARCHIVE.&DATE..&TIME'
  ```

### Using a generation data group (GDG) for archive journal data sets

Use the suffix (+1) to indicate that a GDG base will be used for archive data sets. For example:

```
'YOUR.ARCHIVE.FILE(+1)'
```

You can use the &ID symbol in a GDG name. This symbol resolves to the IMS Connect system name. A GDG base with the IMS Connect system name must exist for the job to complete successfully.

# Defining and managing active and archive journals

These topics describe how to define active journal data sets and then archive them to a direct access storage device (DASD) or magnetic tape.

## Creating an archive JCL skeleton

Create an archive JCL skeleton with the data set allocation settings you need. The Archive Manager task uses this skeleton to generate and submit the archiving job.

When the active journal switches, its contents are written to an archive journal by the Archive Manager task. IMS Connect Extensions automatically switches the active journal when it is full. You can initiate a journal switch manually using the SWITCH host command or JOURNAL SWITCH ISPF dialog command.

The Archive Manager generates the archive job using a JCL skeleton that you create and the archive journal and active journal settings. You specify the name of a member containing the JCL skeleton in the Active Journal Data Set configuration settings. These sample skeletons are provided in the SCEXSAMP library:

- CEXARCH1: Archive to DASD.
- CEXARCH2: Archive to magnetic tape.
- CEXARCH3: Archive to tape or CART, then run a report against the archive.

The Archive Manager submits the generated job to create the archive data set.

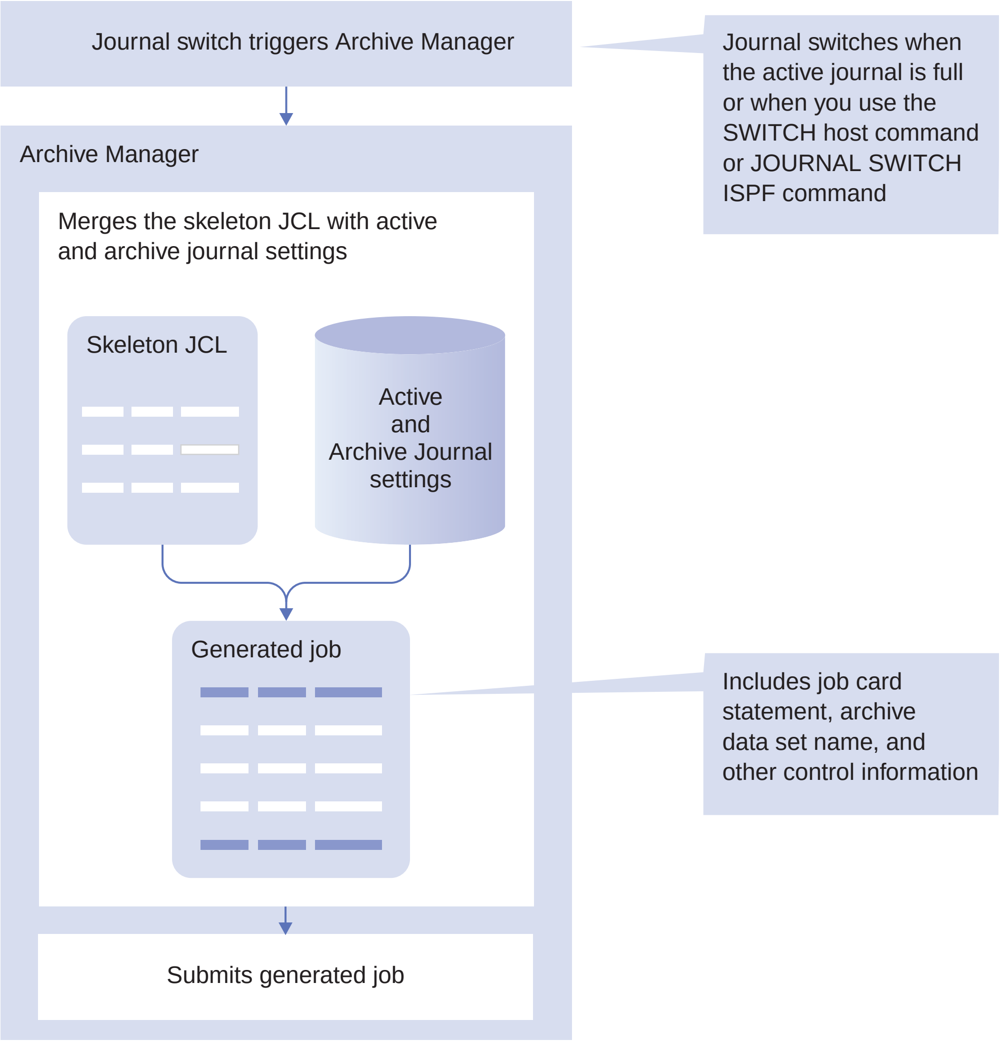

Journal switch triggers Archive Manager
Journal switches when the active journal is full or when you use the SWITCH host command or JOURNAL SWITCH ISPF command
Archive Manager
Merges the skeleton JCL with active and archive journal settings
Skeleton JCL
Active and Archive Journal settings
Generated job
Includes job card statement, archive data set name, and other control information
Submits generated job

*Figure 23. When and how the Archive Manager uses the skeleton JCL*

## Archiving to DASD

Here is a simple JCL skeleton for archiving active journals to a direct access storage device (DASD). A similar skeleton, CEXARCH1, is included in the SCEXSAMP library.

```
//STEP01    EXEC PGM=FUNEXEC,PARM=CEXJARC,REGION=0M
//STEPLIB   DD DISP=SHR,DSN=funpre.SFUNLINK
//          DD DISP=SHR,DSN=cexpre.SCEXLINK 1
//ARCOUT    DD  DSN=%ARCDSN, 2
//              DISP=(,CATLG,DELETE),
//              SPACE=(CYL,(pri,sec),RLSE), 3
//              LIKE=journal.dataset 4
```

**1** Enter the high-level qualifier of the IMS Connect Extensions functional support (SFUNLINK) and program (SCEXLINK) link libraries.

**2** The Archive Manager uses the archive data set name (see "Archive Journal" on page 200) to resolve %ARCDSN. The remaining ARCOUT DD statement contains whatever attributes you would use to allocate the data set.

You can override the archive data set name (set in the ISPF panel) by entering a GDG or symbols and literals. See ARCOUT DD for details.

**3** To calculate appropriate values, consider the size of the active journal data sets and the maximum active, maximum time, maximum size, and maximum volumes settings on the archive journal data set template.

**4** We use the LIKE parameter to copy the data set attributes of the *active* journal data set for the IMS Connect system. Enter the active data set name (see "Configuring active journals" on page 67) suffixed by "`.P01`". For example, if the active journal panel shows "`CEX240.ACTIVE.HWS1`" use:

```
CEX240.ACTIVE.HWS1.P01
```

**Note:** If you do not use the LIKE parameter you need to specify the data set attributes (DCB). It is recommended that you use the LIKE parameter since it is available in any site with SMS.

When the Archive Manager completes the job, it will look similar to this:

```
//ABC JOB (ACCOUNT),'NAME' 1
//STEP01   EXEC PGM=FUNEXEC,PARM=CEXJARC,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=funpre.SFUNLINK
//         DD DISP=SHR,DSN=cexpre.SCEXLINK
//ARCOUT   DD  DSN=USER.D060206.T150645, 2
//             DISP=(,CATLG,DELETE),
//             SPACE=(CYL,(20,20),RLSE),
//             LIKE=CEX240.ACTIVE.HWS1.P01
//REPOSTRY DD  DISP=SHR,DSN=CEX240.REPOSTRY.MM001 3
//ARCHCNTL DD  *
CEX,08,HWS1
/*
//CUSTOM   DD  DUMMY          *** GENERATED STMT ***
//*
```

**1** Job statement information from the Active Journal Data Set panel. See "Configuring active journals" on page 67.

**2** The resolved data set name. In this example, containing the date and time the archive data set was created.

**3** The REPOSTRY, ARCHCNTL, and CUSTOM DD statements are automatically generated by the Archive Manager.

## Archiving to magnetic tape

Here is a simple JCL skeleton for archiving active journals to magnetic tape. A similar skeleton, CEXARCH2, is included in the SCEXSAMP library.

```
//STEP01   EXEC PGM=FUNEXEC,PARM=CEXJARC,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=funpre.SFUNLINK
//         DD DISP=SHR,DSN=cexpre.SCEXLINK 1
//ARCOUT   DD  DSN=%ARCDSN, 2
//             DISP=(,CATLG,DELETE),
//             UNIT=(TAPE,,DEFER)
```

**1** Enter the high-level qualifier of the IMS Connect Extensions functional support (SFUNLINK) and program (SCEXLINK) link libraries.

**2** The Archive Manager uses the archive data set name (see "Archive Journal" on page 200) to resolve `%ARCDSN`. Use the remaining ARCOUT DD statement as you would normally to allocate a data set to tape.

You can override the archive data set name (set in the ISPF panel) by entering a GDG or symbols and literals. See ARCOUT DD for details.

When the Archive Manager completes the job, it will look similar to this:

```
//ABC JOB (ACCOUNT),'NAME' 1
//STEP01   EXEC PGM=FUNEXEC,PARM=CEXJARC,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=funpre.SFUNLINK
//         DD DISP=SHR,DSN=cexpre.SCEXLINK
//ARCOUT   DD  DSN=USER.D060206.T150645, 2
//             DISP=(,CATLG,DELETE),
//             UNIT=(TAPE,,DEFER)
//REPOSTRY DD  DISP=SHR,DSN=CEX240.REPOSTRY.MM001 3
//ARCHCNTL DD  *
CEX,08,HWS1
/*
//CUSTOM   DD  DUMMY          *** GENERATED STMT ***
//*
```

**1** Job statement information from the Active Journal Data Set panel. See "Configuring active journals" on page 67.

**2** The resolved data set name. In this example, containing the date and time the archive data set was created.

**3** The REPOSTRY, ARCHCNTL, and CUSTOM DD statements are automatically generated by the Archive Manager.

## Archive to tape and print report

Here is a JCL skeleton for archiving active journals to magnetic tape. In this example, an additional job step is run to create a summary report.

A similar skeleton, CEXARCH3, is included in the SCEXSAMP library.

```
//STEP01   EXEC PGM=FUNEXEC,PARM=CEXJARC,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=funpre.SFUNLINK
//         DD DISP=SHR,DSN=cexpre.SCEXLINK
//ARCOUT   DD  DSN=%ARCDSN,      1
//             DISP=(,CATLG,DELETE),
//             UNIT=(TAPE,,DEFER),
//             VOL=(,RETAIN)
//*
//* *******************************************************
//* ** Print a summary report                            **
//* *******************************************************
//*
//STEP02   EXEC PGM=FUNEXEC,REGION=8M,PARM='CEXDFPRT' 2
//STEPLIB  DD DISP=SHR,DSN=funpre.SFUNLINK
//         DD DISP=SHR,DSN=cexpre.SCEXLINK
//MSGOUT   DD  SYSOUT=A
//EVNTIN   DD  DISP=OLD,DSN=*.STEP01.ARCOUT, 3
//             VOL=(REF=*.STEP01.ARCOUT)
//*
```

**1** Archiving must be the first job step. The job fails otherwise.

**2** In this example, the second step executes the IMS Connect Extensions print utility program to print a summary report but you can use any program in other job steps.

**3** The EVENTIN DD demonstrates that you can refer back to the archive data set name.

When the Archive Manager completes the job, it will look similar to this:

```
//ABC JOB (ACCOUNT),'NAME' 1
//STEP01   EXEC PGM=FUNEXEC,PARM=CEXJARC,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=funpre.SFUNLINK
//         DD DISP=SHR,DSN=cexpre.SCEXLINK
//ARCOUT   DD  DSN=USER.D060206.T150645, 2
//             DISP=(,CATLG,DELETE),
//             UNIT=(TAPE,,DEFER),
//             VOL=(,RETAIN)
//REPOSTRY DD  DISP=SHR,DSN=CEX240.REPOSTRY.MM001 3
//ARCHCNTL DD  *
CEX,08,HWS1
/*
//CUSTOM   DD  DUMMY          *** GENERATED STMT ***
//*
//* *******************************************************
//* ** Print a summary report                           **
//* *******************************************************
//*
//STEP02   EXEC PGM=FUNEXEC,REGION=8M,PARM='CEXDFPRT'
//STEPLIB  DD DISP=SHR,DSN=funpre.SFUNLINK
//         DD DISP=SHR,DSN=cexpre.SCEXLINK

//MSGOUT   DD  SYSOUT=A
//EVNTIN   DD  DISP=OLD,DSN=*.STEP01.ARCOUT,
//             VOL=(REF=*.STEP01.ARCOUT)
//*
```

**1** Job statement information from the Active Journal Data Set panel. See "Configuring active journals."

**2** The resolved data set name. In this example, containing the date and time the archive data set was created.

**3** The REPOSTRY, ARCHCNTL, and CUSTOM DD statements are automatically generated by the Archive Manager. Additional steps appear after the CUSTOM DD statement.

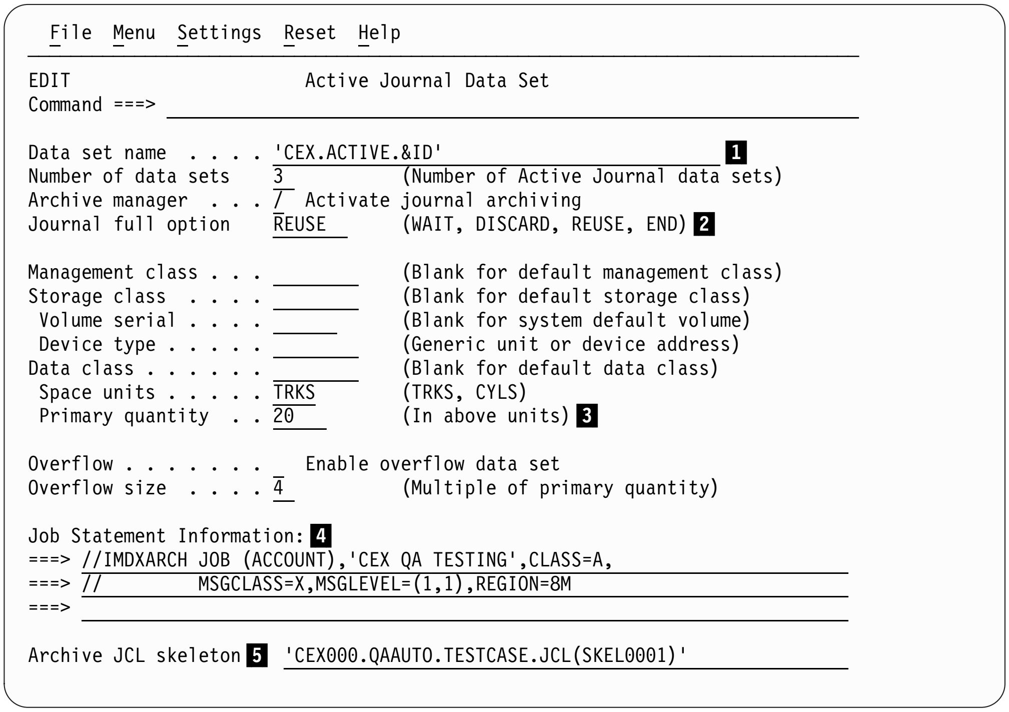
## Configuring active journals

From the Active Journal Data Set panel, you can set active journal options and the archive Job Statement Information and JCL skeleton.

### Procedure

1. Create or edit a system definition from option 1.1 **System Definitions**
2. In the system definition, enter / (slash) next to the **Active Data Set** field to edit the active Journal Data Set template details. The Active Journal Data Set panel appears.

### Results



```
  File  Menu  Settings  Reset  Help
 ______________________________________________________________________________
 EDIT                      Active Journal Data Set
 Command ===>

 Data set name  . . . . 'CEX.ACTIVE.&ID'                            1
 Number of data sets    3          (Number of Active Journal data sets)
 Archive manager  . . . /  Activate journal archiving
 Journal full option    REUSE      (WAIT, DISCARD, REUSE, END) 2

 Management class . . .            (Blank for default management class)
 Storage class  . . . .            (Blank for default storage class)
  Volume serial . . . .            (Blank for system default volume)
  Device type . . . . .            (Generic unit or device address)
 Data class . . . . . .            (Blank for default data class)
  Space units . . . . . TRKS       (TRKS, CYLS)
  Primary quantity  . . 20         (In above units) 3

 Overflow . . . . . . . _  Enable overflow data set
 Overflow size  . . . . 4          (Multiple of primary quantity)

 Job Statement Information: 4
 ===> //IMDXARCH JOB (ACCOUNT),'CEX QA TESTING',CLASS=A,
 ===> //         MSGCLASS=X,MSGLEVEL=(1,1),REGION=8M
 ===>

 Archive JCL skeleton 5  'CEX000.QAAUTO.TESTCASE.JCL(SKEL0001)'
```

*Figure 24. The active journal data set template details*

1 Your active journals are allocated with the name you enter and the suffix "`.Pnn`", where *nn* is the active journal number, for example, `01`, `02`, `03`. If you archive to DASD, note this name. You will need it when setting the attributes of the archive data set.

2 This option determines what happens if all active journal data sets and Overflows (if specified) are full: this is an exceptional condition. Select REUSE to reuse active journal data sets even if the data in them has not been archived, select WAIT to hold all transaction processing until an active journal is archived.

3 See Table 1 on page 62 for sizing estimations.

If you manually allocate an active journal data set to increase its primary space allocation, note that the DASD space must be contiguous, and secondary extents are not permitted.

4 The Job Statement information will appear at the top of the archive job that the Archive Manager generates. You can enter up to three lines. Usually, you can use the default.

5 The Archive JCL skeleton is where you enter the fully qualified name of the archive skeleton you created. It is the archive skeleton that determines the attributes of the archive data sets. It can also contain additional job steps.

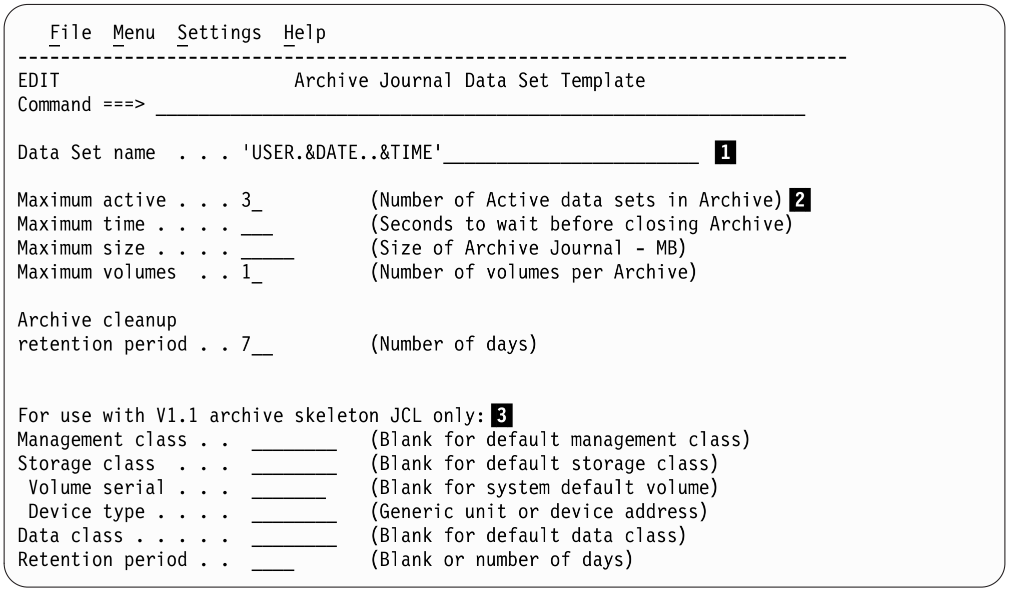
## Creating an archive journal template

From the Archive Journal Data Set panel, enter the archive data set name you want to use.

## Procedure

1. Edit the system definition with the active journal settings you created in "Configuring active journals" on page 67.
2. In the system definition, enter / (slash) next to the **Archive Data Set** field to edit the archive Journal Data Set template details. The Archive Journal Data Set panel appears.

## Results



```
  File  Menu  Settings  Help
----------------------------------------------------------------------------
EDIT                     Archive Journal Data Set Template
Command ===> ____________________________________________________________

Data Set name  . . . 'USER.&DATE..&TIME'________________________ 1

Maximum active . . . 3_         (Number of Active data sets in Archive) 2
Maximum time . . . . ___        (Seconds to wait before closing Archive)
Maximum size . . . . _____      (Size of Archive Journal - MB)
Maximum volumes  . . 1_         (Number of volumes per Archive)

Archive cleanup
retention period . . 7__        (Number of days)


For use with V1.1 archive skeleton JCL only: 3
Management class . .  ________  (Blank for default management class)
Storage class  . . .  ________  (Blank for default storage class)
 Volume serial . . .  _______   (Blank for system default volume)
 Device type . . . .  ________  (Generic unit or device address)
Data class . . . . .  ________  (Blank for default data class)
Retention period . .  ____      (Blank or number of days)
```

*Figure 25. The archive journal data set template details*

**1** Set the name template of the archive data set.

When it generates an archiving job, the Archive Manager uses the literals and symbols you specify in this field to resolve a unique archive data set name. The Archive Manager then replaces the symbol %ARCHDSN with this resolved data set name.

**Note:** You can override the data set name by using literals and symbols directly in the ARCOUT DD statement of the archive skeleton JCL.

**2** You can use the Maximum time, Maximum active, Maximum size, and Maximum volumes fields to tailor how many active journals the Archive Manager can write to one archive data set and how big each archive data set can be before another data set must be created.

**3** The V1.1 skeleton fields have been obsoleted. These fields are only used with archive JCL skeletons from IMS Connect Extensions V1.1. If you have such skeletons, consider upgrading them rather than using these fields.

See:

- "Archive Journal" on page 200 for details on this panel.
- Figure 23 on page 64 for details on the Archive Manager.

# Resizing or redefining active journals

To change the attributes of active journals, you must stop the IMS Connect, delete or rename the existing journal data sets, change the active journal definition, and then restart IMS Connect.

## About this task

After IMS Connect Extensions creates active journals it reuses them as needed. This means that even if you change the allocation parameters of your active journals, you will not have a direct effect on the active journal data sets themselves. To change the size of active journals, you must also delete or rename the journal data sets. When IMS Connect Extensions sees that a required active journal data set does not exist, it will automatically create new ones using the allocation settings that are current for that system definition at that time.

However, simply deleting the active journal data sets could cause errors such as these:

```
FUN2530F Allocation failed for CEX999.QAACT001.ACTIVE.HWS4.P01
FUN2598I IKJ56228I DATA SET CEX999.QAACT001.ACTIVE.HWS4.P01 NOT IN CATALOG
FUN2542W No Active Journal data sets available
FUN2543W No Overflow data sets available
FUN2594W 12 blocks of journal data skipped
```

If you manually allocate an active journal data set to increase its primary space allocation, note that the DASD space must be contiguous, and secondary extents are not permitted.

## Procedure

To correctly resize or redefine journals, follow these steps:

1. Stop the IMS Connect system whose journals you want to resize or redefine. During normal shutdown, IMS Connect Extensions will archive the data in the active journals for you. Confirm that archiving completed successfully by looking for a 0 return code for the archiving job, and this message in the JES output of the archive job:

   ```
   FUN2514I Journal archive complete, n active journal(s) processed
   ```
2. Delete or rename the existing active journal data sets.
3. From the IMS Connect Extensions primary options menu, select option 1.1 **System Definitions**.
4. Enter **S** next to the system and change its active journal settings.
5. Enter / (slash) next to the Active Data Set.
6. In the Active Journal Data Set panel, enter the primary command RESET. When asked to confirm the action, press Enter again.

   The **RESET** command on this panel resets the internal checkpoint record in IMS Connect Extensions for the active journals.
7. Edit the active journal definitions to include the new sizing parameters.
8. Press the Exit function key (F3) to save and exit.
9. Restart the IMS Connect system.

# Chapter 5. Reporting IMS Connect activity

The following topics provide information on how you can use IMS Connect Extensions with IMS PA and IMS PI to report IMS Connect activity.

**Related reference**:

"Print utility" on page 324
The print utility produces report output of formatted IMS Connect Extensions event records. It accepts input from an Active Journal or Archive Journal data set.

## Reporting and analysis with IMS Performance Analyzer

In-depth analysis and reporting of IMS Connect event records is provided by the IBM IMS Performance Analyzer.

IMS Performance Analyzer for z/OS is a performance analysis and tuning aid for IMS DB and IMS TM systems. It processes Log, Monitor, and IMS Connect event data, to provide comprehensive reports for use by managers, database administrators, communications administrators, and system programmers to analyze and tune their IMS systems.

IMS PA is a tool for managers, system programmers, and technical support personnel. It produces management-oriented reports and it produces summary and detailed information for those directly involved in improving system performance. IMS PA can help you:

- Improve system performance
- Improve transaction transit time to manage TCP/IP communication more efficiently
- Use IMS regions and message queues more efficiently
- Reduce virtual and real storage requirements in buffer pools
- Increase the availability of IMS resources
- Evaluate applications and programs against system standards before installation
- Do ongoing system measurement and management reporting
- Debug IMS applications
- Increase the productivity of analysts and programmers
- Determine future system requirements
- Enhance system and program documentation
- Reduce the need to run IMS utilities
- Reduce the requirement to run the monitor reports supplied with IMS
- Provide auditors with valuable data for a number of potential audit tasks

IMS Performance Analyzer provides a comprehensive set of reports from the IMS Connect performance and accounting data collected by IMS Connect Extensions. The reports provide a summary and detailed analysis of IMS Connect transaction transit time, resource usage and resource availability.

The IMS Connect transaction index is generated by IMS Performance Analyzer from IMS Connect Extensions journals. It collates all the information that can be known about the transaction from the available IMS Connect Extensions journal records. This information may include the client ID, IP Address, overall performance, timings of significant events, IMS Connect-exit specific performance, and dozens of additional pieces of information. You can use this index as both a reduced form of the IMS Connect Extensions journal and as input to IMS Problem Investigator and IMS Performance Analyzer.

IMS Performance Analyzer groups reports by functional category:

- Transaction Transit Reports
- Resource Usage Reports
- Trace Reports

Selection Criteria enable you to filter your reporting, for example to include data only for a particular Transaction code, user ID, datastore and only for a specific period of time.

The following sections provide some examples of using IMS Performance Analyzer reports in IMS Connect performance analysis and tuning. For further information, see the *IMS Performance Analyzer Report Reference*.

## Transaction transit reports

These reports provide performance statistics to measure the performance of your IMS Connect transactions.

Transaction Transit (response) time is broken down into its components: Input, Processing (by OTMA), Acknowledgement from the client, and Output. They can help identify any bottlenecks in transaction flow, and are used for monitoring system performance, gathering diagnostic information, and tuning IMS.

### Transit analysis report

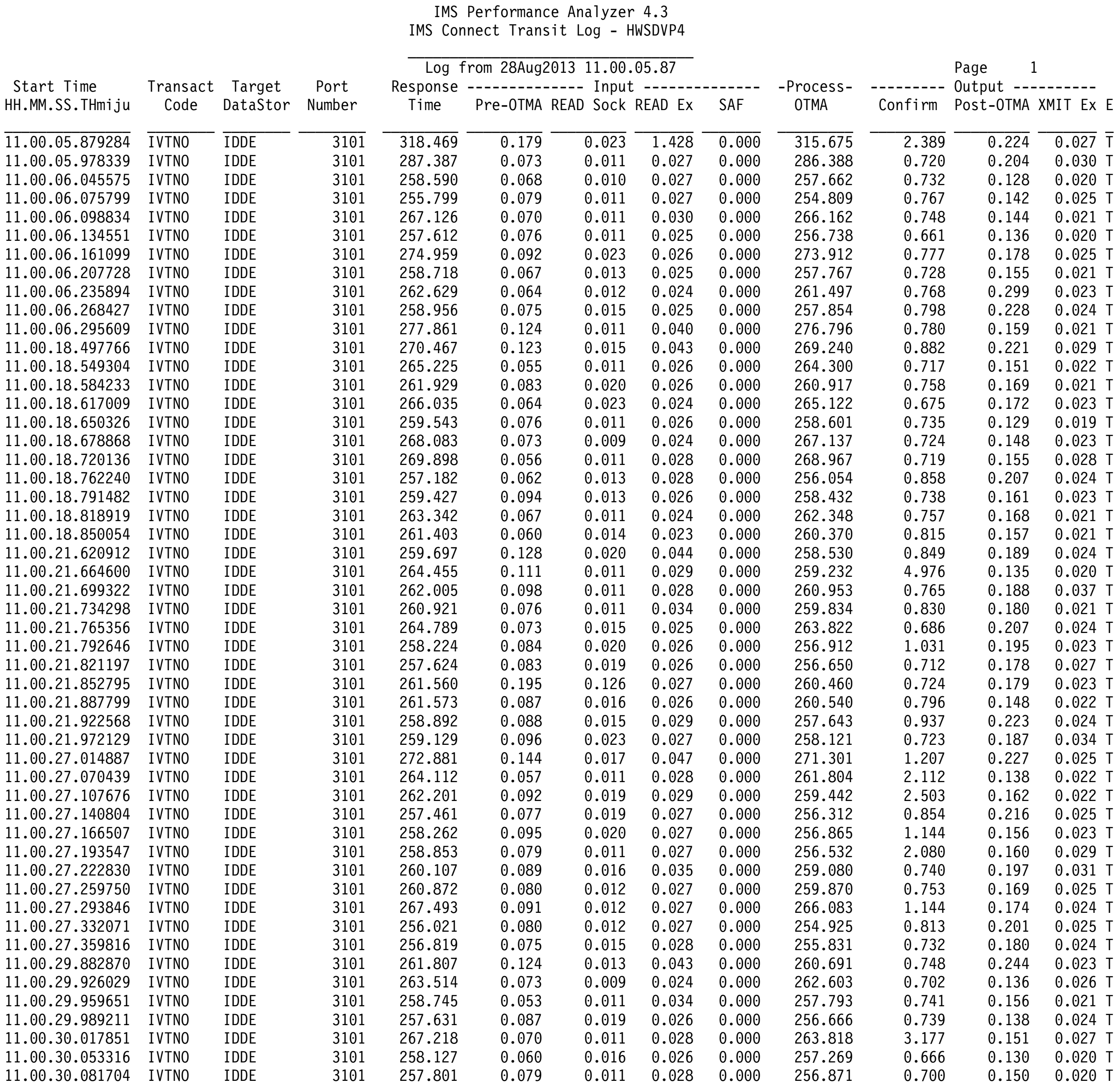
The Transit Analysis provides a summary of IMS Connect transaction performance. You can summarize performance data by one or two sort keys including time of day, transaction code, user ID, datastore (originating and target) and port number. Performance statistics are provided as averages, and optionally, peak percentiles (for example, 90% of transactions completed within the reported time).

```
                                                   IMS Performance Analyzer 4.3
                                              IMS Connect Transit Analysis - HWSDVP4

                                       From 28Aug2013 11.00.05.87  To 28Aug2013 12.02.39.98                 Page     1
Transact   Time   Message       Response -------------- Input --------------  -Process-  --------- Output ----------  Rate Time
  Code           Count         Time   Pre-OTMA READ Sock READ Ex   SAF      OTMA     Confirm  Post-OTMA XMIT Ex  /Sec Outs  NAK
________ ________ _______      _________ _________ _________ _______ _______  _________  _________ _________ _______  ____ ____ ____
ADDINV   11.01.00      20  Avg   264.024     0.070     0.013   0.028   0.000    262.911      0.905     0.136   0.019     0   20    0
                           80%   267.018     0.076     0.014   0.033   0.000    265.903      1.049     0.151   0.021
         11.02.00       5  Avg   262.578     0.082     0.013   0.027   0.000    261.340      1.010     0.146   0.022     0    5    0
                           80%   264.531     0.102     0.015   0.028   0.000    263.015      1.333     0.162   0.027
         12.02.00      25  Avg    15.763     0.151     0.018   0.018   0.000     15.416      0.000     0.195   0.012     0    0    0
                           80%    19.850     0.326     0.030   0.021   0.000     19.493      0.000     0.248   0.013
ADDINV   Subtotal      50  Avg   139.749     0.112     0.015   0.023   0.000    139.007      0.463     0.167   0.016     1   25    0
                           80%   245.262     0.239     0.024   0.028   0.000    244.183      0.878     0.212   0.020

ADDPART  11.01.00      16  Avg   265.352     0.072     0.013   0.027   0.000    264.260      0.878     0.140   0.019     0   16    0
                           80%   270.391     0.082     0.015   0.028   0.000    269.349      1.107     0.165   0.021
         11.02.00       4  Avg   261.480     0.075     0.014   0.027   0.000    260.324      0.960     0.120   0.018     0    4    0
                           80%   261.870     0.084     0.015   0.028   0.000    260.958      1.226     0.128   0.020
         12.02.00      20  Avg    18.446     0.094     0.017   0.018   0.000     17.880      0.000     0.471   0.013     0    0    0
                           80%    25.077     0.110     0.022   0.019   0.000     24.153      0.000     1.326   0.015
ADDPART  Subtotal      40  Avg   141.512     0.083     0.015   0.022   0.000    140.677      0.447     0.304   0.016     1   20    0
                           80%   246.606     0.099     0.019   0.026   0.000    245.532      0.861     0.917   0.019

DLETINV  11.01.00      15  Avg   263.375     0.073     0.012   0.030   0.000    262.091      1.078     0.132   0.023     0   15    0
                           80%   265.725     0.087     0.013   0.038   0.000    264.352      1.528     0.169   0.036
         11.02.00       5  Avg   262.874     0.079     0.012   0.027   0.000    261.724      0.943     0.126   0.021     0    5    0
                           80%   263.772     0.103     0.013   0.030   0.000    262.516      1.192     0.148   0.023
         12.02.00      20  Avg    21.738     0.101     0.017   0.018   0.000     21.392      0.000     0.244   0.014     0    0    0
                           80%    33.000     0.120     0.021   0.021   0.000     32.525      0.000     0.374   0.015
DLETINV  Subtotal      40  Avg   142.494     0.088     0.015   0.024   0.000    141.696      0.522     0.187   0.018     1   20    0
                           80%   245.775     0.109     0.018   0.031   0.000    244.585      1.050     0.293   0.027
 .
 .
 .
IVTNV    11.00.00      44  Avg   265.771     0.086     0.012   0.031   0.000    264.669      0.853     0.161   0.023     1   44    0
                           80%   270.542     0.130     0.015   0.055   0.000    269.441      1.108     0.187   0.028
         11.01.00      22  Avg   258.883     0.074     0.013   0.027   0.000    257.880      0.763     0.164   0.023     0   22    0
                           80%   261.447     0.091     0.015   0.032   0.000    260.411      0.860     0.189   0.030
         12.01.00      55  Avg    18.968     0.279     0.020   0.069   0.000     18.368      0.000     0.320   0.014     1    0    0
                           80%    34.297     0.700     0.029   0.266   0.000     33.709      0.000     0.732   0.016
IVTNV    Subtotal     121  Avg   152.335     0.171     0.016   0.048   0.000    151.480      0.449     0.234   0.019     2   66    0
                           80%   255.850     0.467     0.023   0.182   0.000    254.798      0.831     0.519   0.025

PART     11.01.00      36  Avg   268.255     0.080     0.014   0.031   0.000    266.966      1.063     0.144   0.021     1   36    0
                           80%   290.213     0.104     0.019   0.045   0.000    288.818      1.493     0.184   0.024
         11.02.00       9  Avg   264.284     0.079     0.014   0.028   0.000    262.621      1.426     0.156   0.023     0    9    0
                           80%   266.050     0.100     0.018   0.033   0.000    264.117      2.121     0.187   0.026
         12.02.00      45  Avg    19.877     0.220     0.014   0.054   0.000     19.405      0.000     0.252   0.014     1    0    0
                           80%    34.169     0.736     0.017   0.248   0.000     33.346      0.000     0.409   0.016
PART     Subtotal      90  Avg   143.669     0.150     0.014   0.042   0.000    142.751      0.568     0.199   0.018     2   45    0
                           80%   249.868     0.518     0.018   0.179   0.000    248.547      1.163     0.322   0.022

--------         -------   --------- --------- --------- ------- -------  ---------  --------- --------- -------  ---- ---- ----
Total                 590  Avg   146.760     0.151     0.016   0.040   0.000    145.875      0.507     0.226   0.018     0  306    0
                           80%   250.821     0.572     0.042   0.172   0.000    249.633      1.004     0.489   0.024
```

*Figure 26. IMS Performance Analyzer Transit Analysis Report, ordered by TRANCODE, TIME*

## Transit Log report

The Transit Log provides performance details about every transaction processed by IMS Connect. Information from event records is collected to provide a complete picture of transaction processing. The order of transactions in the report is based on when they end, not when they start.



```
                                         IMS Performance Analyzer 4.3
                                       IMS Connect Transit Log - HWSDVP4
                                       ____________________________________
                                         Log from 28Aug2013 11.00.05.87                                 Page     1
 Start Time       Transact  Target    Port      Response -------------- Input --------------  -Process-  --------- Output ----------
HH.MM.SS.THmiju     Code   DataStor  Number      Time    Pre-OTMA READ Sock READ Ex    SAF      OTMA      Confirm  Post-OTMA XMIT Ex E
_______________  ________  ________  ________  _________  ________ ________  _______  _______  _________  _________  ________  _______ _
11.00.05.879284  IVTNO     IDDE          3101    318.469     0.179    0.023    1.428    0.000    315.675      2.389     0.224    0.027 T
11.00.05.978339  IVTNO     IDDE          3101    287.387     0.073    0.011    0.027    0.000    286.388      0.720     0.204    0.030 T
11.00.06.045575  IVTNO     IDDE          3101    258.590     0.068    0.010    0.027    0.000    257.662      0.732     0.128    0.020 T
11.00.06.075799  IVTNO     IDDE          3101    255.799     0.079    0.011    0.027    0.000    254.809      0.767     0.142    0.025 T
11.00.06.098834  IVTNO     IDDE          3101    267.126     0.070    0.011    0.030    0.000    266.162      0.748     0.144    0.021 T
11.00.06.134551  IVTNO     IDDE          3101    257.612     0.076    0.011    0.025    0.000    256.738      0.661     0.136    0.020 T
11.00.06.161099  IVTNO     IDDE          3101    274.959     0.092    0.023    0.026    0.000    273.912      0.777     0.178    0.025 T
11.00.06.207728  IVTNO     IDDE          3101    258.718     0.067    0.013    0.025    0.000    257.767      0.728     0.155    0.021 T
11.00.06.235894  IVTNO     IDDE          3101    262.629     0.064    0.012    0.024    0.000    261.497      0.768     0.299    0.023 T
11.00.06.268427  IVTNO     IDDE          3101    258.956     0.075    0.015    0.025    0.000    257.854      0.798     0.228    0.024 T
11.00.06.295609  IVTNO     IDDE          3101    277.861     0.124    0.011    0.040    0.000    276.796      0.780     0.159    0.021 T
11.00.18.497766  IVTNO     IDDE          3101    270.467     0.123    0.015    0.043    0.000    269.240      0.882     0.221    0.029 T
11.00.18.549304  IVTNO     IDDE          3101    265.225     0.055    0.011    0.026    0.000    264.300      0.717     0.151    0.022 T
11.00.18.584233  IVTNO     IDDE          3101    261.929     0.083    0.020    0.026    0.000    260.917      0.758     0.169    0.021 T
11.00.18.617009  IVTNO     IDDE          3101    266.035     0.064    0.023    0.024    0.000    265.122      0.675     0.172    0.023 T
11.00.18.650326  IVTNO     IDDE          3101    259.543     0.076    0.011    0.026    0.000    258.601      0.735     0.129    0.019 T
11.00.18.678868  IVTNO     IDDE          3101    268.083     0.073    0.009    0.024    0.000    267.137      0.724     0.148    0.023 T
11.00.18.720136  IVTNO     IDDE          3101    269.898     0.056    0.011    0.028    0.000    268.967      0.719     0.155    0.028 T
11.00.18.762240  IVTNO     IDDE          3101    257.182     0.062    0.013    0.028    0.000    256.054      0.858     0.207    0.024 T
11.00.18.791482  IVTNO     IDDE          3101    259.427     0.094    0.013    0.026    0.000    258.432      0.738     0.161    0.023 T
11.00.18.818919  IVTNO     IDDE          3101    263.342     0.067    0.011    0.024    0.000    262.348      0.757     0.168    0.021 T
11.00.18.850054  IVTNO     IDDE          3101    261.403     0.060    0.014    0.023    0.000    260.370      0.815     0.157    0.021 T
11.00.21.620912  IVTNO     IDDE          3101    259.697     0.128    0.020    0.044    0.000    258.530      0.849     0.189    0.024 T
11.00.21.664600  IVTNO     IDDE          3101    264.455     0.111    0.011    0.029    0.000    259.232      4.976     0.135    0.020 T
11.00.21.699322  IVTNO     IDDE          3101    262.005     0.098    0.011    0.028    0.000    260.953      0.765     0.188    0.037 T
11.00.21.734298  IVTNO     IDDE          3101    260.921     0.076    0.011    0.034    0.000    259.834      0.830     0.180    0.021 T
11.00.21.765356  IVTNO     IDDE          3101    264.789     0.073    0.015    0.025    0.000    263.822      0.686     0.207    0.024 T
11.00.21.792646  IVTNO     IDDE          3101    258.224     0.084    0.020    0.026    0.000    256.912      1.031     0.195    0.023 T
11.00.21.821197  IVTNO     IDDE          3101    257.624     0.083    0.019    0.026    0.000    256.650      0.712     0.178    0.027 T
11.00.21.852795  IVTNO     IDDE          3101    261.560     0.195    0.126    0.027    0.000    260.460      0.724     0.179    0.023 T
11.00.21.887799  IVTNO     IDDE          3101    261.573     0.087    0.016    0.026    0.000    260.540      0.796     0.148    0.022 T
11.00.21.922568  IVTNO     IDDE          3101    258.892     0.088    0.015    0.029    0.000    257.643      0.937     0.223    0.024 T
11.00.21.972129  IVTNO     IDDE          3101    259.129     0.096    0.023    0.027    0.000    258.121      0.723     0.187    0.034 T
11.00.27.014887  IVTNO     IDDE          3101    272.881     0.144    0.017    0.047    0.000    271.301      1.207     0.227    0.025 T
11.00.27.070439  IVTNO     IDDE          3101    264.112     0.057    0.011    0.028    0.000    261.804      2.112     0.138    0.022 T
11.00.27.107676  IVTNO     IDDE          3101    262.201     0.092    0.019    0.029    0.000    259.442      2.503     0.162    0.022 T
11.00.27.140804  IVTNO     IDDE          3101    257.461     0.077    0.019    0.027    0.000    256.312      0.854     0.216    0.025 T
11.00.27.166507  IVTNO     IDDE          3101    258.262     0.095    0.020    0.027    0.000    256.865      1.144     0.156    0.023 T
11.00.27.193547  IVTNO     IDDE          3101    258.853     0.079    0.011    0.027    0.000    256.532      2.080     0.160    0.029 T
11.00.27.222830  IVTNO     IDDE          3101    260.107     0.089    0.016    0.035    0.000    259.080      0.740     0.197    0.031 T
11.00.27.259750  IVTNO     IDDE          3101    260.872     0.080    0.012    0.027    0.000    259.870      0.753     0.169    0.025 T
11.00.27.293846  IVTNO     IDDE          3101    267.493     0.091    0.012    0.027    0.000    266.083      1.144     0.174    0.024 T
11.00.27.332071  IVTNO     IDDE          3101    256.021     0.080    0.012    0.027    0.000    254.925      0.813     0.201    0.025 T
11.00.27.359816  IVTNO     IDDE          3101    256.819     0.075    0.015    0.028    0.000    255.831      0.732     0.180    0.024 T
11.00.29.882870  IVTNO     IDDE          3101    261.807     0.124    0.013    0.043    0.000    260.691      0.748     0.244    0.023 T
11.00.29.926029  IVTNO     IDDE          3101    263.514     0.073    0.009    0.024    0.000    262.603      0.702     0.136    0.026 T
11.00.29.959651  IVTNO     IDDE          3101    258.745     0.053    0.011    0.034    0.000    257.793      0.741     0.156    0.021 T
11.00.29.989211  IVTNO     IDDE          3101    257.631     0.087    0.019    0.026    0.000    256.666      0.739     0.138    0.024 T
11.00.30.017851  IVTNO     IDDE          3101    267.218     0.070    0.011    0.028    0.000    263.818      3.177     0.151    0.027 T
11.00.30.053316  IVTNO     IDDE          3101    258.127     0.060    0.016    0.026    0.000    257.269      0.666     0.130    0.020 T
11.00.30.081704  IVTNO     IDDE          3101    257.801     0.079    0.011    0.028    0.000    256.871      0.700     0.150    0.020 T
```

*Figure 27. IMS Performance Analyzer Transit Log Report*

## Transit extract

The IMS Connect Transit Extract gathers performance details about every transaction processed by IMS Connect.

You can request a List Extract or Summary Extract or both. The List Extract provides similar details to the Transit Log report, while the Summary Extract summarizes these details over a specified time interval, typically 15 minutes. The extract data is suitable for importing into DB2 or PC tools from where you can run queries or produce reports and graphs.

# Resource usage reports

The resource usage reports contain detailed and summary information on the use and availability of IMS Connect resources, including TCP/IP ports and tpipes.

## Port usage report

The Port Usage report provides a summary of the TCP/IP ports used by the IMS Connect system. For each port, average statistics are provided for port depth, message processed count, and ACCEPT, READ, and WRITE socket counts. Peak percentile statistics are provided for Input READ and ACK/NAK READ Socket counts.

```
                                          IMS Performance Analyzer 4.3
                                        IMS Connect Port Usage - MKT040
                                        ________________________________
                            From 05Aug2011 12.49.46.73  To 05Aug2011 12.55.54.06                      Page     1
       -- Depth --   Message    ACCEPT   ---- READ -----  --- Input READ ----  -- ACK/NAK READ ---  ---- WRITE ----
 Port   Avg   Max     Count      Count      Count    Len    Average   90% Peak   Average   90% Peak    Count    Len
 _____  _____ _____  _________  _________  _________ _____  _________ _________  _________ _________  _________ _____
  7801     3     5       1481         16       5896    53      0.419     1.316     28.778   312.099       2939   162
  7802    14    20       2084         20       8376    54      0.343     1.045     69.580   505.263       4168   162
  7803     3     4        374          4       1504    54      0.324     0.669    106.111   644.518        748   162
  7804     3     4        417          4       1676    54      0.370     0.981    114.022   666.428        834   162
```

```
                                          IMS Performance Analyzer 4.3
                                        IMS Connect Port Usage - PFX040
                                        ________________________________
                            From 28Aug2013 16.21.05.92  To 28Aug2013 16.27.29.06                      Page     1
       -- Depth --   Message    ACCEPT   ---- READ -----  --- Input READ ----  -- ACK/NAK READ ---  ---- WRITE ----
 Port   Avg   Max     Count      Count      Count    Len    Average   90% Peak   Average   90% Peak    Count    Len
 _____  _____ _____  _________  _________  _________ _____  _________ _________  _________ _________  _________ _____
  7801     1     1       4761          2      19044    54      0.168     0.962      1.115     8.799       9521   162
  7802     1     1       3814          2      15254    54      0.163     0.786      0.988    10.649       7626   162
  7803     1     2       4495          2      17978    54      0.199     2.461      1.023     8.048       8989   162
  7804     1     1       3664          2      14654    54      0.178     1.270      0.969    21.847       7326   162
  7805     1     2       4395          2      17578    54      0.165     0.844      0.771     4.453       8789   162
  7806     1     1       3582          2      14326    54      0.166     1.253      0.859     7.684       7162   162
  7807     1     2       4368          2      17470    54      0.150     0.618      0.966    10.623       8735   162
  7808     1     1       4008          2      16030    54      0.178     1.654      0.633     3.114       8015   161
```

*Figure 28. IMS Performance Analyzer Port Usage Report*

## Resume Tpipe report

The Resume Tpipe report provides a summary of resume tpipe command activity. The report provides command statistics, including command count, and a breakdown by command type: Auto (with timeout), No Auto, and Single. Command statistics include count of commands issued, IMS messages received, Negative responses (tpipe queue empty), NAK, and timeout interval.

```
                                          IMS Performance Analyzer 4.3
                                       IMS Connect Resume Tpipe - HWSDVP4
                                       __________________________________
                            From 07May2013 09.22.48.91  To 07May2013 09.23.47.22                      Page     1
          --------------- Noauto ---------------  ---------------- Auto ----------------  --------- Single ---------
                              --- Msg ---    Avg                       --- Msg ---    Avg                         Avg
 Tpipe    Count NResp  Fail  Avg   Max   Timeout  Count NResp  Fail  Avg   Max   Timeout  Count NResp  Fail  Timeout
 ________ _____ _____ _____ _____ _____ _______  _____ _____ _____ _____ _____ _______  _____ _____ _____ _______
 HWS$DEF      1     1     0     0     0    0.25      0     0     0     0     0    0.00      0     0     0    0.00
 IPISRT01     1     1     0     0     0    0.25      0     0     0     0     0    0.00      0     0     0    0.00
 IPISRT02     1     1     0     0     0    0.25      0     0     0     0     0    0.00      0     0     0    0.00
 IPISRT03     0     0     0     0     0    0.00      0     0     0     0     0    0.00      3     3     0    0.25
 IPISRT04     3     3     0     0     0    0.25      0     0     0     0     0    0.00      0     0     0    0.00
 IPISRT05     0     0     0     0     0    0.00      1     1     0     0     0   20.00      0     0     0    0.00
 IPISRT06     1     1     0     0     0    0.25      0     0     0     0     0    0.00      0     0     0    0.00
 REROUTED     1     1     0     0     0    0.25      0     0     0     0     0    0.00      0     0     0    0.00
```

*Figure 29. IMS Performance Analyzer Resume Tpipe Report summarized by Tpipe*

## ACK/NAK report

The ACK/NAK report provides a summary of acknowledgement activity for transactions that use Sync Level=CONFIRM. Positive acknowledgement (ACK) and negative acknowledgement (NAK) statistics are reported for each transaction code. NAK is further broken down as negative acknowledgement from either OTMA (NAK sense code) or from the Client.

```
                                          IMS Performance Analyzer 4.3
                                          IMS Connect ACK/NAK - HWSDVP4
                                         ______________________________
                              From 13Feb2013 15.39.04.19  To 13Feb2013 15.42.02.90                     Page     1
Transact  Target    Confirm  ------ ACK -------  --- Client NAK --- --- OTMA NAK ------------------------------------------
  Code   Datastor    Count     Count  Avg Elaps    Count  Avg Elaps   Count  Sense Code
________ ________  ________  ________ _________  ________ _________ ________  _____________________________________________
DVPTRAN5 IDDF           16           16     1.475         0     0.000        0
DSPALLI  IDDF            2            2   251.279         0     0.000        0
IVTNO    IDDF            8            2     0.949         6     1.044        0
IVTNV    IDDF            3            3    84.420         0     0.000        0
PART     IDDF            2            2   250.993         0     0.000        0
```

*Figure 30. IMS Performance Analyzer ACK/NAK Report*

## Exception Events report

The Exception Events report provides details about events that cause transactions to fail or that signal critical resources are no longer available.

- The List report provides a list (in chronological order) of all exception events.
- The Summary report provides a summary of each exception event encountered and a count of the number of times it occurred.

```
                                                   IMS Performance Analyzer 4.3
                                                     IMS Connect Exception List
                                               _____________________________________
                                                  Report from 24Oct2008 17.14.02.86                               Page     1
 Event Time      System    ID Description                     Information                                         Start Time
_______________  ________  __ _______________________________  ____________________________________________________  _______________
17.14.02.867096  IADH      11 Datastore Un-Available
17.14.32.393973  IADH      2D Datastore status - Normal       Tmember=XCFMIADH DS=IMSC Status=0003
17.14.32.394020  IADH      2D Datastore status - Normal       Tmember=XCFMIADH DS=IMS1 Status=0003
17.26.02.620158  IADH      2D Datastore status - Warning      Tmember=XCFMIADH DS=IMSC Status=0002
17.26.04.620242  IADH      45 OTMA time-out                   Key=C33935502F10CA83 TOV=29                           17.26.02.618972
17.33.44.054865  IADH      2D Datastore status - Severe       Tmember=XCFMIADH DS=IMSC Status=0001
17.33.46.054282  IADH      45 OTMA time-out                   Key=C33937083DEB0943 TOV=29                           17.33.44.051886
17.33.47.058685  IADH      42 Msg from OTMA response is NAK   Key=C339370B1B7D7401 DS=XCFMIADH Tpipe=7901           17.33.47.057745
                                                              RSN=0030
17.33.50.076218  IADH      47 Session error                   Key=C339370EF0DF1941 Type=READ                        17.33.50.076206
17.33.50.082014  IADH      47 Session error                   Key=C339369868312A63 Type=READFAIL                    17.33.50.080570
17.40.26.595114  IADH      2D Datastore status - Normal       Tmember=XCFMIADH DS=IMSC Status=0003
17.41.36.658482  IADH      47 Session error                   Key=C33938CBDF9A5301 Type=READ                        17.41.36.658469

                                                   IMS Performance Analyzer 4.2
                                             IMS Connect Exception Summary - IADH
                                          ___________________________________________
                                       From 24Oct2008 17.14.02.86  To 24Oct2008 17.43.48.86                       Page    16
                                 ID Description                                            Total
                                 __ _________________________________________________  ________
                                 11 Datastore Un-Available                                    1
                                 2D Datastore status - Severe                                 1
                                 2D Datastore status - Warning                                1
                                 2D Datastore status - Normal                                 3
                                 42 Msg from OTMA response is NAK                             1
                                 45 OTMA time-out                                             2
                                 47 Session error                                             3
```

*Figure 31. IMS Performance Analyzer Exception Events Report*

## Log Gap Analysis report

The Log Gap Analysis report contains information on periods of time where log records are not being cut, potentially highlighting an external system event that may have caused IMS to slow down.

```
                                        IMS Performance Analyzer 4.3
                                              Log Gap Analysis
                                              ________________
 Data from:    19.08.58.73  05Aug2011                                                                Page     1
    ID  Time             System      Elapsed     File      LSN               Code
    __  ____             ______      _______     ____      ___               ____
    01  19.30.47.966496  MKT040                  CEX01001  0000000000000032  A048
        19.30.53.211241  MKT040        5.244745  CEX01001  0000000000000033  A00B
    02  19.32.57.818168  MKT040                  CEX01001  0000000000000629  A048
        19.35.26.777484  MKT040      148.959316  CEX01001  000000000000062A  A00B
    03  19.35.26.788590  MKT040                  CEX01001  0000000000000630  A048
        19.35.32.194705  MKT040        5.406115  CEX01001  0000000000000631  A00B
    04  19.57.55.991939  MKT040                  CEX01001  000000000000B317  A048
        19.58.01.427607  MKT040        5.435668  CEX01001  000000000000B318  A00B
    05  19.59.10.575613  MKT040                  CEX01001  000000000000C44C  A048
        19.59.15.990179  MKT040        5.414566  CEX01001  000000000000C44D  A00B
```

*Figure 32. IMS Performance Analyzer Log Gap Analysis Report*

## Trace reports

The trace provides a list of transactions, each with detailed information about every event in the life of that transaction. At a glance, you can see when a transaction starts, followed by all the events associated with the transaction in the order they occurred.

```
                                         IMS Performance Analyzer 4.3
                                         IMS Connect Trace - PFX040
                                         ____________________________
                                        Trace from 28Aug2013 16.21.05.97                          Page     1
Start/+Relative   Elapsed  ID Description                    Information
_______________  ________  __ _______________________________ _____________________________________________________________
16.21.05.975521   *Start*  3C Prepare Read Socket             Key=CBE0DF4789B45384 Depth=1 Port=7801 Sock=9
      +0.000052     0.052  49 Read Socket                     Port=7801 Sock=9
      +0.000081     0.029  3D Message Exit called for READ    Exit=HWSSMPL1
      +0.000136     0.054  3E Message Exit return for READ    RC=00 RSN=00 IP=172.17.69.25 DSOrig=DS05 DSTarg=DS07 TC=PART
                                                              Commit Mode=1 Synch Level=CONFIRM Socket Type=Persistent
      +0.010150    10.014  41 Message sent to OTMA            TPipe=7801 MSG=Transaction AltLTERM=LTAC10
      +0.155437   145.286  42 Message received from OTMA      TPipe=7801 MSG=Data Sense=0000
      +0.156333     0.895  3D Message Exit called for XMIT    Exit=HWSSMPL1
      +0.156353     0.020  3E Message Exit return for XMIT    RC=00 RSN=00
      +0.156425     0.072  4A Write Socket                    Port=7801 Sock=9
      +0.158842     2.416  49 Read Socket                     Port=7801 Sock=9
      +0.158859     0.016  49 Read Socket                     Port=7801 Sock=9
      +0.158866     0.007  3D Message Exit called for READ    Exit=HWSSMPL1
      +0.158883     0.016  3E Message Exit return for READ    RC=00 RSN=00 IP=172.17.69.25 DSOrig=DS05 DSTarg=DS07 TC=PART
                                                              Commit Mode=1 Synch Level=CONFIRM Socket Type=Persistent
      +0.158950     0.067  41 Message sent to OTMA            TPipe=7801 MSG=Response RESP=ACK AltLTERM=LTAC10
      +0.160659     1.708  42 Message received from OTMA      TPipe=7801 MSG=Commit Conf CONFIRM=Committed Sense=0000
      +0.160699     0.039  46 De-allocate Session             RSN=COMT
      +0.160706     0.007  3D Message Exit called for XMIT    Exit=HWSSMPL1
      +0.160714     0.008  3E Message Exit return for XMIT    RC=00 RSN=00
      +0.160744     0.029  4A Write Socket                    Port=7801 Sock=9
      +0.160752     0.008  48 Trigger event                   Trigger=MESSAGE
```

*Figure 33. IMS Performance Analyzer Trace Report*

# Reporting and analysis with IMS Problem Investigator

IBM IMS Problem Investigator for z/OS is a problem analysis tool for IMS DB and IMS TM.

IMS Problem Investigator enables IMS administrators and programmers to interactively explore formatted, interpreted, and easily customizable views of log records; identifying and analyzing problems quickly, without requiring an expert understanding of log data structures and the relationships between log records.

With IMS Problem Investigator, you can:

- View formatted logs with detailed field descriptions.

- Navigate to an exact point in time within a log file.
- Investigate specific problem areas. For example, transaction, database, security, or checkpoint processing.
- Merge log files to combine different aspects of IMS processing into a single view.
- Track the flow of a transaction in a single system or across a sysplex.
- Determine response times and latencies.
- Produce formatted audit reports of transactions and events.

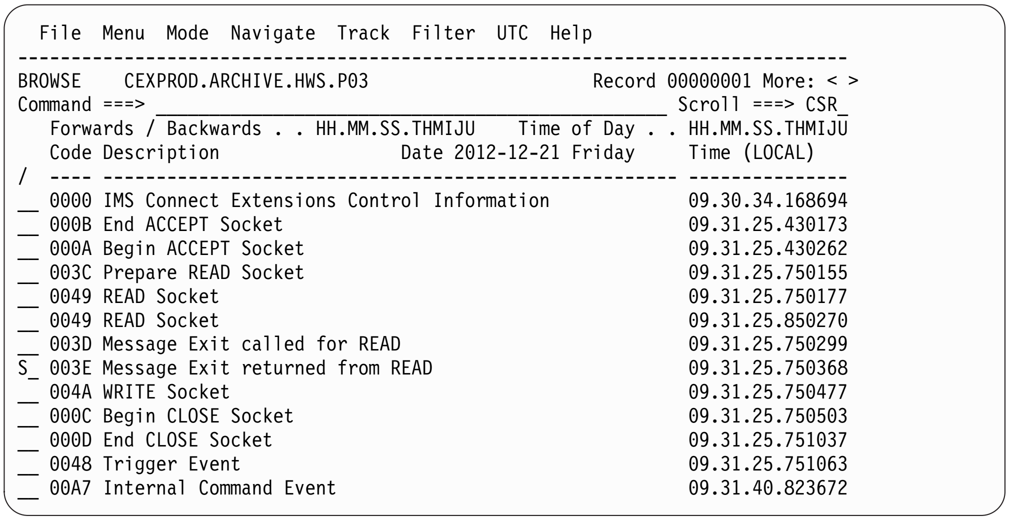
IMS Problem Investigator can interpret IMS Connect event data collected by IMS Connect Extensions and present it in the context of the related IMS log, monitor, and CQS records.

## Browse journal data set

IMS Problem Investigator has a powerful online viewer and batch reporting facility that allows you to investigate problems by navigating through log data quickly and easily to pinpoint the source of problems or track the flow of events.

IMS PI offers its full reporting and viewing capabilities to assist in the analysis of event data in archive journal data sets.

The following figure shows event records from a socket session. You can select records for a formatted browse of record contents. Here, we select a message being rejected by the READ message exit.



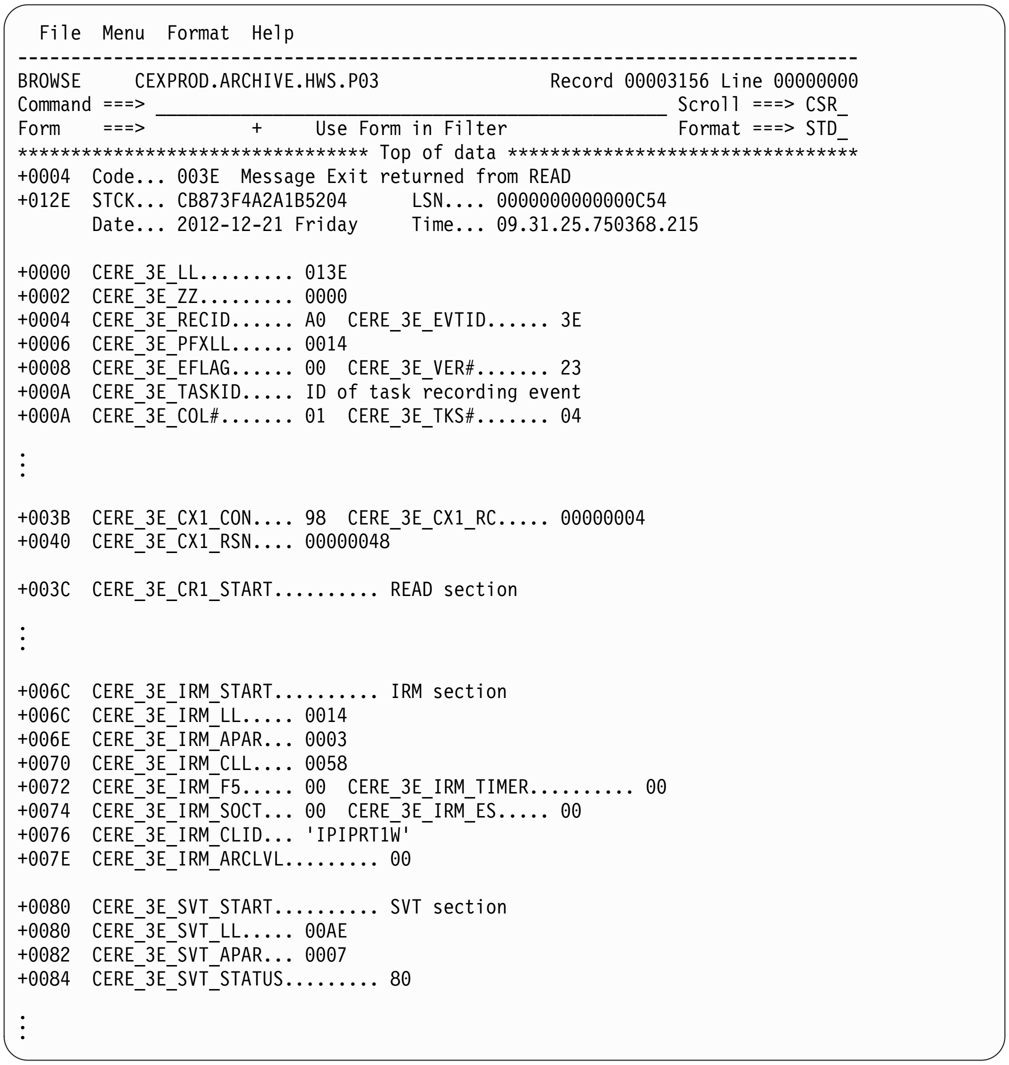
```
  File  Menu  Mode  Navigate  Track  Filter  UTC  Help
------------------------------------------------------------------------------
BROWSE    CEXPROD.ARCHIVE.HWS.P03                    Record 00000001 More: < >
Command ===> ________________________________________________ Scroll ===> CSR_
   Forwards / Backwards . . HH.MM.SS.THMIJU    Time of Day . . HH.MM.SS.THMIJU
   Code Description                 Date 2012-12-21 Friday     Time (LOCAL)
/  ---- -------------------------------------------------- ---------------
__ 0000 IMS Connect Extensions Control Information            09.30.34.168694
__ 000B End ACCEPT Socket                                     09.31.25.430173
__ 000A Begin ACCEPT Socket                                   09.31.25.430262
__ 003C Prepare READ Socket                                   09.31.25.750155
__ 0049 READ Socket                                           09.31.25.750177
__ 0049 READ Socket                                           09.31.25.850270
__ 003D Message Exit called for READ                          09.31.25.750299
S_ 003E Message Exit returned from READ                       09.31.25.750368
__ 004A WRITE Socket                                          09.31.25.750477
__ 000C Begin CLOSE Socket                                    09.31.25.750503
__ 000D End CLOSE Socket                                      09.31.25.751037
__ 0048 Trigger Event                                         09.31.25.751063
__ 00A7 Internal Command Event                                09.31.40.823672
```

*Figure 34. IMS Problem Investigator formatted browse of an archive journal data set*

## Browse event records

You can use IMS Problem Investigator to display event records in a more readable format.

The following figure shows the detail from the READ message exit.

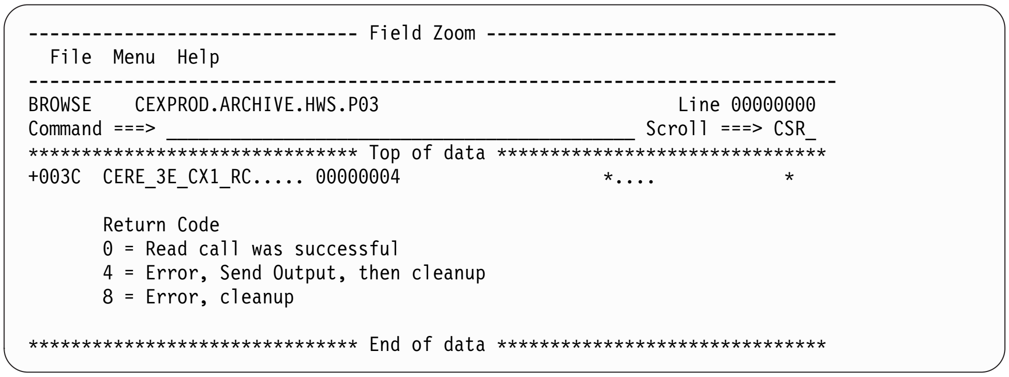


```
  File  Menu  Format  Help
-------------------------------------------------------------------------------
BROWSE     CEXPROD.ARCHIVE.HWS.P03               Record 00003156 Line 00000000
Command ===> ________________________________________________ Scroll ===> CSR_
Form    ===>          +     Use Form in Filter               Format ===> STD_
******************************** Top of data ********************************
+0004  Code... 003E  Message Exit returned from READ
+012E  STCK... CB873F4A2A1B5204      LSN.... 0000000000000C54
       Date... 2012-12-21 Friday     Time... 09.31.25.750368.215

+0000  CERE_3E_LL......... 013E
+0002  CERE_3E_ZZ......... 0000
+0004  CERE_3E_RECID...... A0  CERE_3E_EVTID...... 3E
+0006  CERE_3E_PFXLL...... 0014
+0008  CERE_3E_EFLAG...... 00  CERE_3E_VER#....... 23
+000A  CERE_3E_TASKID..... ID of task recording event
+000A  CERE_3E_COL#....... 01  CERE_3E_TKS#....... 04

:

+003B  CERE_3E_CX1_CON.... 98  CERE_3E_CX1_RC..... 00000004
+0040  CERE_3E_CX1_RSN.... 00000048

+003C  CERE_3E_CR1_START.......... READ section

:

+006C  CERE_3E_IRM_START.......... IRM section
+006C  CERE_3E_IRM_LL..... 0014
+006E  CERE_3E_IRM_APAR... 0003
+0070  CERE_3E_IRM_CLL.... 0058
+0072  CERE_3E_IRM_F5..... 00  CERE_3E_IRM_TIMER.......... 00
+0074  CERE_3E_IRM_SOCT... 00  CERE_3E_IRM_ES..... 00
+0076  CERE_3E_IRM_CLID... 'IPIPRT1W'
+007E  CERE_3E_IRM_ARCLVL......... 00

+0080  CERE_3E_SVT_START.......... SVT section
+0080  CERE_3E_SVT_LL..... 00AE
+0082  CERE_3E_SVT_APAR... 0007
+0084  CERE_3E_SVT_STATUS......... 80

:
```

*Figure 35. IMS Problem Investigator formatted display of event record*

## Field zoom

You can zoom on point-and-shoot fields, for example, to show details for a specific return code.



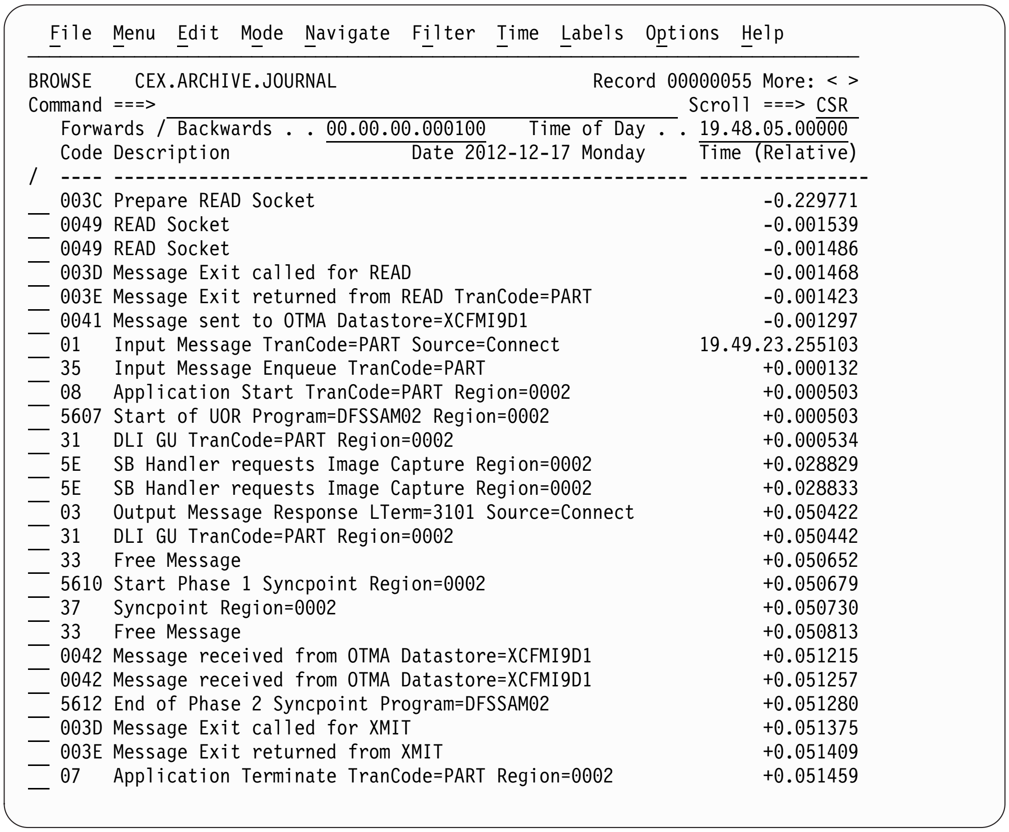
```
------------------------------ Field Zoom --------------------------------
  File  Menu  Help
--------------------------------------------------------------------------
BROWSE    CEXPROD.ARCHIVE.HWS.P03                          Line 00000000
Command ===> ____________________________________________ Scroll ===> CSR_
****************************** Top of data *******************************
+003C  CERE_3E_CX1_RC..... 00000004                  *....           *

       Return Code
       0 = Read call was successful
       4 = Error, Send Output, then cleanup
       8 = Error, cleanup

****************************** End of data *******************************
```

*Figure 36. IMS Problem Investigator zoom to display field details*

## Merge IMS and IMS Connect log files

IMS Problem Investigator can be used to merge log files to combine different aspects of IMS processing into a single view.

The following figure shows a transaction that arrived from TCP/IP, was routed by IMS Connect, and was then processed by IMS. This figure shows the time it took from when IMS Connect received the transaction to when IMS began processing the transaction. The overall time it took the IMS application to process the transaction can be seen at the bottom of the figure.



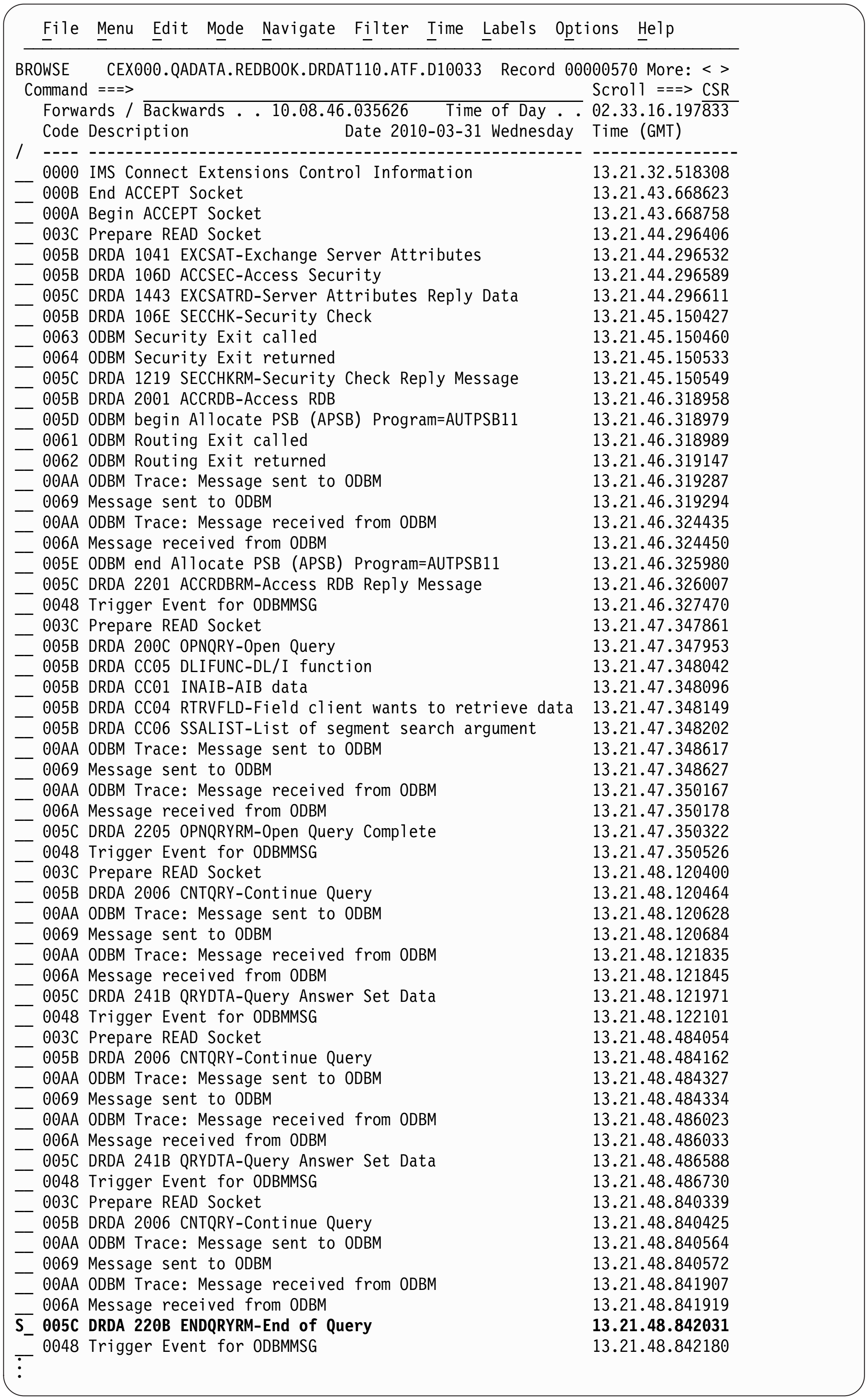
```
   File  Menu  Edit  Mode  Navigate  Filter  Time  Labels  Options  Help
──────────────────────────────────────────────────────────────────────────────
BROWSE    CEX.ARCHIVE.JOURNAL                        Record 00000055 More: < >
Command ===> ________________________________________________ Scroll ===> CSR
   Forwards / Backwards . . 00.00.00.000100    Time of Day . . 19.48.05.00000
   Code Description                 Date 2012-12-17 Monday     Time (Relative)
/  ---- ----------------------------------------------------- ----------------
__ 003C Prepare READ Socket                                          -0.229771
__ 0049 READ Socket                                                  -0.001539
__ 0049 READ Socket                                                  -0.001486
__ 003D Message Exit called for READ                                 -0.001468
__ 003E Message Exit returned from READ TranCode=PART                -0.001423
__ 0041 Message sent to OTMA Datastore=XCFMI9D1                      -0.001297
__ 01   Input Message TranCode=PART Source=Connect             19.49.23.255103
__ 35   Input Message Enqueue TranCode=PART                          +0.000132
__ 08   Application Start TranCode=PART Region=0002                  +0.000503
__ 5607 Start of UOR Program=DFSSAM02 Region=0002                    +0.000503
__ 31   DLI GU TranCode=PART Region=0002                             +0.000534
__ 5E   SB Handler requests Image Capture Region=0002                +0.028829
__ 5E   SB Handler requests Image Capture Region=0002                +0.028833
__ 03   Output Message Response LTerm=3101 Source=Connect            +0.050422
__ 31   DLI GU TranCode=PART Region=0002                             +0.050442
__ 33   Free Message                                                 +0.050652
__ 5610 Start Phase 1 Syncpoint Region=0002                          +0.050679
__ 37   Syncpoint Region=0002                                        +0.050730
__ 33   Free Message                                                 +0.050813
__ 0042 Message received from OTMA Datastore=XCFMI9D1                +0.051215
__ 0042 Message received from OTMA Datastore=XCFMI9D1                +0.051257
__ 5612 End of Phase 2 Syncpoint Program=DFSSAM02                    +0.051280
__ 003D Message Exit called for XMIT                                 +0.051375
__ 003E Message Exit returned from XMIT                              +0.051409
__ 07   Application Terminate TranCode=PART Region=0002              +0.051459
```

*Figure 37. A TCP/IP transaction: entering from IMS Connect and processed by IMS*

## Using IMS Problem Investigator in a Open Database environment

IMS Problem Investigator offers customers using DRDA clients in an ODBM environment a detailed record of the flow of events between the client, IMS Connect, ODBM, and IMS.

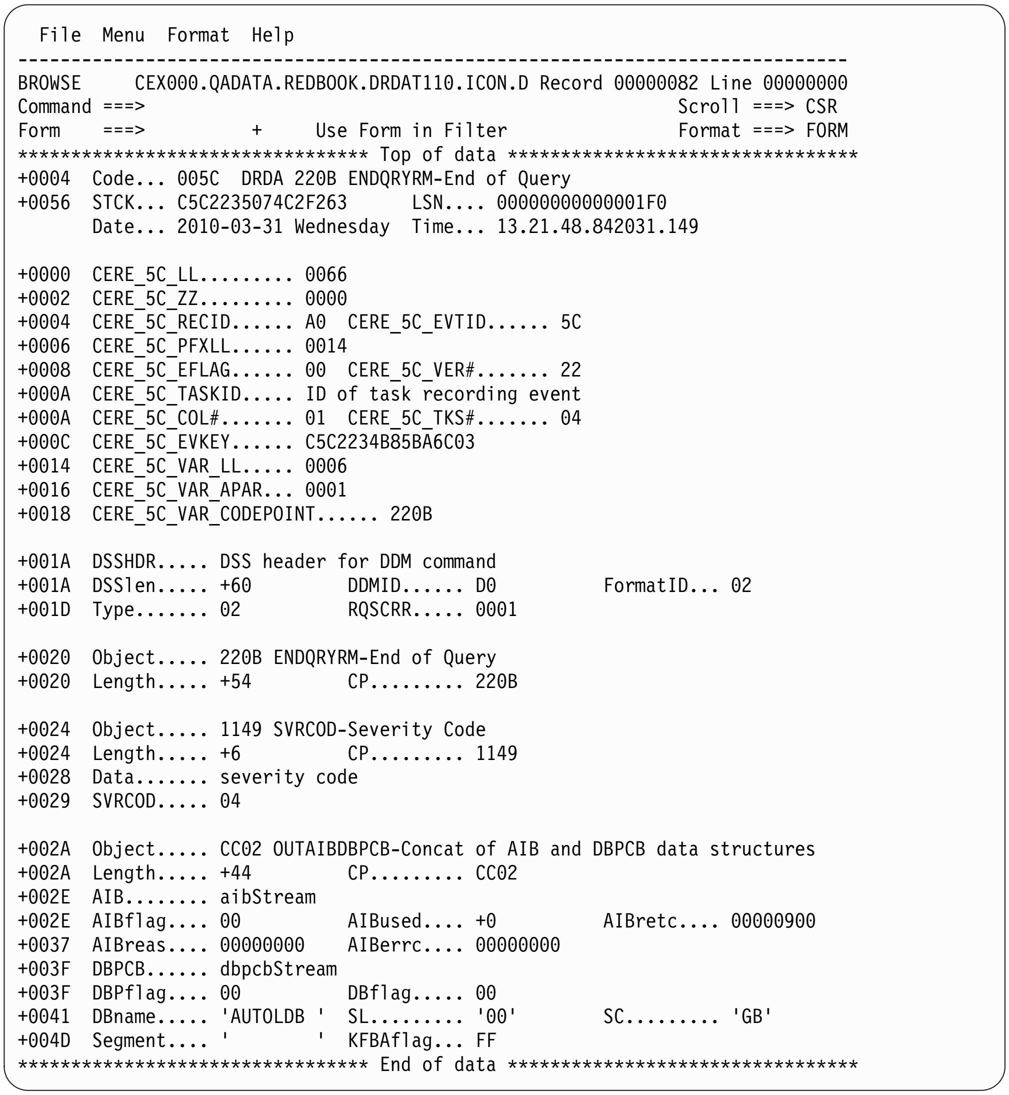
In this example the merge logs give a complete picture from the requests made from the DRDA client to IMS Connect, IMS Connect requests to ODBM, and the activity within IMS. (Records relating to READ Socket and WRITE Socket events have been omitted for clarity.)



```
   File  Menu  Edit  Mode  Navigate  Filter  Time  Labels  Options  Help
 ------------------------------------------------------------------------------
BROWSE    CEX000.QADATA.REDBOOK.DRDAT110.ATF.D10033  Record 00000570 More: < >
 Command ===> ________________________________________________ Scroll ===> CSR
   Forwards / Backwards . . 10.08.46.035626    Time of Day . . 02.33.16.197833
   Code Description                 Date 2010-03-31 Wednesday  Time (GMT)
/  ---- ------------------------------------------------------ ----------------
__ 0000 IMS Connect Extensions Control Information             13.21.32.518308
__ 000B End ACCEPT Socket                                      13.21.43.668623
__ 000A Begin ACCEPT Socket                                    13.21.43.668758
__ 003C Prepare READ Socket                                    13.21.44.296406
__ 005B DRDA 1041 EXCSAT-Exchange Server Attributes            13.21.44.296532
__ 005B DRDA 106D ACCSEC-Access Security                       13.21.44.296589
__ 005C DRDA 1443 EXCSATRD-Server Attributes Reply Data        13.21.44.296611
__ 005B DRDA 106E SECCHK-Security Check                        13.21.45.150427
__ 0063 ODBM Security Exit called                              13.21.45.150460
__ 0064 ODBM Security Exit returned                            13.21.45.150533
__ 005C DRDA 1219 SECCHKRM-Security Check Reply Message        13.21.45.150549
__ 005B DRDA 2001 ACCRDB-Access RDB                            13.21.46.318958
__ 005D ODBM begin Allocate PSB (APSB) Program=AUTPSB11        13.21.46.318979
__ 0061 ODBM Routing Exit called                               13.21.46.318989
__ 0062 ODBM Routing Exit returned                             13.21.46.319147
__ 00AA ODBM Trace: Message sent to ODBM                       13.21.46.319287
__ 0069 Message sent to ODBM                                   13.21.46.319294
__ 00AA ODBM Trace: Message received from ODBM                 13.21.46.324435
__ 006A Message received from ODBM                             13.21.46.324450
__ 005E ODBM end Allocate PSB (APSB) Program=AUTPSB11          13.21.46.325980
__ 005C DRDA 2201 ACCRDBRM-Access RDB Reply Message            13.21.46.326007
__ 0048 Trigger Event for ODBMMSG                              13.21.46.327470
__ 003C Prepare READ Socket                                    13.21.47.347861
__ 005B DRDA 200C OPNQRY-Open Query                            13.21.47.347953
__ 005B DRDA CC05 DLIFUNC-DL/I function                        13.21.47.348042
__ 005B DRDA CC01 INAIB-AIB data                               13.21.47.348096
__ 005B DRDA CC04 RTRVFLD-Field client wants to retrieve data  13.21.47.348149
__ 005B DRDA CC06 SSALIST-List of segment search argument      13.21.47.348202
__ 00AA ODBM Trace: Message sent to ODBM                       13.21.47.348617
__ 0069 Message sent to ODBM                                   13.21.47.348627
__ 00AA ODBM Trace: Message received from ODBM                 13.21.47.350167
__ 006A Message received from ODBM                             13.21.47.350178
__ 005C DRDA 2205 OPNQRYRM-Open Query Complete                 13.21.47.350322
__ 0048 Trigger Event for ODBMMSG                              13.21.47.350526
__ 003C Prepare READ Socket                                    13.21.48.120400
__ 005B DRDA 2006 CNTQRY-Continue Query                        13.21.48.120464
__ 00AA ODBM Trace: Message sent to ODBM                       13.21.48.120628
__ 0069 Message sent to ODBM                                   13.21.48.120684
__ 00AA ODBM Trace: Message received from ODBM                 13.21.48.121835
__ 006A Message received from ODBM                             13.21.48.121845
__ 005C DRDA 241B QRYDTA-Query Answer Set Data                 13.21.48.121971
__ 0048 Trigger Event for ODBMMSG                              13.21.48.122101
__ 003C Prepare READ Socket                                    13.21.48.484054
__ 005B DRDA 2006 CNTQRY-Continue Query                        13.21.48.484162
__ 00AA ODBM Trace: Message sent to ODBM                       13.21.48.484327
__ 0069 Message sent to ODBM                                   13.21.48.484334
__ 00AA ODBM Trace: Message received from ODBM                 13.21.48.486023
__ 006A Message received from ODBM                             13.21.48.486033
__ 005C DRDA 241B QRYDTA-Query Answer Set Data                 13.21.48.486588
__ 0048 Trigger Event for ODBMMSG                              13.21.48.486730
__ 003C Prepare READ Socket                                    13.21.48.840339
__ 005B DRDA 2006 CNTQRY-Continue Query                        13.21.48.840425
__ 00AA ODBM Trace: Message sent to ODBM                       13.21.48.840564
__ 0069 Message sent to ODBM                                   13.21.48.840572
__ 00AA ODBM Trace: Message received from ODBM                 13.21.48.841907
__ 006A Message received from ODBM                             13.21.48.841919
S_ 005C DRDA 220B ENDQRYRM-End of Query                        13.21.48.842031
__ 0048 Trigger Event for ODBMMSG                              13.21.48.842180
:
```

*Figure 38. Sequence of records relating to a DRDA request*

Here we have the details of a selected DRDA response. You can see structure of the reply, including all of its objects (parameters).



```
  File  Menu  Format  Help
------------------------------------------------------------------------------
BROWSE     CEX000.QADATA.REDBOOK.DRDAT110.ICON.D Record 00000082 Line 00000000
Command ===>                                                  Scroll ===> CSR
Form    ===>          +     Use Form in Filter                Format ===> FORM
********************************* Top of data ********************************
+0004  Code... 005C  DRDA 220B ENDQRYRM-End of Query
+0056  STCK... C5C2235074C2F263      LSN.... 00000000000001F0
       Date... 2010-03-31 Wednesday  Time... 13.21.48.842031.149

+0000  CERE_5C_LL......... 0066
+0002  CERE_5C_ZZ......... 0000
+0004  CERE_5C_RECID...... A0  CERE_5C_EVTID...... 5C
+0006  CERE_5C_PFXLL...... 0014
+0008  CERE_5C_EFLAG...... 00  CERE_5C_VER#....... 22
+000A  CERE_5C_TASKID..... ID of task recording event
+000A  CERE_5C_COL#....... 01  CERE_5C_TKS#....... 04
+000C  CERE_5C_EVKEY...... C5C2234B85BA6C03
+0014  CERE_5C_VAR_LL..... 0006
+0016  CERE_5C_VAR_APAR... 0001
+0018  CERE_5C_VAR_CODEPOINT...... 220B

+001A  DSSHDR..... DSS header for DDM command
+001A  DSSlen..... +60        DDMID...... D0          FormatID... 02
+001D  Type....... 02         RQSCRR..... 0001

+0020  Object..... 220B ENDQRYRM-End of Query
+0020  Length..... +54        CP......... 220B

+0024  Object..... 1149 SVRCOD-Severity Code
+0024  Length..... +6         CP......... 1149
+0028  Data....... severity code
+0029  SVRCOD..... 04

+002A  Object..... CC02 OUTAIBDBPCB-Concat of AIB and DBPCB data structures
+002A  Length..... +44        CP......... CC02
+002E  AIB........ aibStream
+002E  AIBflag.... 00         AIBused.... +0          AIBretc.... 00000900
+0037  AIBreas.... 00000000   AIBerrc.... 00000000
+003F  DBPCB...... dbpcbStream
+003F  DBPflag.... 00         DBflag..... 00
+0041  DBname..... 'AUTOLDB '  SL......... '00'       SC......... 'GB'
+004D  Segment.... '        '  KFBAflag... FF
********************************* End of data ********************************
```

## Reports and extracts

You can request reports and extracts for an IMS Connect system or a particular archive journal data set. To analyze an IMS Connect system in a specified period, Archive File Selection is available to automatically select the relevant archive data sets.

```
V2R3M0             IMS V12.1                IMS Problem Investigator - Log Report                                Page   2

+0004  Code... 0001  Connect Region Initialization
+001A  STCK... CB873CB5CFF94406      LSN.... 0000000000000003      Record... 3
       Date... 2013-06-18 Tuesday    Time... 09.19.53.147284.251

+0000  CERE_01_LL......... 002A                        CERE_01_ZZ......... 0000                        CERE_01_RECID...... A0
+0005  CERE_01_EVTID...... 01  CERE_01_PFXLL...... 0014                        CERE_01_EFLAG...... 00  CERE_01_VER#....... 23
+000A  CERE_01_TASKID..... ID of task recording event
+000A  CERE_01_COL#....... 01  CERE_01_TKS#....... 01
+000C  CERE_01_EVKEY...... 'EVNT    '                  CERE_01_VAR_LL..... 0006                        CERE_01_VAR_APAR... 0001
+0018  CERE_01_VAR_VVRR... 1210


+0004  Code... 0003  Support Task created
+0024  STCK... CB873CB5D5FB5A06      LSN.... 0000000000000004      Record... 4
       Date... 2013-06-18 Tuesday    Time... 09.19.53.171893.626

+0000  CERE_03_LL......... 0034                        CERE_03_ZZ......... 0000                        CERE_03_RECID...... A0
+0005  CERE_03_EVTID...... 03  CERE_03_PFXLL...... 0014                        CERE_03_EFLAG...... 00  CERE_03_VER#....... 23
+000A  CERE_03_TASKID..... ID of task recording event
+000A  CERE_03_COL#....... 01  CERE_03_TKS#....... 02
+000C  CERE_03_EVKEY...... 'EVNT    '                  CERE_03_VAR_LL..... 0008                        CERE_03_VAR_APAR... 0001
+0018  CERE_03_VAR_FLAG... Variable data Flag Field
+0018  CERE_03_VAR_FLAG1.......... 20                  CERE_03_VAR_FLAG2.......... 00
+001A  CERE_03_VAR_PORT... +0

+001C  CERE_03_BSE_LL..... Base section segment
+001C  CERE_03_BSE_LL..... 0008                        CERE_03_BSE_APAR... 0001
+0020  CERE_03_BSE_@TCB... 008B3BE0


+0004  Code... 0012  TMEMBER joins XCF Group
+004C  STCK... CB873CB6CCA99886      LSN.... 0000000000000005      Record... 5
       Date... 2013-06-18 Tuesday    Time... 09.19.54.182297.532

+0000  CERE_12_LL......... 005C                        CERE_12_ZZ......... 0000                        CERE_12_RECID...... A0
+0005  CERE_12_EVTID...... 12  CERE_12_PFXLL...... 0014                        CERE_12_EFLAG...... 00  CERE_12_VER#....... 23
+000A  CERE_12_TASKID..... ID of task recording event
+000A  CERE_12_COL#....... 01  CERE_12_TKS#....... 03
+000C  CERE_12_EVKEY...... 'EVNT    '

+0014  CERE_12_DSIB....... Datastore Information Block
+0014  CERE_12_DLL........ 0038                        CERE_12_BLKID...... 02  CERE_12_FLAG....... 20  CERE_12_VVRR....... 1210
+001A  CERE_12_APAR_SEQ... 0001                        CERE_12_DS_NAME.... 'HWSDEVAE        '
+002C  CERE_12_DATA_LEN... 00000000                    CERE_12_DATA_ADDR.......... 00000000
+0034  CERE_12_RC......... 00000000                    CERE_12_RSN........ 00000000
+003C  CERE_12_TPIPE_NAME......... '        '          CERE_12_SVTTOKN.... 0000000000000000


+0004  Code... 0012  TMEMBER joins XCF Group
+004C  STCK... CB873CB6D0CE2906      LSN.... 0000000000000006      Record... 6
       Date... 2013-06-18 Tuesday    Time... 09.19.54.199266.563

+0000  CERE_12_LL......... 005C                        CERE_12_ZZ......... 0000                        CERE_12_RECID...... A0
+0005  CERE_12_EVTID...... 12  CERE_12_PFXLL...... 0014                        CERE_12_EFLAG...... 00  CERE_12_VER#....... 23
+000A  CERE_12_TASKID..... ID of task recording event
+000A  CERE_12_COL#....... 01  CERE_12_TKS#....... 03
+000C  CERE_12_EVKEY...... 'EVNT    '
```

⋮

*Figure 39. IMS Problem Investigator formatted report of journal data set*

# Chapter 6. Tracing message flows

Tracing events are additional event records that are produced by IMS Connect Extensions when tracing is activated. You can query and set the status of tracing through the ISPF Operations dialog.

Tracing records are written to the Active Journal data set as event records. The Set Tracing dialog has several options for controlling when tracing events are produced and what data is included in the event records.

- For an OTMA workload you can specify a port and one of several other conditions to restrict when trace events are produced. For an Open Database workload you can only specify a port to restrict when trace events are produced.
- For an OTMA workload, you can set **Tracing Level** to 2 to write an additional section containing client application data to the tracing event records.

The current tracing status and tracing level are included in the X'00' event record.

You can also use the QUERY and TRACE host commands to query, start, and stop tracing. See the members CEXRXC11 and CEXRXC45 in the SCEXSAMP library for detailed examples.

**Recorder Trace:** You cannot query the status of the IMS Connect Recorder Trace through IMS Connect Extensions. However, you can start and stop Recorder tracing in the IMS Connect Extensions command shell or by using the **RS** and **RP** line actions in the Operations dialog.

**Related concepts**:

Chapter 4, "Event collection," on page 55
IMS Connect Extensions continuously collects events as incoming messages or DRDA requests are processed or when other changes occur, such as when IMS Connect starts up and shuts down or when a datastore goes offline.

## Starting the IMS Connect Extensions trace

You can start tracing for an active IMS Connect system through the Operations dialog or Status Monitor.

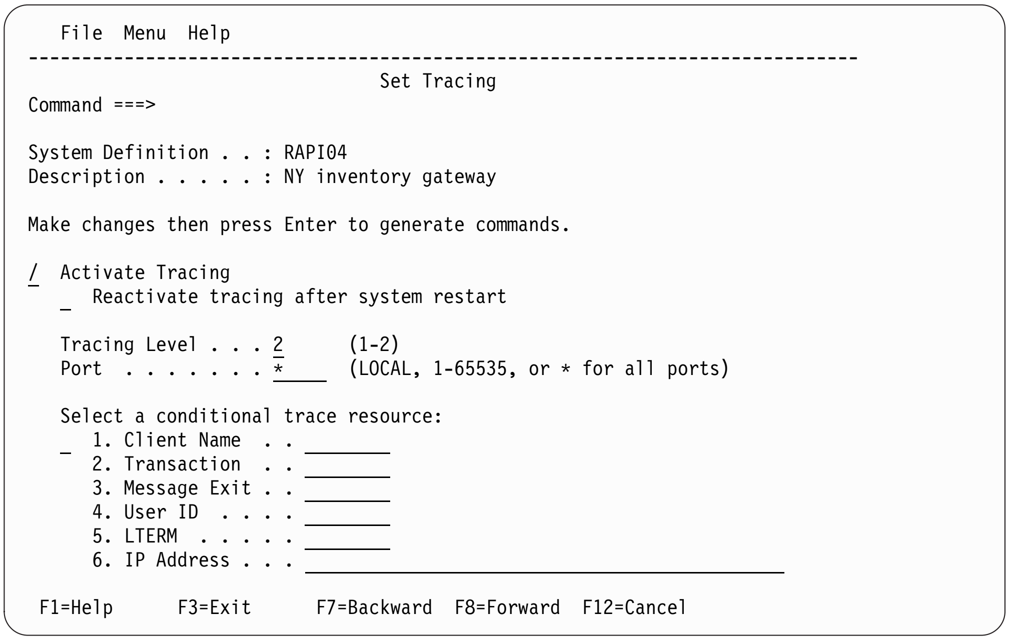
### About this task

Tracing continues until it is deactivated or until a system restart. You can enable tracing to continue after a shutdown and restart of IMS Connect by selecting the option **Reactivate tracing after system restart**. You can only make a trace persistent at the same time as you activate it.

### Procedure

1. From the primary option menu, select option **2 Operations**.
2. Set the View to option **2 Systems**.
3. Enter the characters **TR** next to an active system.
4. In the Set Tracing dialog, select the **Activate Tracing** option. If required, set additional tracing conditions. IMS Connect Extensions will only record messages that match the tracing conditions.

**Note:** Conditional tracing is not applicable when tracing ODBM activity. To trace ODBM activity, simply specify the port.



```
   File  Menu  Help
------------------------------------------------------------------------------
                                 Set Tracing
Command ===>

System Definition . . : RAPI04
Description . . . . . : NY inventory gateway

Make changes then press Enter to generate commands.

/  Activate Tracing
   _  Reactivate tracing after system restart

   Tracing Level . . . 2      (1-2)
   Port  . . . . . . . *____  (LOCAL, 1-65535, or * for all ports)

   Select a conditional trace resource:
   _  1. Client Name  . . ________
      2. Transaction  . . ________
      3. Message Exit . . ________
      4. User ID  . . . . ________
      5. LTERM  . . . . . ________
      6. IP Address . . . _____________________________________________

 F1=Help      F3=Exit      F7=Backward  F8=Forward  F12=Cancel
```

*Figure 40. Set Tracing command panel*

5. Press **Enter**. The Command Processor panel appears with the command set.
6. Type / (slash) next to all commands and then press **Enter**. Confirm that the Response column shows all zeros (0000), indicating that the command was successful.
7. Press the Exit function key (F3) twice to return to the Systems view.

**Related reference**:

"TRACE" on page 264
Use the **TRACE** host command to start or stop tracing. You can specify filter conditions to control which additional event records are written to the Active Journal data set.

## Querying the status of tracing

You can query whether tracing is active for an IMS Connect system through the Operations dialog or Status Monitor.

### Procedure

1. From the primary option menu, select option **2 Operations**.
2. Set the View to option **2 Systems**.
3. Enter the characters **TR** next to an active system. The Set Tracing dialog appears.

   If tracing is active the **Activate Tracing** option appears selected. The tracing criteria, level, port, and other conditions for this trace also appear on the panel.

**Related reference**:

"QUERY" on page 254
The **QUERY** command reports on statistics and settings for several IMS Connect Extensions resources. These include: the current tracing options; the routing status and the number of pending messages for one or more datastores; the ACEE cache

status, the active routing plan; and active session details.

## Stopping the IMS Connect Extensions trace

You can stop tracing for an active IMS Connect system through the Operations dialog or Status Monitor.

### Procedure

1. From the primary option menu, select option **2 Operations**.
2. Set the View to option **2 Systems**.
3. Enter the characters **TR** next to an active system.
4. In the Set Tracing dialog, deselect **Activate Tracing**.
5. Press Enter.
6. In the Command Processor panel, type / (slash) next to SET TRACING INACTIVE and then press Enter. Confirm that the Response column shows all zeros (0000), indicating that the command was successful.

**Related reference**:

"TRACE" on page 264
Use the **TRACE** host command to start or stop tracing. You can specify filter conditions to control which additional event records are written to the Active Journal data set.

# Part 3. Managing IMS Connect systems

These topics describe the Operations dialog, which you can use to monitor and interact with IMS Connect systems, and the CEX host command environment for REXX, which you can use to automate operations tasks by issuing IMS Connect Extensions commands from REXX execs.

**Topics:**

# Chapter 7. Overview of the operations dialog

The Operations dialog shows you all IMS Connect systems in the definitions repository, their status, and information about the IMS Connect Extensions components that they are running.

Potentially, you can view all IMS Connect systems that can be accessed via the TCP/IP network (or local adapter) through this dialog. The following figure shows the organization of the dialog and selected line actions:

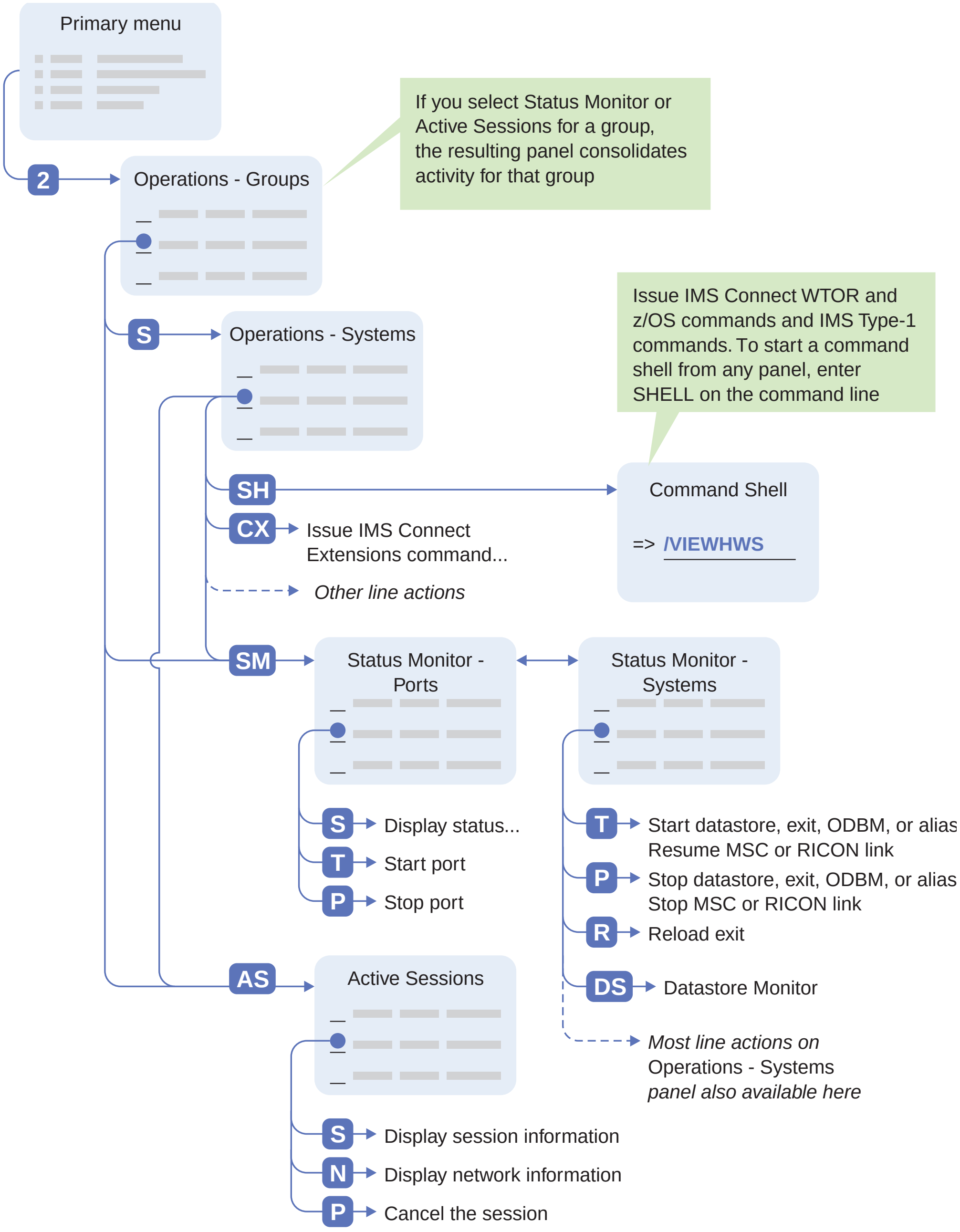

Primary menu
2
Operations - Groups
If you select Status Monitor or Active Sessions for a group, the resulting panel consolidates activity for that group
S
Operations - Systems
Issue IMS Connect WTOR and z/OS commands and IMS Type-1 commands. To start a command shell from any panel, enter SHELL on the command line
SH
Command Shell
=> /VIEWHWS
CX
Issue IMS Connect Extensions command...
Other line actions
SM
Status Monitor - Ports
Status Monitor - Systems
S
Display status...
T
Start port
P
Stop port
T
Start datastore, exit, ODBM, or alias Resume MSC or RICON link
P
Stop datastore, exit, ODBM, or alias Stop MSC or RICON link
R
Reload exit
DS
Datastore Monitor
Most line actions on Operations - Systems panel also available here
AS
Active Sessions
S
Display session information
N
Display network information
P
Cancel the session

*Figure 41. IMS Connect Extensions Operations dialog navigation*

You can use the Operations dialog to view real-time information about your IMS Connect systems and perform various operations on those systems. An operation can be an IMS Connect WTOR command or IMS Connect z/OS command, an IMS Type-1 command, or an IMS Connect Extensions enhancement. For example, you can stop a datastore by entering P next to it (STOPDS in IMS Connect command syntax), or you can initialize an IMS Connect Extensions enhanced trace for a system.

The Operations dialog itself shows you which IMS Connect systems are active and the IMS Connect Extensions version they are running, as well as the port and host name of the IMS Connect Extensions console listener. You can view your systems in groups (for example, as a sysplex) or individually.

For more information on the system management operations you can perform from this dialog see:

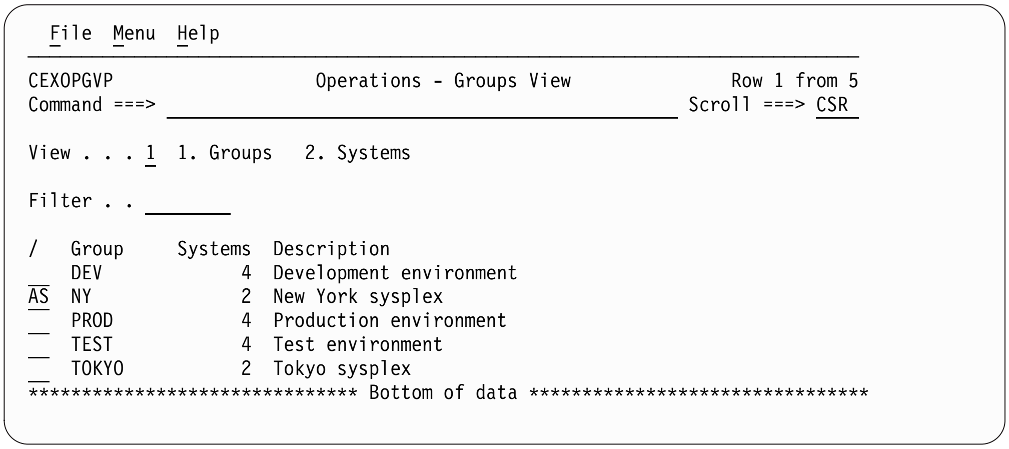
- Chapter 8, “Issuing commands using the Operations dialog,” on page 97
- Chapter 9, “Monitoring and controlling systems and their components,” on page 101
- Chapter 10, “Monitoring and controlling active sessions,” on page 109
- “Viewing message logs” on page 95

## Operations - Groups view

The Operations - Groups dialog presents your systems in groups. Grouping systems enables you to logically organize them, and then perform operations according to that organization.

A group can represent any arrangement. For example, a group can represent a sysplex, an environment (development, test, and production), or a function (payroll, HR, or inventory). You can place the same system in more than one group. Once in groups, you can view systems depending on the problem domain you are trying to address.

The following figure shows the Groups view with an example configuration.



```
  File  Menu  Help
──────────────────────────────────────────────────────────────────────────────
CEXOPGVP                    Operations - Groups View               Row 1 from 5
Command ===> ________________________________________________ Scroll ===> CSR

View . . . 1  1. Groups   2. Systems

Filter . . ________

/   Group     Systems  Description
__  DEV             4  Development environment
AS  NY              2  New York sysplex
__  PROD            4  Production environment
__  TEST            4  Test environment
__  TOKYO           2  Tokyo sysplex
******************************* Bottom of data ********************************
```

*Figure 42. Example of groups view in the Operations dialog*

You can perform the following line actions from this panel:

**S** View the systems in the group.

**AS** View active sessions occurring within the group (such as a sysplex) and perform operations on those sessions (such as stopping them).

**SM** View the components of all the systems in the group and their activity with the Status Monitor. Use this option to manage exits and datastores for an entire group.

To define groups, see "Defining system groups" on page 219.

### Grouping systems improves the performance of the ISPF dialog

The Operations dialog polls each system you are viewing to determine whether that system is active or inactive. The dialog waits for a response until a timeout value has elapsed. If there are many inactive systems, or if there is high-network latency, the polling operation may degrade the responsiveness of the dialog. To overcome this issue, put the IMS Connect systems in groups, and then always view systems via their group. This minimizes the number of systems that are polled at one time.

## Operations - Systems view

The Operations - Systems view provides summary information about the IMS Connect systems and their IMS Connect Extensions components. From the Systems view, you can perform most operations against systems.

You can access the Systems view by selecting a system in the Group view, or view all systems by selecting **2 Systems** in the Operations dialog. When you have many systems, it is recommended that you always access a subset of those systems by selecting a group. This improves the performance of the ISPF dialog and makes the systems easier to manage.

**Note:** When you first enter Systems view in the Operations dialog, you will be prompted for your password for those active IMS Connect systems that also have command access control enabled. Systems with PassTicket validation active do not require a password.

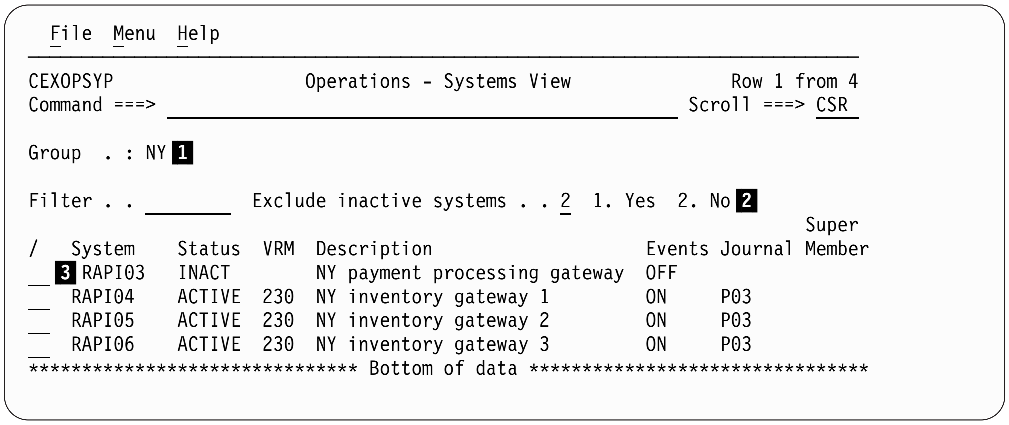
- Take care to enter the password correctly as it is a non-display field.
- Passwords might be case sensitive.

If this user ID has the same password for all IMS Connect systems on which command access control is enabled, then you can select "Use this password for all IMS Connect systems" to bypass any further prompting for passwords.

If you do not want to enter commands against a system with active access control, press the Cancel function key (F12) to close the password prompt. You can then proceed to monitor all systems and control those systems for which command access control is inactive.

### Options and actions

The following figure shows the systems view with an example configuration.



```
  File  Menu  Help
 ──────────────────────────────────────────────────────────────────────────────
 CEXOPSYP                  Operations - Systems View                Row 1 from 4
 Command ===> ________________________________________________ Scroll ===> CSR

 Group  . : NY 1

 Filter . . ________  Exclude inactive systems . . 2  1. Yes  2. No 2
                                                                         Super
 /   System    Status  VRM  Description                    Events Journal Member
 __ 3 RAPI03   INACT        NY payment processing gateway  OFF
 __  RAPI04    ACTIVE  230  NY inventory gateway 1         ON     P03
 __  RAPI05    ACTIVE  230  NY inventory gateway 2         ON     P03
 __  RAPI06    ACTIVE  230  NY inventory gateway 3         ON     P03
 ******************************* Bottom of data ********************************
```

*Figure 43. Example of systems view in the operations dialog*

**1** If you enter Systems view by selecting a group, that group is displayed at the top of the panel.

**2** You can further limit which systems appear by specifying a filter, or by excluding inactive systems. A filter selects systems with a name matching the string or substring you enter. You can specify multicharacter wildcards with asterisks (*) and single character wildcards with percent signs (%). For example:

`*` Matches all definitions.

`AB*` Matches all definitions with names starting with AB. For example, ABCDE.

`A%B*` Matches ACBDD but not ACDBD.

**3** You can view information for each system, such as whether it is active, the IMS Connect Extensions version it is running, and the status of event collection. You can also perform these tasks:

- Sort systems by column by tabbing to the line under the column and pressing Enter.
- View additional columns of information by pressing the Right function key (F11).
- Perform many operations such as stopping systems, starting traces, and managing exits, datastores, or sessions through line actions.

**SM or S**
Use the Status Monitor to view the components of the system and their activity. Use this option to manage exits and datastores.

**AS** View active sessions on the system and perform operations such as stopping sessions.

**DS** Use the Datastore Monitor to view detailed information about datastores and start, stop, drain, or resume selected datastores.

**CX** Issue IMS Connect Extensions commands.

**L** View message log.

**SH** Issue IMS Connect/IMS Type-1 commands from the command shell.

**PU** Manage Publisher clients.

**TR** Set trace parameters; start or stop tracing.

**RS** Start Recorder Trace.

**RP** Stop Recorder Trace.

**J** Switch the current active journal to the next journal.

**P** Stop system.

**F** Stop system (with Force).

**X** Exclude (hide) this system. Enter **REFRESH** on the command line to redisplay all excluded systems.

**AC** Display ACEE Cache statistics, including cache space usage and statistics on different types of cache services requests.

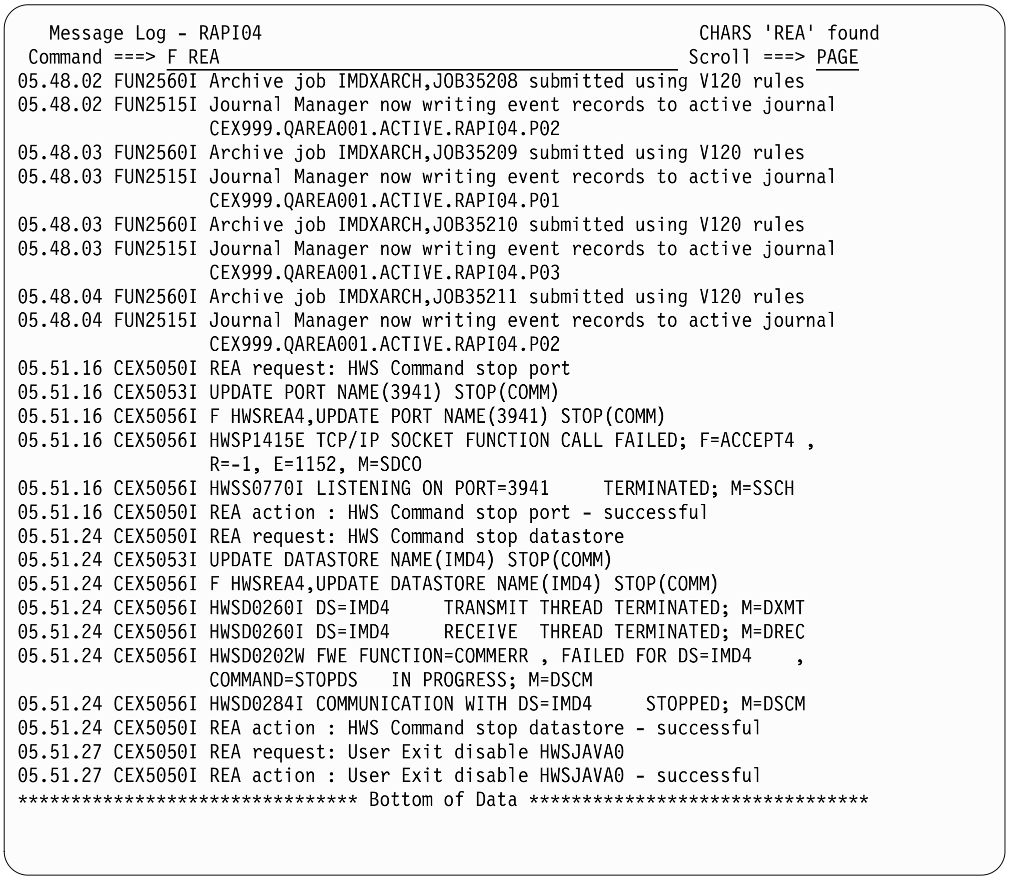
## Viewing message logs

Message logs show you information about IMS Connect and IMS Connect Extensions activity.

You can view IMS Connect message logs from all operations panels except the Operations - Groups View panel by entering an **L** next to that system or one of its components, or by using the **LOG** primary command. For example:

- The status of archiving.
- A list of recent commands and their outcome.
- Any errors that may have occurred.

Figure 44 on page 96 shows an example message log. As shown in this example, you can use standard ISPF Browse commands. For example, enter **FIND** *string* to find a string within the log.



```
   Message Log - RAPI04                                          CHARS 'REA' found
 Command ===> F REA                                             Scroll ===> PAGE
05.48.02 FUN2560I Archive job IMDXARCH,JOB35208 submitted using V120 rules
05.48.02 FUN2515I Journal Manager now writing event records to active journal
                  CEX999.QAREA001.ACTIVE.RAPI04.P02
05.48.03 FUN2560I Archive job IMDXARCH,JOB35209 submitted using V120 rules
05.48.03 FUN2515I Journal Manager now writing event records to active journal
                  CEX999.QAREA001.ACTIVE.RAPI04.P01
05.48.03 FUN2560I Archive job IMDXARCH,JOB35210 submitted using V120 rules
05.48.03 FUN2515I Journal Manager now writing event records to active journal
                  CEX999.QAREA001.ACTIVE.RAPI04.P03
05.48.04 FUN2560I Archive job IMDXARCH,JOB35211 submitted using V120 rules
05.48.04 FUN2515I Journal Manager now writing event records to active journal
                  CEX999.QAREA001.ACTIVE.RAPI04.P02
05.51.16 CEX5050I REA request: HWS Command stop port
05.51.16 CEX5053I UPDATE PORT NAME(3941) STOP(COMM)
05.51.16 CEX5056I F HWSREA4,UPDATE PORT NAME(3941) STOP(COMM)
05.51.16 CEX5056I HWSP1415E TCP/IP SOCKET FUNCTION CALL FAILED; F=ACCEPT4 ,
                  R=-1, E=1152, M=SDCO
05.51.16 CEX5056I HWSS0770I LISTENING ON PORT=3941     TERMINATED; M=SSCH
05.51.16 CEX5050I REA action : HWS Command stop port - successful
05.51.24 CEX5050I REA request: HWS Command stop datastore
05.51.24 CEX5053I UPDATE DATASTORE NAME(IMD4) STOP(COMM)
05.51.24 CEX5056I F HWSREA4,UPDATE DATASTORE NAME(IMD4) STOP(COMM)
05.51.24 CEX5056I HWSD0260I DS=IMD4     TRANSMIT THREAD TERMINATED; M=DXMT
05.51.24 CEX5056I HWSD0260I DS=IMD4     RECEIVE  THREAD TERMINATED; M=DREC
05.51.24 CEX5056I HWSD0202W FWE FUNCTION=COMMERR , FAILED FOR DS=IMD4    ,
                  COMMAND=STOPDS   IN PROGRESS; M=DSCM
05.51.24 CEX5056I HWSD0284I COMMUNICATION WITH DS=IMD4     STOPPED; M=DSCM
05.51.24 CEX5050I REA action : HWS Command stop datastore - successful
05.51.27 CEX5050I REA request: User Exit disable HWSJAVA0
05.51.27 CEX5050I REA action : User Exit disable HWSJAVA0 - successful
******************************** Bottom of Data ********************************
```

*Figure 44. IMS Connect message log*

# Chapter 8. Issuing commands using the Operations dialog

You can issue IMS Connect WTOR commands and IMS Connect z/OS commands and IMS Type-1 commands from IMS Connect Extensions. You can enter commands as line actions in the Operations dialog, or through a command shell.

This enables you to:

- Control multiple IMS Connect systems from a single point.
- Define granular access control to IMS Connect commands.
- View the real-time, enterprise-wide, impact of commands on sessions and message processing activity.

As a general rule, IMS Type-1 commands, IMS Connect WTOR commands and IMS Connect z/OS commands that return an output are only available from the command shell. Commands that perform an action, for example, stopping a datastore, are available both as line actions and from the command shell.

The advantages of issuing commands as line actions are that:

- It is intuitive and easy to use: you do not need to be familiar with the syntax of IMS Connect commands.
- You can see the results of your actions in the dialog. For example, if you stop a datastore, you can see its status change and the number of messages it is processing drop to zero.

The command shell provides a command-line interface for entering commands in IMS Connect and IMS Connect z/OS (MVS) command form. The advantages of this mode are that:

- You can view the output of **QUERY** commands.
- You can use familiar IMS Connect or IMS Type-1 command syntax.

You can enable or disable command shell access for a given IMS Connect system. You can also control access to specific commands using a SAF-compliant external security manager such as RACF.

## Issuing IMS Connect and IMS Connect Extensions commands as line actions

You can issue IMS Connect WTOR commands and IMS Connect z/OS commands and IMS Connect Extensions commands as line actions in the Operations dialog.

### About this task

You can use IMS Connect Extensions commands to perform the following operations:

- Start and stop resource traces.
- Reload and add user exits dynamically.
- Manually switch the active journal for a system.
- Clear an entry from the ACEE cache.
- Refresh SAF class rules.

- Modify IMS Connect Extensions settings dynamically with SET and **REFRESH** commands.

All of these commands are available for the IMS Connect Extensions command shell dialog. To access the command shell, type the **CX** line action next to a system in the Operations dialog. Some of these commands can also be issued as separate line actions. Enter / (slash) next to a system or resource in the Operations dialog to list the available line actions.

**Note:** IMS Connect Type-2 commands are not supported by IMS Connect Extensions.

### Procedure

To issue IMS Connect WTOR commands and IMS Connect z/OS commands and IMS Connect Extensions commands as line actions:

1. From the primary menu, select option 2 **Operations**.
2. Enter a line command next to an active system.

## Issuing IMS Connect and IMS Type-1 commands from the command shell

Users who have the appropriate level of authority can use the command shell to issue IMS Connect WTOR commands and IMS Connect z/OS commands and IMS Type-1 commands from IMS Connect Extensions. The shell is particularly useful for displaying the output of QUERY commands.

### About this task

The current context (that is, the target IMS Connect system or datastore) is shown at the top of the panel. Entering / (slash) with no command displays the command shell where you can specify the system or datastore and select from a list of recently issued commands.

The command shell initially defaults to the IMS Connect system, so to enter an IMS Type-1 command you must use the Prompt function key (F4) to specify the target datastore.

**Tips:**

1. Press F1 to view all available IMS Connect WTOR commands and IMS Connect z/OS commands.

   The available IMS Type-1 commands are listed in the *IMS Commands* reference documentation.
2. Enter / (slash) to view the command recall dialog, which lists previously entered commands. If you enter an IMS Connect command from this dialog, do not precede it with a forward slash.

   You can clear the display of previous commands and responses by entering CLEAR on the command line.
3. If the command returns an output it appears on the screen in an ISPF Browse session, as shown in Figure 46 on page 100. This means you can use standard ISPF Browse commands. For example, enter **FIND** *string* to search the output for a string.

## Procedure

To issue a command from the shell:

1. From the IMS Connect Extensions ISPF dialog, start the shell using either of these methods:
   - Enter the primary command **SHELL** from any panel (you may need to then select the system for which you want to use the shell).
   - Type **SH** next to a system in the operations dialog.
2. Enter the command, preceded by a / (slash). IMS Connect commands can be entered in either IMS Connect command form or the IMS Connect z/OS (MVS) command form. For example:

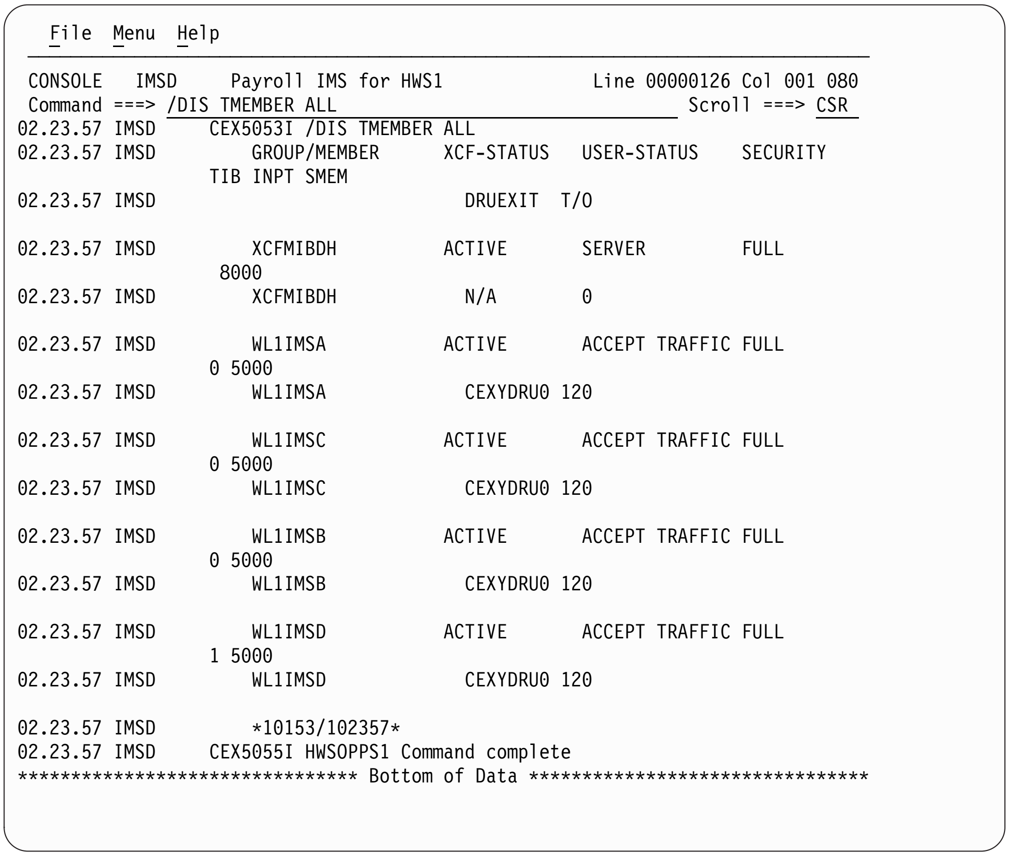
   ```
   /VIEWPORT 3702
   ```

   or

   ```
   /QUERY PORT NAME(3702)
   ```

## Results

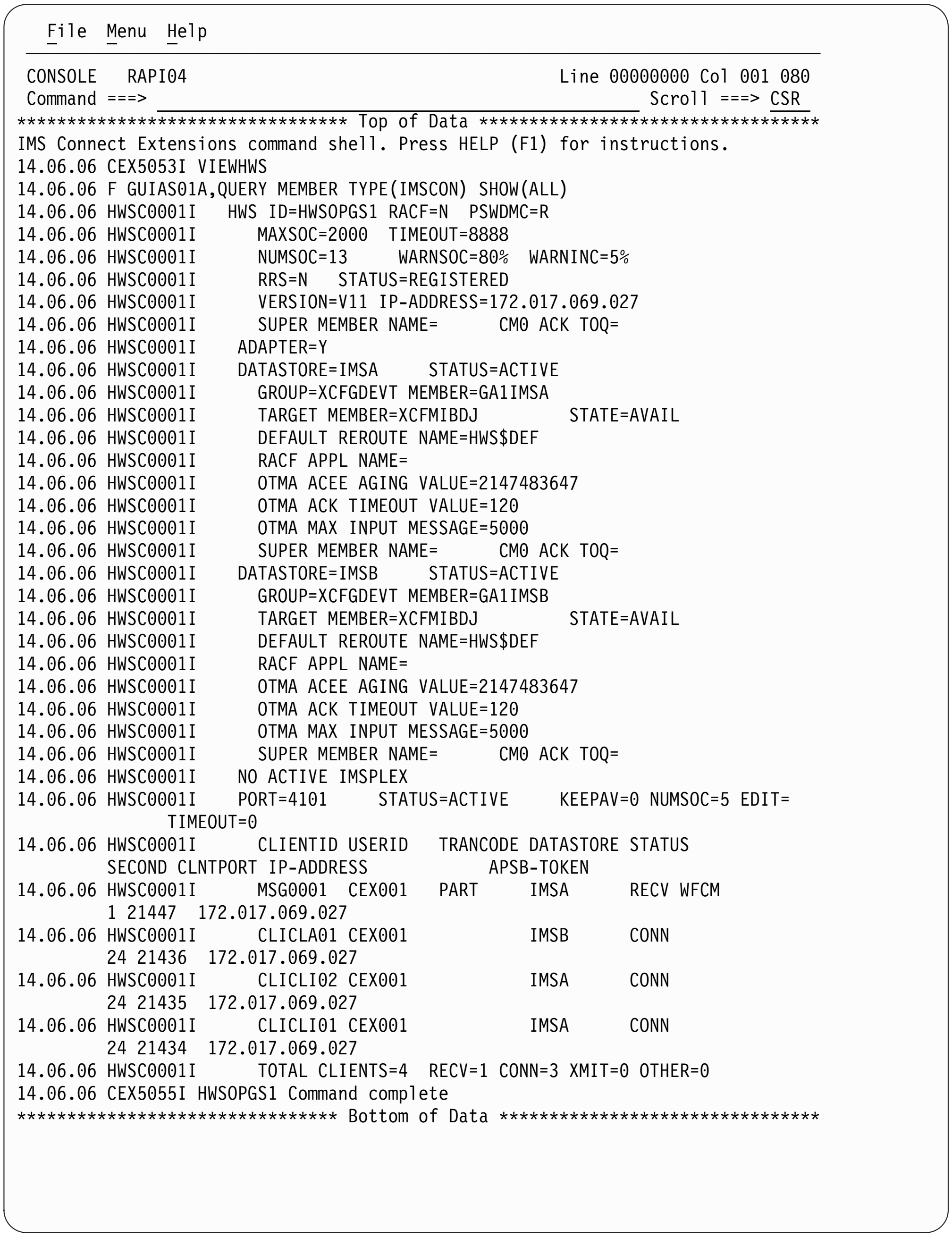
The following figure shows an example of issuing an IMS Type-1 command.



```
 CONSOLE   IMSD     Payroll IMS for HWS1             Line 00000126 Col 001 080
 Command ===> /DIS TMEMBER ALL                                Scroll ===> CSR
02.23.57 IMSD     CEX5053I /DIS TMEMBER ALL
02.23.57 IMSD         GROUP/MEMBER      XCF-STATUS   USER-STATUS    SECURITY
                  TIB INPT SMEM
02.23.57 IMSD                             DRUEXIT  T/O

02.23.57 IMSD         XCFMIBDH          ACTIVE       SERVER         FULL
                   8000
02.23.57 IMSD         XCFMIBDH            N/A        0

02.23.57 IMSD         WL1IMSA           ACTIVE       ACCEPT TRAFFIC FULL
                  0 5000
02.23.57 IMSD         WL1IMSA             CEXYDRU0 120

02.23.57 IMSD         WL1IMSC           ACTIVE       ACCEPT TRAFFIC FULL
                  0 5000
02.23.57 IMSD         WL1IMSC             CEXYDRU0 120

02.23.57 IMSD         WL1IMSB           ACTIVE       ACCEPT TRAFFIC FULL
                  0 5000
02.23.57 IMSD         WL1IMSB             CEXYDRU0 120

02.23.57 IMSD         WL1IMSD           ACTIVE       ACCEPT TRAFFIC FULL
                  1 5000
02.23.57 IMSD         WL1IMSD             CEXYDRU0 120

02.23.57 IMSD         *10153/102357*
02.23.57 IMSD     CEX5055I HWSOPPS1 Command complete
******************************* Bottom of Data ********************************
```

*Figure 45. Example: Issuing an IMS Type-1 command from the command shell*

The following figure shows an example of issuing an IMS Connect command.



```
 CONSOLE   RAPI04                                    Line 00000000 Col 001 080
 Command ===>                                                  Scroll ===> CSR
******************************* Top of Data ********************************
IMS Connect Extensions command shell. Press HELP (F1) for instructions.
14.06.06 CEX5053I VIEWHWS
14.06.06 F GUIAS01A,QUERY MEMBER TYPE(IMSCON) SHOW(ALL)
14.06.06 HWSC0001I   HWS ID=HWSOPGS1 RACF=N  PSWDMC=R
14.06.06 HWSC0001I      MAXSOC=2000  TIMEOUT=8888
14.06.06 HWSC0001I      NUMSOC=13     WARNSOC=80%  WARNINC=5%
14.06.06 HWSC0001I      RRS=N   STATUS=REGISTERED
14.06.06 HWSC0001I      VERSION=V11 IP-ADDRESS=172.017.069.027
14.06.06 HWSC0001I      SUPER MEMBER NAME=      CM0 ACK TOQ=
14.06.06 HWSC0001I    ADAPTER=Y
14.06.06 HWSC0001I    DATASTORE=IMSA     STATUS=ACTIVE
14.06.06 HWSC0001I      GROUP=XCFGDEVT MEMBER=GA1IMSA
14.06.06 HWSC0001I      TARGET MEMBER=XCFMIBDJ         STATE=AVAIL
14.06.06 HWSC0001I      DEFAULT REROUTE NAME=HWS$DEF
14.06.06 HWSC0001I      RACF APPL NAME=
14.06.06 HWSC0001I      OTMA ACEE AGING VALUE=2147483647
14.06.06 HWSC0001I      OTMA ACK TIMEOUT VALUE=120
14.06.06 HWSC0001I      OTMA MAX INPUT MESSAGE=5000
14.06.06 HWSC0001I      SUPER MEMBER NAME=      CM0 ACK TOQ=
14.06.06 HWSC0001I    DATASTORE=IMSB     STATUS=ACTIVE
14.06.06 HWSC0001I      GROUP=XCFGDEVT MEMBER=GA1IMSB
14.06.06 HWSC0001I      TARGET MEMBER=XCFMIBDJ         STATE=AVAIL
14.06.06 HWSC0001I      DEFAULT REROUTE NAME=HWS$DEF
14.06.06 HWSC0001I      RACF APPL NAME=
14.06.06 HWSC0001I      OTMA ACEE AGING VALUE=2147483647
14.06.06 HWSC0001I      OTMA ACK TIMEOUT VALUE=120
14.06.06 HWSC0001I      OTMA MAX INPUT MESSAGE=5000
14.06.06 HWSC0001I      SUPER MEMBER NAME=      CM0 ACK TOQ=
14.06.06 HWSC0001I    NO ACTIVE IMSPLEX
14.06.06 HWSC0001I    PORT=4101     STATUS=ACTIVE     KEEPAV=0 NUMSOC=5 EDIT=
               TIMEOUT=0
14.06.06 HWSC0001I      CLIENTID USERID   TRANCODE DATASTORE STATUS
         SECOND CLNTPORT IP-ADDRESS            APSB-TOKEN
14.06.06 HWSC0001I      MSG0001  CEX001   PART     IMSA      RECV WFCM
         1 21447  172.017.069.027
14.06.06 HWSC0001I      CLICLA01 CEX001            IMSB      CONN
         24 21436  172.017.069.027
14.06.06 HWSC0001I      CLICLI02 CEX001            IMSA      CONN
         24 21435  172.017.069.027
14.06.06 HWSC0001I      CLICLI01 CEX001            IMSA      CONN
         24 21434  172.017.069.027
14.06.06 HWSC0001I      TOTAL CLIENTS=4  RECV=1 CONN=3 XMIT=0 OTHER=0
14.06.06 CEX5055I HWSOPGS1 Command complete
****************************** Bottom of Data ******************************
```

Figure 46. Example: Issuing an IMS Connect command from the command shell

# Chapter 9. Monitoring and controlling systems and their components

Use the Status Monitor to view activity on one or more IMS Connect systems, and to perform actions such as starting and stopping datastores, ports, ODBMs, IMS aliases, and exits; stopping and resuming communications on an MSC TCP/IP link or remote IMS Connect instance; and reloading a user exit dynamically.

To monitor and control systems and their components:

1. From the IMS Connect Extensions ISPF dialog, select option 2 **Operations**.
2. Enter **SM** next to a system or a group. If you enter the Status Monitor from a group, all components or ports within that group appear.
3. Select the view: either 1 (Ports view) or 2 (Systems view).

The Datastore Monitor provides more detailed information about datastores and enables selected datastores to be started, stopped, drained, or resumed. To open the Datastore Monitor from the Systems view, enter **DS** next to a system or datastore.

**Notes:**

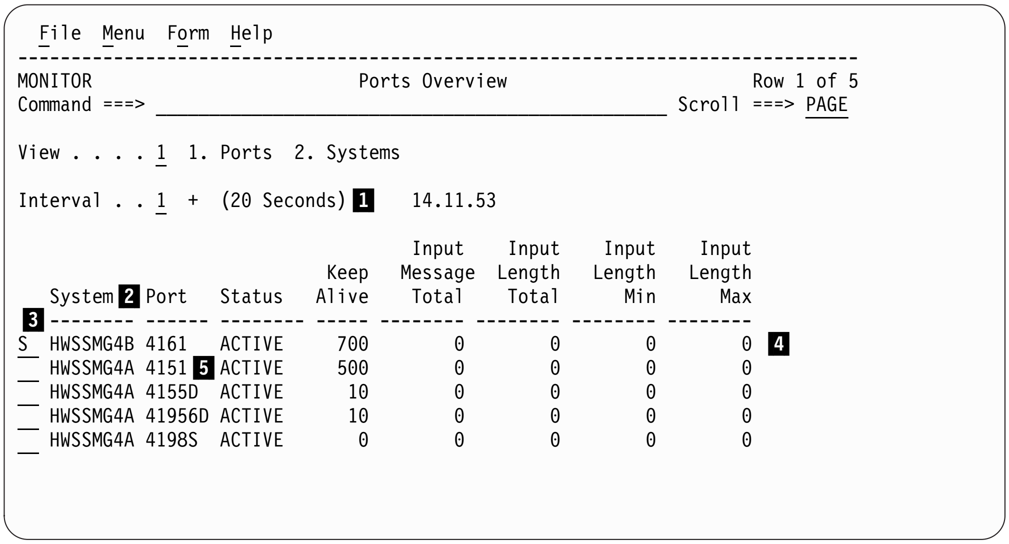
1. To display activity for a system, statistics collection must be active.
2. You can select which time zone to use (server, local, or GMT) from option 0 **Profile**.
3. To refresh information press Enter. To refresh information every 20 seconds, enter the primary command: `GO`

   To stop the `GO` command, press the escape key (Esc).

## Status Monitor: ports view

Status Monitor option 1 shows the TCP/IP ports of the IMS Connect systems you selected.

The following figure shows an example of the Ports Overview panel:



```
  File  Menu  Form  Help
 --------------------------------------------------------------------------------
 MONITOR                         Ports Overview                       Row 1 of 5
 Command ===> ________________________________________________ Scroll ===> PAGE

 View . . . . 1  1. Ports  2. Systems

 Interval . . 1  +  (20 Seconds) 1   14.11.53

                                       Input    Input    Input    Input
                               Keep  Message   Length   Length   Length
   System 2 Port   Status     Alive    Total    Total      Min      Max
 3 -------- ------ -------- ----- -------- -------- -------- --------
 S  HWSSMG4B 4161   ACTIVE      700        0        0        0        0  4
 __ HWSSMG4A 4151 5 ACTIVE      500        0        0        0        0
 __ HWSSMG4A 4155D  ACTIVE       10        0        0        0        0
 __ HWSSMG4A 41956D ACTIVE       10        0        0        0        0
 __ HWSSMG4A 4198S  ACTIVE        0        0        0        0        0
```

*Figure 47. Status Monitor: Ports Overview*

1 You can view activity for the most recent 20-second or 1-minute interval.

2 You can tab to the column heading and press Enter to sort by that column.

3 Use the **P** (stop) and **T** (start) line actions to stop and start a port; type **S** next to a port to view details about activity for that port. Details are available in the following views:

**History**
View summary activity on that port. A new line is written to the display each time you press Enter but data is only collected at the interval you set (either 1 minute or 20 seconds).

**System**
View activity on that port by datastore, datastore group, and exit, for the interval that was last collected. This view is identical to the system view, but activity is only shown for the selected port.

4 You can select what columns to display with the FORM command. The form applies to both the system and the port view. If the FORM specifies more columns than can fit on a single panel, scroll to those columns with the Right function key (F11) and scroll back with the Left function key (F10).

5 The suffix on the port number identifies the port type.

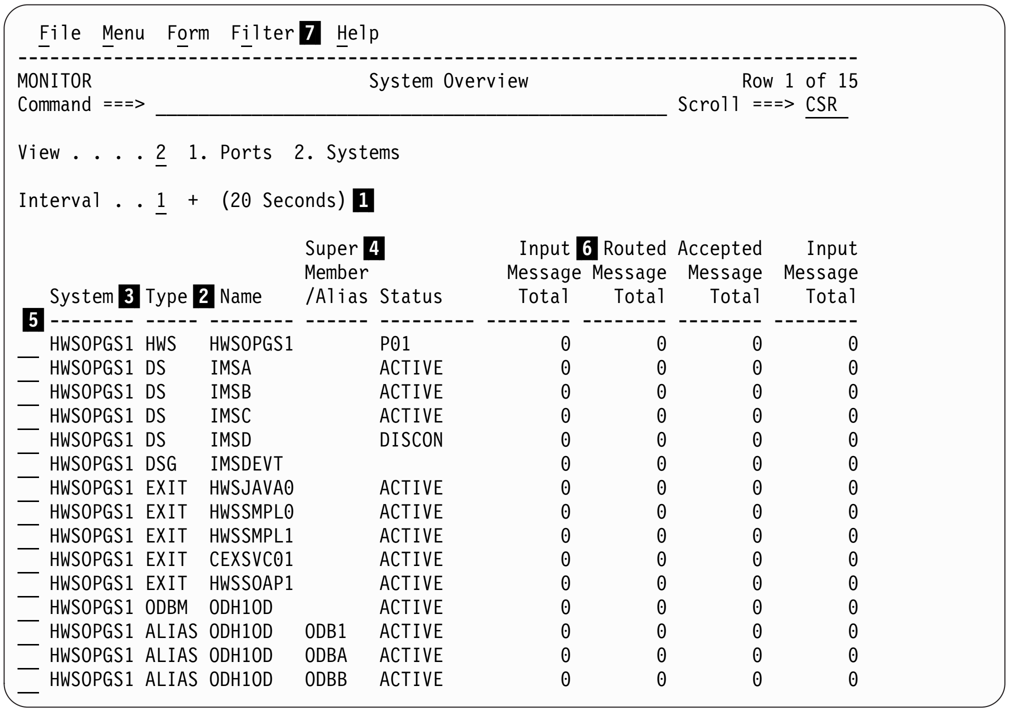
- D indicates a DRDA port.
- S indicates an SSL port.
- No suffix indicates a TCP/IP port defined on the PORT or PORTID parameter of the TCPIP configuration statement.

IMS Connect Extensions detects when a new OTMA or DRDA port has been dynamically added in IMS and automatically adds it to the in-memory list of ports known to IMS Connect Extensions. This allows you to view session statistics for any activity on the added ports without having to restart the IMS Connect instance.

## Status Monitor: systems view

Status Monitor option 2 shows the components of the IMS Connect systems you selected.

The following figure shows an example of the System Overview panel:



```
  File  Menu  Form  Filter 7  Help
-------------------------------------------------------------------------------
MONITOR                         System Overview                     Row 1 of 15
Command ===> ________________________________________________ Scroll ===> CSR

View . . . . 2  1. Ports  2. Systems

Interval . . 1  +  (20 Seconds) 1

                           Super 4           Input 6 Routed Accepted    Input
                           Member          Message Message  Message  Message
   System 3 Type 2 Name    /Alias Status      Total   Total    Total    Total
 5 -------- ----- -------- ------ --------- -------- -------- -------- --------
__ HWSOPGS1 HWS   HWSOPGS1        P01              0        0        0        0
__ HWSOPGS1 DS    IMSA            ACTIVE           0        0        0        0
__ HWSOPGS1 DS    IMSB            ACTIVE           0        0        0        0
__ HWSOPGS1 DS    IMSC            ACTIVE           0        0        0        0
__ HWSOPGS1 DS    IMSD            DISCON           0        0        0        0
__ HWSOPGS1 DSG   IMSDEVT                          0        0        0        0
__ HWSOPGS1 EXIT  HWSJAVA0        ACTIVE           0        0        0        0
__ HWSOPGS1 EXIT  HWSSMPL0        ACTIVE           0        0        0        0
__ HWSOPGS1 EXIT  HWSSMPL1        ACTIVE           0        0        0        0
__ HWSOPGS1 EXIT  CEXSVC01        ACTIVE           0        0        0        0
__ HWSOPGS1 EXIT  HWSSOAP1        ACTIVE           0        0        0        0
__ HWSOPGS1 ODBM  ODH1OD          ACTIVE           0        0        0        0
__ HWSOPGS1 ALIAS ODH1OD   ODB1   ACTIVE           0        0        0        0
__ HWSOPGS1 ALIAS ODH1OD   ODBA   ACTIVE           0        0        0        0
__ HWSOPGS1 ALIAS ODH1OD   ODBB   ACTIVE           0        0        0        0
```

*Figure 48. Status Monitor: System Overview*

**1** The interval determines the time window that the summary statistics represent.

**2** The panel summarizes statistics for each system component:

**HWS** The IMS Connect system

**DSG** Datastore group

**DS** Datastore

**EXIT** User exit

**ODBM** Open Database Manager target

**ALIAS** IMS alias

**MSC** IMS Multiple Systems Coupling link

**RICON** Remote IMS Connect system

Enter / (slash) next to a component to view available actions.

**3** The System column shows the name of the system.

**4** The Super Member/Alias column shows the name of the IMS Connect super member or IMS alias.

**Tip:** Use the filter to create a view of a super member or IMS alias across the sysplex.

**5** Enter **S** next to a system or its component to get an historical view of activity for that part.

Use the **P** (stop) and **T** (start) line actions to stop and start a system component or to stop and resume communications on an MSC physical link or with a remote IMS Connect instance. You cannot stop a datastore group.

Enter **DS** next to a system or datastore to open the Datastore Monitor.

The line actions available for a system in the Status Monitor are similar to those available for a system in the Operations dialog. See "Operations - Systems view" on page 93 for a summary card or enter / (slash) next to a component to view all commands available for that component.

**6** You can select what columns to display with the FORM command, or press the Form function key (F5). The form applies to both the system and the port view. If the FORM specifies more columns than can fit on a single panel, scroll to those columns with the Right function key (F11) and scroll back with the Left function key (F10). You can also sort components by any column by tabbing to the column heading and pressing Enter.

**7** You can define a filter by using the Filter function key (F6), Filter action bar menu, or **FILTER** primary command. You can use a filter to display only those systems matching a string or substring from the name of the system component, the component type, system name, or status.

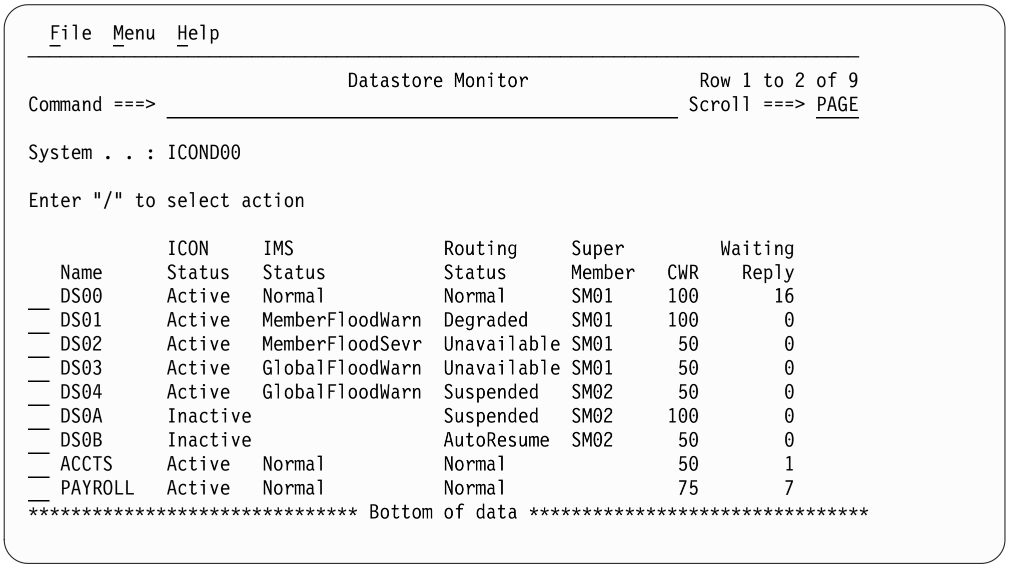
Toggle a filter on and off with these primary commands:

```
FILTER ON
FILTER OFF
```

## Datastore Monitor

The Datastore Monitor provides more detailed information about datastores and enables selected datastores to be started, stopped, drained, or resumed.

The Datastore Monitor dialog is an extension of the Status Monitor that provides additional options for managing datastores.



```
  File  Menu  Help
 ------------------------------------------------------------------------------
                              Datastore Monitor              Row 1 to 2 of 9
 Command ===> ________________________________________________ Scroll ===> PAGE

 System . . : ICOND00

 Enter "/" to select action

              ICON     IMS              Routing     Super         Waiting
    Name      Status   Status           Status      Member   CWR    Reply
 __ DS00      Active   Normal           Normal      SM01     100       16
 __ DS01      Active   MemberFloodWarn  Degraded    SM01     100        0
 __ DS02      Active   MemberFloodSevr  Unavailable SM01      50        0
 __ DS03      Active   GlobalFloodWarn  Unavailable SM01      50        0
 __ DS04      Active   GlobalFloodWarn  Suspended   SM02      50        0
 __ DS0A      Inactive                  Suspended   SM02     100        0
 __ DS0B      Inactive                  AutoResume  SM02      50        0
 __ ACCTS     Active   Normal           Normal                50        1
 __ PAYROLL   Active   Normal           Normal                75        7
 ****************************** Bottom of data ********************************
```

*Figure 49. Datastore Monitor*

The Datastore Monitor provides the following additional status information:

**ICON status**

This field shows if the XCF connection is Active, Inactive, or Discon (disconnected).

**IMS Status**

This field shows the availability of an active IMS system for OTMA transactions. If the ICON status value is inactive or disconnected, the IMS Status will be blank. If the ICON status value is active, the IMS Status values are:

**Normal**

The IMS system is available for work.

**Unknown**

The target IMS system is V9 or earlier.

**MemberFloodWarn**

The datastore is in a Member Flood Warning state.

**MemberFloodSevr**

The datastore is in a Member Flood Severe state.

**GlobalFloodWarn**

The datastore is in a Global Flood Warning state.

**GlobalFloodSevr**

The datastore is in a Global Flood Severe state.

See *IMS Communications and Connections* for more information on the Degraded (Warning) and Unavailable (severe) flood states.

**Routing Status**

This field shows the status of the datastore as a candidate for OTMA Routing in IMS Connect Extensions. The values are:

**Normal**

The datastore is potentially a candidate for routing.

**NormalWLB**

The datastore is potentially a candidate for routing, and workload balancing is active.

**Degraded**

The datastore might be a candidate for routing but only if there are no other candidates in Normal state. This could be due to a warning-level IMS status.

**Unavailable**

The datastore is not a candidate for routing. This could be due to a severe-level IMS status.

**SusCWRZero**

The datastore has been suspended as a candidate for routing. Workload balancing is active and the datastore has a capacity weight ratio of zero.

**Suspended**

The datastore has been suspended as a candidate for routing. The datastore has been drained without the autoresume option. The datastore might be stopped - see ICON Status.

**SusAutoRes**
The datastore has been suspended as a candidate for routing. It will autoresume when the datastore has been stopped and then becomes active again.

**AutoRes**
The datastore is stopped but if it is started it will automatically become a candidate for routing.

| **Super Member**
| The SMEMBER defined in the IMS Connect system configuration.

**CWR** This field shows the capacity weight rating of the datastore relative to other datastores.

A value of zero indicates that the datastore is not currently a candidate for routing.

**Waiting Reply**
The number of OTMA sessions waiting for a reply from the datastore.

If IMS Connect Extensions in the target IMS Connect system does not support this field, all of the Waiting Reply values will be spaces.

### Line actions

**T** Start the datastore.

**P** Stop the datastore.

**DA** Drain the datastore, withdraw it as a candidate for routing, and set AutoResume on.

**D** Drain the datastore, withdraw it as a candidate for routing, and set AutoResume off.

**R** Restore the datastore as a candidate for routing (Resume).

## Customizing the Status Monitor

Use the Form Definition panel to select what information to display in the Status Monitor.

You can make the following types of changes:
- Add columns. For example, show the number of bytes of a given message type processed per second.
- Remove columns.
- Change the order of columns.

To display the Form Definition panel from any Status Monitor view, enter the **FORM** command or press the Form key (F5) when viewing systems, system details, ports, and port details.

To restore the Status Monitor form to its defaults, enter the **RESet FORM** primary command from the Form Definition panel.

### Changing the default Status Monitor layout

You can enter the **FORM** command when viewing systems, system details, ports, and port details. Each of these panels has its own form, so when you modify the form it only changes the appearance of the relevant panel.

The following table shows the type of information that you can set in the form.

*Table 3. Message types for which statistics are collected*

| Name | Description | Count 1 functions? | Length 2 functions? |
|---|---|---|---|
| INPUT | All messages passed to the READ exit. A message might not always arrive at the READ exit. For example, if the message has an invalid message ID. | ✓ | ✓ |
| RTPIPE | Resume TPIPE messages processed by the READ exit. | ✓ | ✓ |
| SENDONLY | Send-Only messages processed by the READ exit. | ✓ | ✓ |
| SENTERR | "Sent in error" messages are those messages processed by the XMIT routine that contain the RSM ID string in the output data. Not included are messages where the RSM specifies RC=4 at READ exit return. This statistic does not apply to an IWEB client. | ✓ | ✓ |
| SENTOK | "Sent OK" messages are any messages not containing the RSM ID string. These include responses, application data, command responses, and DFS messages. | ✓ | ✓ |
| ACCEPTED | Messages processed by the READ exit with RC=0 that were not then rejected by IMS Connect Extensions because of pacing, security, or routing. | ✓ | |
| REJECTED | Total of messages rejected. | ✓ | |
| ROUTED | Messages routed by IMS Connect Extensions to a target datastore that is different from the original destination. | ✓ | |
| ACK | Positive acknowledgment (ACK) from the remote client. | ✓ | |
| NAK | Negative acknowledgment (NAK) from the remote client. | ✓ | |
| DEALLOC | Deallocation response from remote client. | ✓ | |
| RETURNED | Input messages for which the READ exit returned RC=4. Such messages include an RSM, ping response, or user data. | ✓ | |
| REJEXER | Messages rejected by the EXER exit routine because of a protocol error. | ✓ | |
| REJSEC | Messages rejected by IMS Connect Extensions security processing. | ✓ | |
| REJPACE | Messages rejected by IMS Connect Extensions pacing. | ✓ | |
| REJROUTE | Messages rejected by IMS Connect Extensions routing processing. | ✓ | |
| REQUEST | Messages passed to the message exit READ routine that were not considered response or control type messages. For example, transaction input, IMS commands, and conversational messages. | ✓ | |
| IGNORE | Messages passed to the message exit routine that were not processed by IMS Connect Extensions. Includes cases where the exit was marked as Inactive. | ✓ | |
| REJOTHER | "Rejected other". This count includes messages returned from the message exit READ routine that were rejected by IMS Connect Extensions due to an error processing the returned OTMA structure. When a message is rejected in this manner, the counts for IMS Connect system and the user exit are incremented by one: because the datastore cannot be determined, the message is counted for an "unknown" datastore. | ✓ | |

**Notes:**

1 IMS Connect Extensions collects count statistics for each message type. These can be expressed with the following functions (set in the Func column):

**TOTAL**
Number of messages processed in the interval.

**%TOTAL**
Percentage of messages processed in the interval relative to the number of messages of the same type processed in all other intervals. (Useful only when browsing details of systems; elsewhere, always 100 or 0.)

**RATE** The average rate messages were processed per second over the interval.

**2** For some messages, message length statistics are also available with these functions (set in the Func column):

**AVE** Average volume in bytes processed during the interval.

**MAX** Size in bytes of largest message processed in the interval.

**MIN** Size in bytes of the smallest message processed in the interval.

**TOTAL**
Size in bytes of all messages processed in the interval.

**%TOTAL**
Percentage of total bytes processed during the interval relative to the number of total bytes of messages of that type processed in all intervals. (Useful only when browsing details of systems; elsewhere, always 100 or 0.)

**RATE** The average number of bytes per second processed during the interval.

# Chapter 10. Monitoring and controlling active sessions

You can list active sessions on groups of systems (for example, a sysplex) or on individual systems.

Active sessions are connections (such as socket sessions) that are currently processing message exchanges between a client and IMS Connect. Because active sessions typically complete quickly, you will generally display them when you are aware of a transaction that is taking longer to complete than expected.

There are three types of sessions:

**OTMA**
Open Transaction Manager Access sessions

**ODBM**
Open Database Manager sessions

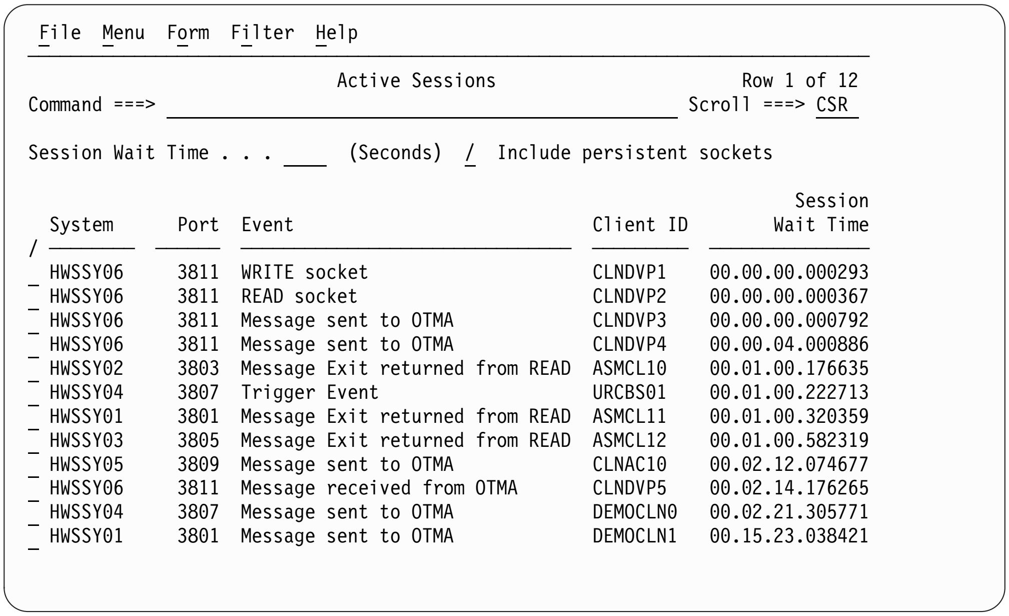
**MSC** Multiple Systems Coupling sessions

**Note:** To display active session details for a system, event collection must be active and the collection level must be greater than 1.

## Displaying active sessions

To list active sessions:

1. From the IMS Connect Extensions ISPF dialog, select option 2 **Operations**.
2. Enter **AS** next to a system or a group. If you select active sessions for a group, then active sessions on any system in that group may appear.



```
  File  Menu  Form  Filter  Help
 ──────────────────────────────────────────────────────────────────────────────
                             Active Sessions                      Row 1 of 12
Command ===> ________________________________________________ Scroll ===> CSR

Session Wait Time . . . ____  (Seconds)  /  Include persistent sockets

                                                                      Session
  System      Port  Event                           Client ID       Wait Time
/ ────────  ──────  ──────────────────────────────  ─────────  ───────────────
_ HWSSY06     3811  WRITE socket                    CLNDVP1    00.00.00.000293
_ HWSSY06     3811  READ socket                     CLNDVP2    00.00.00.000367
_ HWSSY06     3811  Message sent to OTMA            CLNDVP3    00.00.00.000792
_ HWSSY06     3811  Message sent to OTMA            CLNDVP4    00.00.04.000886
_ HWSSY02     3803  Message Exit returned from READ ASMCL10    00.01.00.176635
_ HWSSY04     3807  Trigger Event                   URCBS01    00.01.00.222713
_ HWSSY01     3801  Message Exit returned from READ ASMCL11    00.01.00.320359
_ HWSSY03     3805  Message Exit returned from READ ASMCL12    00.01.00.582319
_ HWSSY05     3809  Message sent to OTMA            CLNAC10    00.02.12.074677
_ HWSSY06     3811  Message received from OTMA      CLNDVP5    00.02.14.176265
_ HWSSY04     3807  Message sent to OTMA            DEMOCLN0   00.02.21.305771
_ HWSSY01     3801  Message sent to OTMA            DEMOCLN1   00.15.23.038421
```

*Figure 50. The Active Sessions panel (showing a customized view)*

### Actions

Here are tasks you can perform from the Active Sessions panel:

- To refresh the list, press Enter.
- To drain an active session, enter **D** next to it. Eligible IMS Connect sessions are marked for closure at the next inbound message.
- To stop an active session, enter **P** next to it.
- To view additional columns of information about sessions, scroll right using the Right function key (F11).
- To change the order of columns or to perform additional customization, use the FORM primary command or press the Form function key (F5).
- To analyze an active session in more detail, enter **S** (session information) next to it.
- To sort active sessions by a given column, tab to the column heading and press Enter.
- Filter sessions by:
  - Listing only those sessions that have been waiting since the last event for longer than a given time by entering a **Session Wait Time** in seconds.
  - Including or excluding persistent socket sessions by toggling **Include persistent sockets**. If you select this option, idle connections will be displayed. These idle sessions may be in one of the following states:
    - Sockets in the Read Prepare state
    - Sockets relating to SSL connections in the Begin Secure Environment Select state
    - Sockets relating to Local Adapter connections in the Local Message Send-then-Receive state
    - Open Database sockets in Trigger Event state

    Active persistent socket sessions appear regardless of the state of this toggle.
  - Defining and then applying a filter with the FILTER primary command (function key F6).

**Related tasks**:

"Filtering active sessions" on page 115
Use the Active Sessions Filter panel to define selection criteria. Only sessions matching the selection criteria will appear.

## Analyzing an active session - OTMA activity

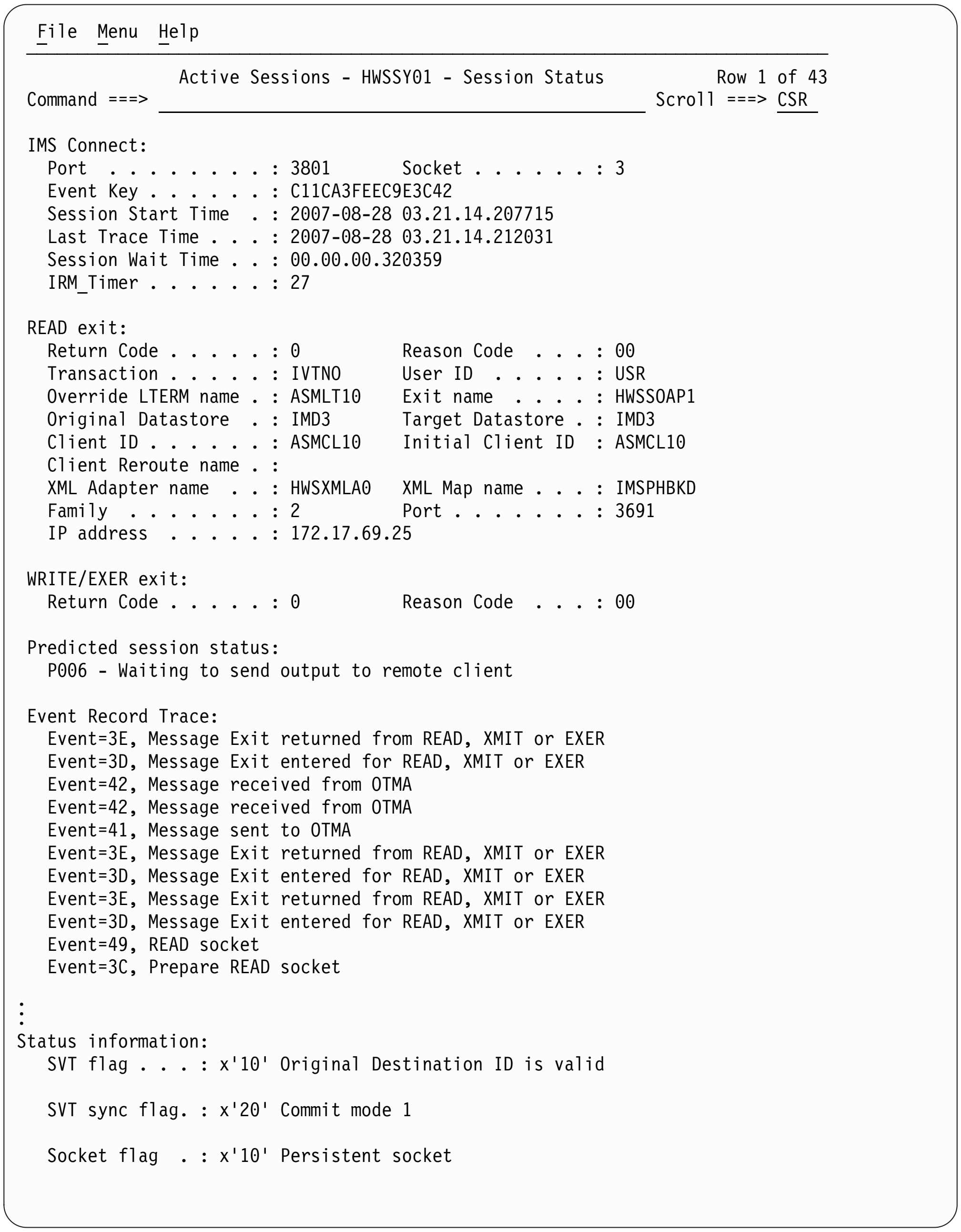
The Session Details panel provides detailed information for a session. To access the Session Details panel, enter **S** next to the session.

### About this task

Figure 50 on page 109 is an example of a customized Active Sessions form showing OTMA activity.

### Procedure

Enter **S** next to one of the OTMA sessions displayed in the Active Sessions panel. The following figure shows an example of a detailed view of an OTMA session.



```
 File  Menu  Help
 ------------------------------------------------------------------------------
               Active Sessions - HWSSY01 - Session Status          Row 1 of 43
Command ===> ________________________________________________ Scroll ===> CSR

IMS Connect:
  Port  . . . . . . . . : 3801       Socket . . . . . . : 3
  Event Key . . . . . . : C11CA3FEEC9E3C42
  Session Start Time  . : 2007-08-28 03.21.14.207715
  Last Trace Time . . . : 2007-08-28 03.21.14.212031
  Session Wait Time . . : 00.00.00.320359
  IRM_Timer . . . . . . : 27

READ exit:
  Return Code . . . . . : 0          Reason Code  . . . : 00
  Transaction . . . . . : IVTNO      User ID  . . . . . : USR
  Override LTERM name . : ASMLT10    Exit name  . . . . : HWSSOAP1
  Original Datastore  . : IMD3       Target Datastore . : IMD3
  Client ID . . . . . . : ASMCL10    Initial Client ID  : ASMCL10
  Client Reroute name . :
  XML Adapter name  . . : HWSXMLA0   XML Map name . . . : IMSPHBKD
  Family  . . . . . . . : 2          Port . . . . . . . : 3691
  IP address  . . . . . : 172.17.69.25

WRITE/EXER exit:
  Return Code . . . . . : 0          Reason Code  . . . : 00

Predicted session status:
  P006 - Waiting to send output to remote client

Event Record Trace:
  Event=3E, Message Exit returned from READ, XMIT or EXER
  Event=3D, Message Exit entered for READ, XMIT or EXER
  Event=42, Message received from OTMA
  Event=42, Message received from OTMA
  Event=41, Message sent to OTMA
  Event=3E, Message Exit returned from READ, XMIT or EXER
  Event=3D, Message Exit entered for READ, XMIT or EXER
  Event=3E, Message Exit returned from READ, XMIT or EXER
  Event=3D, Message Exit entered for READ, XMIT or EXER
  Event=49, READ socket
  Event=3C, Prepare READ socket

:
:
Status information:
   SVT flag . . . : x'10' Original Destination ID is valid

   SVT sync flag. : x'20' Commit mode 1

   Socket flag  . : x'10' Persistent socket
```

*Figure 51. Active Sessions - Session Detail (returned from exit)*

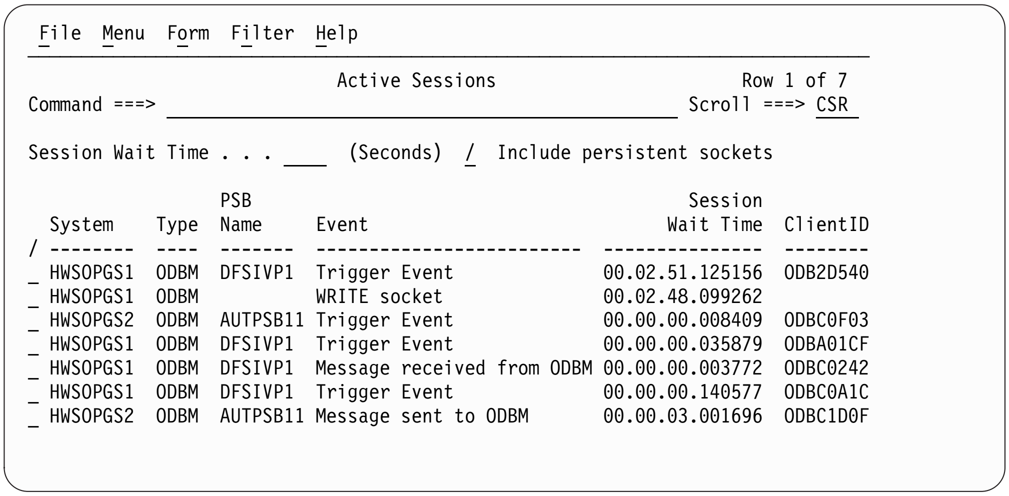
This panel displays different levels of information depending on the state of the session. For example, a session in the READ socket event state contains basic information. However, as shown in this example, when it returns from the exit the session contains more information.

## Analyzing an active session - ODBM activity

The Session Details panel provides detailed information showing activity for a selected ODBM session.

## About this task

The following figure is an example of a customized Active Sessions form showing ODBM activity.

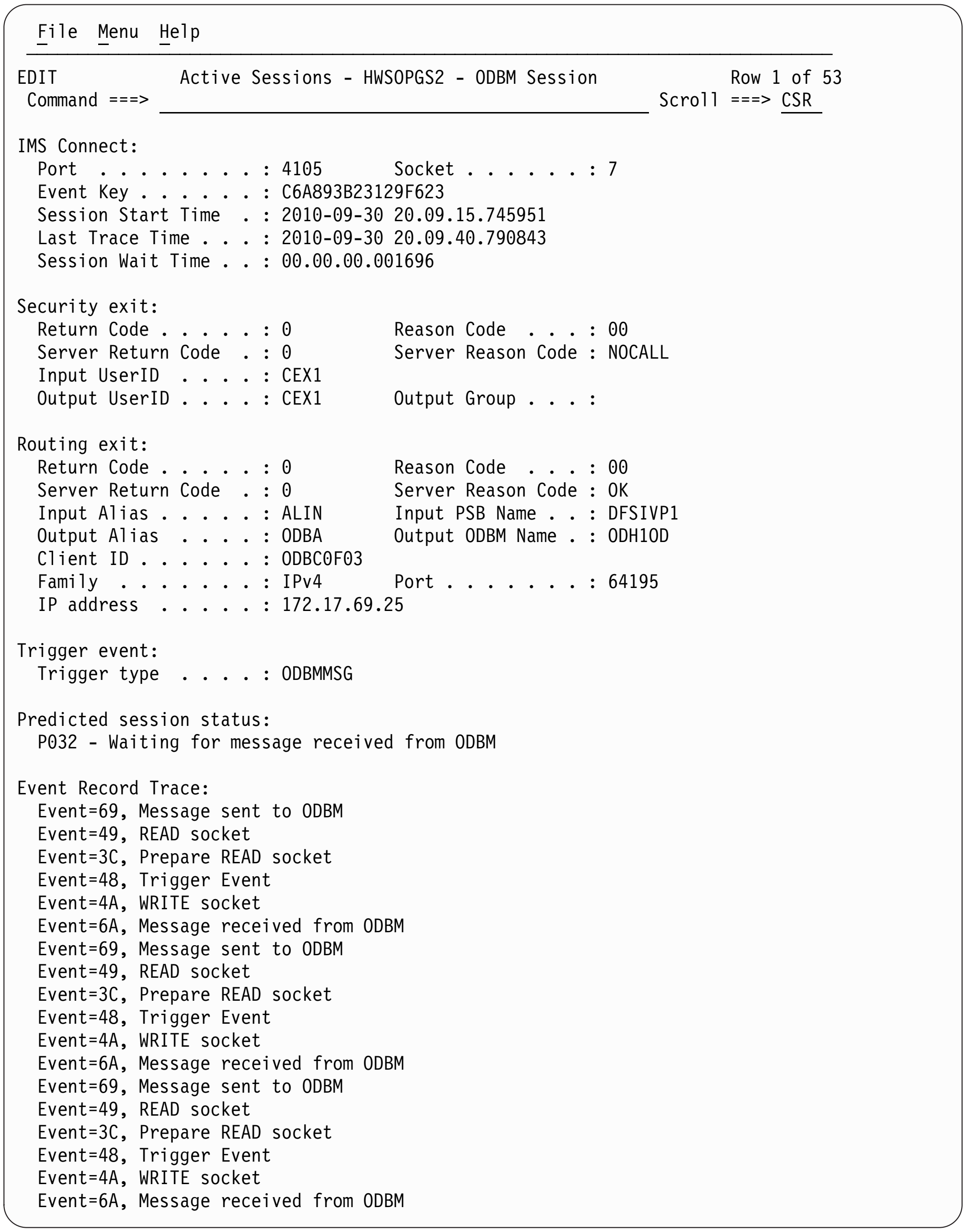


```
 File  Menu  Form  Filter  Help
───────────────────────────────────────────────────────────────────────────────
                             Active Sessions                      Row 1 of 7
Command ===> ________________________________________________ Scroll ===> CSR_

Session Wait Time . . . ____  (Seconds)  /  Include persistent sockets

                  PSB                                               Session
  System    Type  Name     Event                                  Wait Time  ClientID
/ --------  ----  -------  -------------------------  ---------------  --------
_ HWSOPGS1  ODBM  DFSIVP1  Trigger Event              00.02.51.125156  ODB2D540
_ HWSOPGS1  ODBM           WRITE socket               00.02.48.099262
_ HWSOPGS2  ODBM  AUTPSB11 Trigger Event              00.00.00.008409  ODBC0F03
_ HWSOPGS1  ODBM  DFSIVP1  Trigger Event              00.00.00.035879  ODBA01CF
_ HWSOPGS1  ODBM  DFSIVP1  Message received from ODBM 00.00.00.003772  ODBC0242
_ HWSOPGS1  ODBM  DFSIVP1  Trigger Event              00.00.00.140577  ODBC0A1C
_ HWSOPGS2  ODBM  AUTPSB11 Message sent to ODBM       00.00.03.001696  ODBC1D0F
```

*Figure 52. Active Sessions panel showing ODBM activity (customized view)*

## Procedure

Enter **S** next to one of the ODBM sessions displayed in the Active Sessions panel. The following figure shows an example of a detailed view of an ODBM session.



```
  File  Menu  Help
 ________________________________________________________________________________
EDIT            Active Sessions - HWSOPGS2 - ODBM Session             Row 1 of 53
 Command ===> ________________________________________________ Scroll ===> CSR

IMS Connect:
  Port  . . . . . . . . : 4105       Socket . . . . . . : 7
  Event Key . . . . . . : C6A893B23129F623
  Session Start Time  . : 2010-09-30 20.09.15.745951
  Last Trace Time . . . : 2010-09-30 20.09.40.790843
  Session Wait Time . . : 00.00.00.001696

Security exit:
  Return Code . . . . . : 0          Reason Code  . . . : 00
  Server Return Code  . : 0          Server Reason Code : NOCALL
  Input UserID  . . . . : CEX1
  Output UserID . . . . : CEX1       Output Group . . . :

Routing exit:
  Return Code . . . . . : 0          Reason Code  . . . : 00
  Server Return Code  . : 0          Server Reason Code : OK
  Input Alias . . . . . : ALIN       Input PSB Name . . : DFSIVP1
  Output Alias  . . . . : ODBA       Output ODBM Name . : ODH1OD
  Client ID . . . . . . : ODBC0F03
  Family  . . . . . . . : IPv4       Port . . . . . . . : 64195
  IP address  . . . . . : 172.17.69.25

Trigger event:
  Trigger type  . . . . : ODBMMSG

Predicted session status:
  P032 - Waiting for message received from ODBM

Event Record Trace:
  Event=69, Message sent to ODBM
  Event=49, READ socket
  Event=3C, Prepare READ socket
  Event=48, Trigger Event
  Event=4A, WRITE socket
  Event=6A, Message received from ODBM
  Event=69, Message sent to ODBM
  Event=49, READ socket
  Event=3C, Prepare READ socket
  Event=48, Trigger Event
  Event=4A, WRITE socket
  Event=6A, Message received from ODBM
  Event=69, Message sent to ODBM
  Event=49, READ socket
  Event=3C, Prepare READ socket
  Event=48, Trigger Event
  Event=4A, WRITE socket
  Event=6A, Message received from ODBM
```

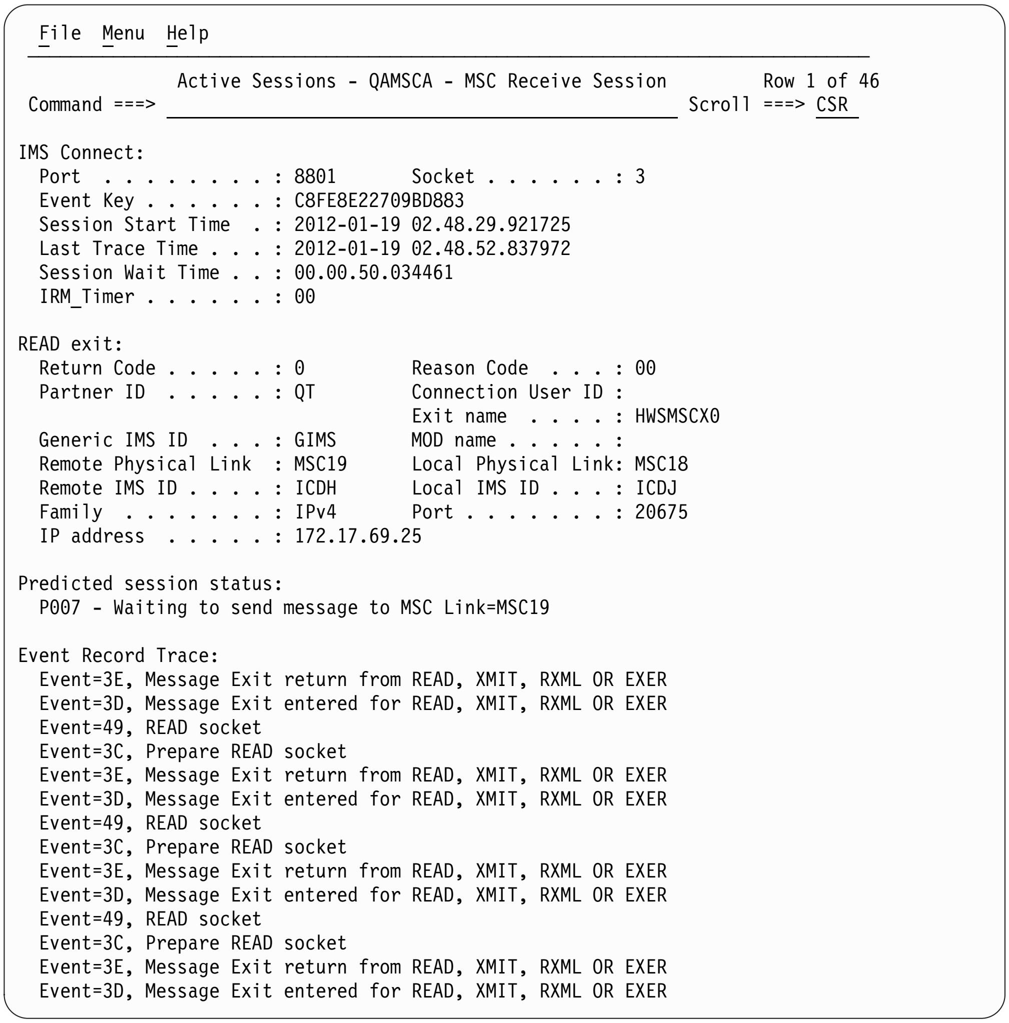
*Figure 53. Active Sessions - Session Detail: ODBM activity*

## Analyzing an active session - MSC activity

The Session Details panel provides detailed information showing activity for a selected MSC session.

### Procedure

Enter **S** next to one of the MSC sessions displayed in the Active Sessions panel. The following figure shows an example of a detailed view of an MSC session.



```
 File  Menu  Help
──────────────────────────────────────────────────────────────────────────────
              Active Sessions - QAMSCA - MSC Receive Session         Row 1 of 46
 Command ===> ________________________________________________ Scroll ===> CSR

IMS Connect:
  Port  . . . . . . . . : 8801       Socket . . . . . . : 3
  Event Key . . . . . . : C8FE8E22709BD883
  Session Start Time  . : 2012-01-19 02.48.29.921725
  Last Trace Time . . . : 2012-01-19 02.48.52.837972
  Session Wait Time . . : 00.00.50.034461
  IRM_Timer . . . . . . : 00

READ exit:
  Return Code . . . . . : 0          Reason Code  . . . : 00
  Partner ID  . . . . . : QT         Connection User ID :
                                     Exit name  . . . . : HWSMSCX0
  Generic IMS ID  . . . : GIMS       MOD name . . . . . :
  Remote Physical Link  : MSC19      Local Physical Link: MSC18
  Remote IMS ID . . . . : ICDH       Local IMS ID . . . : ICDJ
  Family  . . . . . . . : IPv4       Port . . . . . . . : 20675
  IP address  . . . . . : 172.17.69.25

Predicted session status:
  P007 - Waiting to send message to MSC Link=MSC19

Event Record Trace:
  Event=3E, Message Exit return from READ, XMIT, RXML OR EXER
  Event=3D, Message Exit entered for READ, XMIT, RXML OR EXER
  Event=49, READ socket
  Event=3C, Prepare READ socket
  Event=3E, Message Exit return from READ, XMIT, RXML OR EXER
  Event=3D, Message Exit entered for READ, XMIT, RXML OR EXER
  Event=49, READ socket
  Event=3C, Prepare READ socket
  Event=3E, Message Exit return from READ, XMIT, RXML OR EXER
  Event=3D, Message Exit entered for READ, XMIT, RXML OR EXER
  Event=49, READ socket
  Event=3C, Prepare READ socket
  Event=3E, Message Exit return from READ, XMIT, RXML OR EXER
  Event=3D, Message Exit entered for READ, XMIT, RXML OR EXER
```

*Figure 54. Active Sessions - Session Detail: MSC activity*

## Viewing TCP/IP information for a session

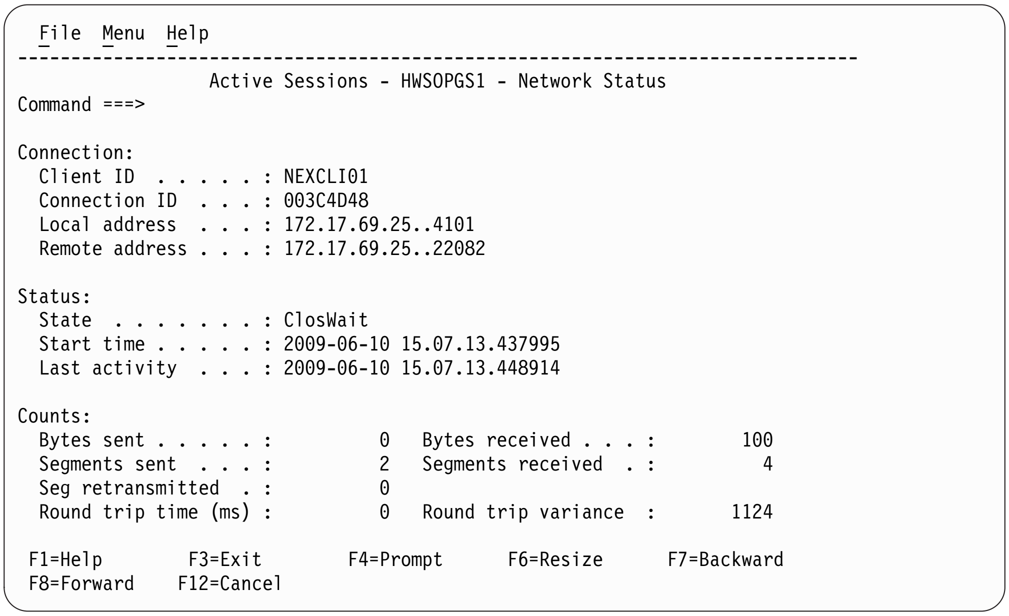
The Network Status panel provides TCP/IP information for a selected session.

### About this task

The information shown in the Network Status panel is similar to that provided by the **NETSTAT** command.

### Procedure

Enter **N** next to one of the sessions displayed in the Active Sessions panel.



```
  File  Menu  Help
-------------------------------------------------------------------------------
                  Active Sessions - HWSOPGS1 - Network Status
Command ===>

Connection:
  Client ID  . . . . . : NEXCLI01
  Connection ID  . . . : 003C4D48
  Local address  . . . : 172.17.69.25..4101
  Remote address . . . : 172.17.69.25..22082

Status:
  State  . . . . . . . : ClosWait
  Start time . . . . . : 2009-06-10 15.07.13.437995
  Last activity  . . . : 2009-06-10 15.07.13.448914

Counts:
  Bytes sent . . . . . :          0   Bytes received . . . :        100
  Segments sent  . . . :          2   Segments received  . :          4
  Seg retransmitted  . :          0
  Round trip time (ms) :          0   Round trip variance  :       1124

 F1=Help        F3=Exit        F4=Prompt      F6=Resize      F7=Backward
 F8=Forward    F12=Cancel
```

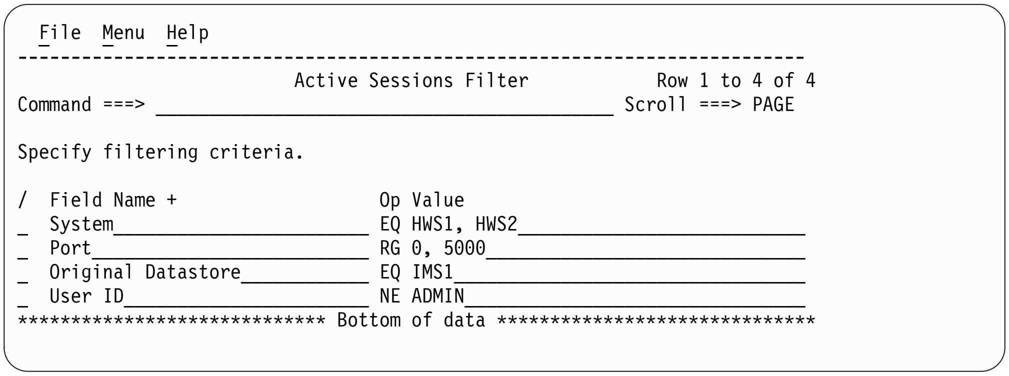
*Figure 55. Example of a Network Status panel showing TCP/IP information*

## Filtering active sessions

Use the Active Sessions Filter panel to define selection criteria. Only sessions matching the selection criteria will appear.

### About this task

From the Active Sessions panel, the primary command FILTER prompts the Active Sessions Filter panel.



```
  File  Menu  Help
--------------------------------------------------------------------------
                          Active Sessions Filter           Row 1 to 4 of 4
Command ===> ___________________________________________ Scroll ===> PAGE

Specify filtering criteria.

/  Field Name +                 Op Value
_  System________________________ EQ HWS1, HWS2___________________________
_  Port__________________________ RG 0, 5000______________________________
_  Original Datastore____________ EQ IMS1_________________________________
_  User ID_______________________ NE ADMIN________________________________
***************************** Bottom of data ******************************
```

*Figure 56. Sessions filter*

You can toggle the state of a filter with the following commands:

```
FILTER ON
FILTER OFF
```

Each selection criterion has the following parts:

**Field Name**
: Tab to the field name and press the Prompt function key (F4) to select a field.

**Op** Enter one of the following operators:

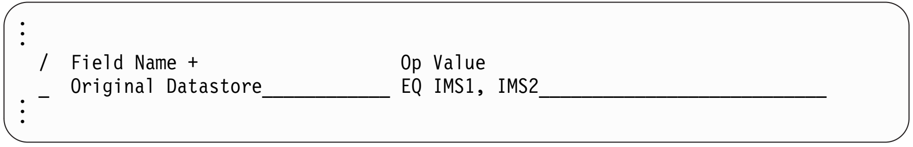
**EQ** Equal to
**NE** Not equal to
**LT** Less than
**LE** Less than or equal to
**GT** Greater than
**GE** Greater than or equal to
**RG** Range: use a comma to delimit that range. For example, to indicate the 0 - 10 range, type:

```
0,10
```

**Value** The field value or, for the EQ and NE operators, comma delimited list of values.

Here are examples of selection criteria:

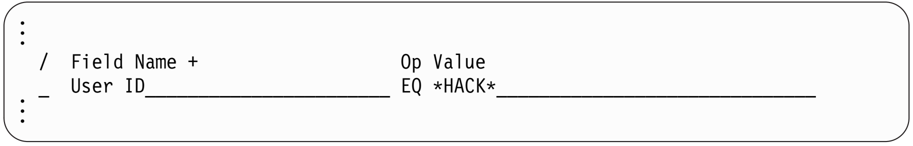
**Show sessions with IMS1 or IMS2 as the original destination datastore**
Set the **Original Datastore** field to equal a list of values:



```
⋮
 /  Field Name +                   Op Value
 _  Original Datastore___________ EQ IMS1, IMS2___________________________
⋮
```

To show sessions that do not have IMS1 or IMS2 as the original datastore, use the NE operator.

**Match sessions by using wildcards**
Use an asterisk (*) to match any characters; use a percent sign (%) to match any single character:

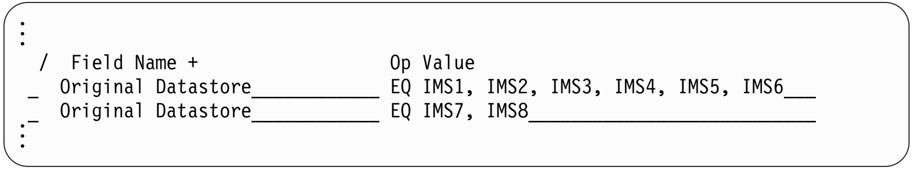


```
⋮
 /  Field Name +                   Op Value
 _  User ID_______________________ EQ *HACK*______________________________
⋮
```

**Match any value when the values do not fit on one line**
To enter a list of values on more than one row, for example if all the values do not fit on one line, enter them as follows:

1. Type the field name, operator, and some of the values on the first line.
2. Type line action **I** to insert a new row or **R** to repeat the current row and press Enter.
3. Use the same field name and operator as the previous row and continue entering values.
4. Repeat these steps to insert more than 2 rows.

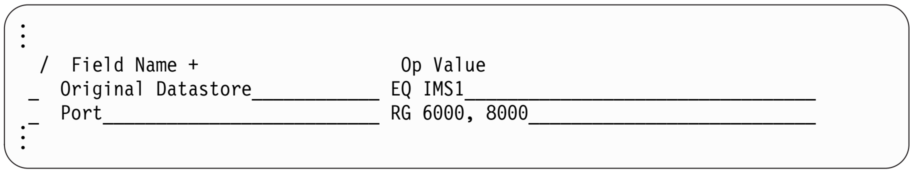
Here is an example that will select sessions with any of the datastores IMS1-IMS8 as the original datastore:



```
⋮
 /  Field Name +                  Op Value
 _ Original Datastore____________ EQ IMS1, IMS2, IMS3, IMS4, IMS5, IMS6___
 _ Original Datastore____________ EQ IMS7, IMS8___________________________
⋮
```

**Show sessions on ports 6000 - 8000 with IMS1 as the original destination datastore**

Here is an example of how to find sessions using more than one selection criteria:



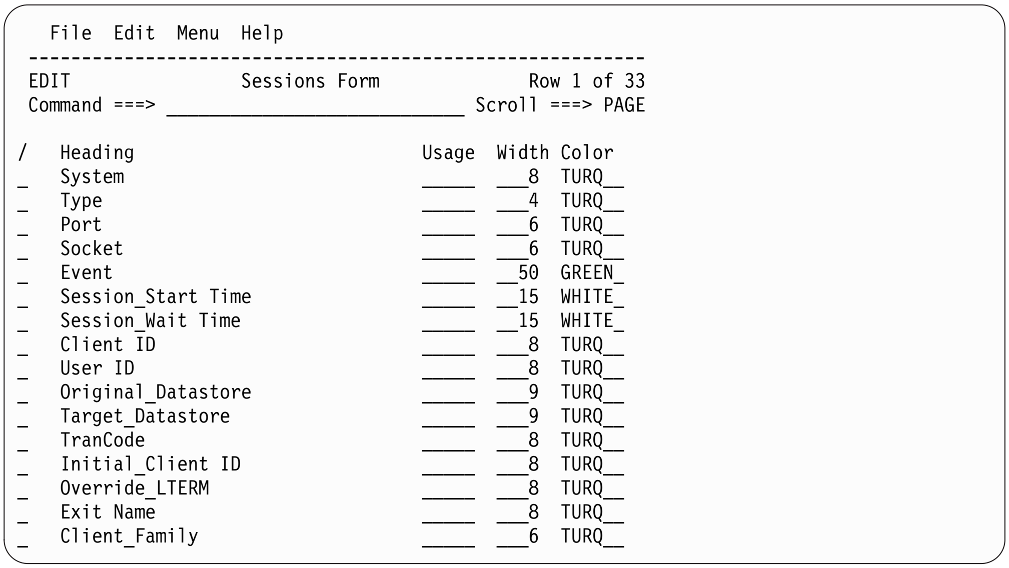
```
⋮
 /  Field Name +                    Op Value
 _  Original Datastore____________  EQ IMS1____________________________
 _  Port__________________________  RG 6000, 8000______________________
⋮
```

When you enter different criteria on different rows, an 'AND' relationship applies.

## Customizing the Active Sessions panel

Use the Sessions Form panel to customize the Active Sessions panel by selecting which columns to display, and the column names, order, width, and color.

From the Active Sessions panel, the primary command **FORM** invokes the Sessions Form panel.



```
    File  Edit  Menu  Help
  ----------------------------------------------------------
  EDIT                 Sessions Form             Row 1 of 33
  Command ===> ____________________________ Scroll ===> PAGE

 /   Heading                            Usage  Width Color
 _   System                             _____  ___8  TURQ__
 _   Type                               _____  ___4  TURQ__
 _   Port                               _____  ___6  TURQ__
 _   Socket                             _____  ___6  TURQ__
 _   Event                              _____  __50  GREEN_
 _   Session_Start Time                 _____  __15  WHITE_
 _   Session_Wait Time                  _____  __15  WHITE_
 _   Client ID                          _____  ___8  TURQ__
 _   User ID                            _____  ___8  TURQ__
 _   Original_Datastore                 _____  ___9  TURQ__
 _   Target_Datastore                   _____  ___9  TURQ__
 _   TranCode                           _____  ___8  TURQ__
 _   Initial_Client ID                  _____  ___8  TURQ__
 _   Override_LTERM                     _____  ___8  TURQ__
 _   Exit Name                          _____  ___8  TURQ__
 _   Client_Family                      _____  ___6  TURQ__
```

*Figure 57. Sessions Form panel*

### Active Sessions panel - default fields

This topic describes the default fields on the Sessions Form panel.

**System**
The IMS Connect system name.

**Type** The session type: OTMA or ODBM.

**Port** The incoming TCP/IP port.

**Socket**
The TCP/IP connection socket.

**Event** The event name.

**Session_Start Time**
The time the session started, in the format `yyyy-mm-dd-hh.mm.ss.cccccc` (where cccccc=microseconds). This time will reflect the **Zone settings for time displays** as specified in the user's IMS Connect Extensions Profile Settings.

**Session_Wait Time**
How long the session has been waiting for a response. In other words, the difference between the current time (taken at the latest refresh) and the time of the last event. The time is in the format `hh.mm.ss.cccccc` (where cccccc=microseconds).

**Client ID**
An identifier of the connecting client.

**User ID**
The user ID of the connecting user.

**Original_Datastore**
The datastore which the message originally assigned as the target.

**Target_Datastore**
The datastore assigned as the target by IMS Connect Extensions.

**TranCode**
The transaction code of the message.

**Initial_Client ID**
| The client identifier at the time the connection was first established.

**Override_LTERM**
The LTERM override name.

**Exit Name**
The name of the exit that processed the session.

**Client_Family**
The addressing family specifying the type of IP address. For example, IPv4 (2) or IPv6 (19).

**Client_Port**
The port number of the client.

**Client_IP_Address**
The IP address of the client.

**Socket_Flag**
The socket flag: persistent, non-persistent, or transaction socket.

**PSB_Name**
The PSB name on the DRDA request.

**ODBM_Name**
The ODBM assigned as the target by IMS Connect Extensions.

**In_Alias**
The alias which the DRDA request originally assigned as the target.

**Out_Alias**
The alias assigned as the target by IMS Connect Extensions.

**Out_UserID**
The user ID provided by the Security exit. This can be blank if the default IMS Connect security exit is being used or if the security exit does not return values for this field.

**Out_Group**
The group name provided by the Security exit. This can be blank if the default IMS Connect security exit is being used or if the security exit does not return values for this field.

**MSC_RmtIMSID**
The MSC remote IMS ID.

**MSC_RemPlkID**
The MSC remote physical link ID.

**MSC_ConnUID**
The MSC connection user ID.

**MSC_GenIMSID**
The MSC generic IMS ID.

**MSC_LclIMSID**
The MSC local IMS ID.

**MSC_LclPlkID**
The MSC local physical link ID.

**IRM_Timer**
The IRM_Timer value for the OTMA or MSC session, in hex format. The value will be blank for all ODBM sessions. For OTMA and MSC sessions where the target IMS Connect system is running a release of IMS Connect Extensions earlier than V2R3, the value will be a "?".

**Message_Count**
The number of input messages received so far by the session. See "Rebalancing sessions across IMS Connect systems" on page 120.

## Changing the default Active Sessions panel layout

You can perform the following tasks from the Sessions Form panel.

### Change which fields are displayed

**Preselect a standard set of fields for your workload**
To preselect for display only the fields relevant to your workload, you can enter one of the following primary commands from the Sessions Form panel.

**Note:** The RESET FORM command first restores the Active Sessions panel to its defaults, so any other changes you have made to the form will be discarded.

```
RESET FORM OTMA
RESET FORM ODBM
RESET FORM MSC
```

**Remove a column**
To remove a column from being displayed on the Active Sessions panel:

1. Enter **O** next to the name of the column you want to remove.

   The letters OMIT appear in the Usage column. If the form is on, the column will not appear.

   You can also type **OMIT** directly in the Usage column.
2. Enter an **O** again to restore the field.

### Change how fields are displayed

**Change the order in which fields are displayed**

The order in which fields are listed in the Sessions Form panel is the same order in which they will be displayed from left to right in the Active Sessions Overview panel. Use the following line actions to move fields up or down the list:

| | |
|---|---|
| **M** | Move |
| **A** | After |
| **B** | Before |

**Change the name of a column**

To change the name of a column, overtype it. You can use underscores to specify line breaks. For example:

```
Session_Wait Time
```

displays as:

```
Session
Wait Time
```

**Change the width of a column**

To change the width of a column, type a number (of characters) in the Width column and press Enter.

If the width is too short to accommodate either the field name or the data, it will be truncated on the right.

**Change the color of a column**

To change the color of a field, enter one of the following values in the Color column:

```
RED
PINK
GREEN
YELLOW
BLUE
TURQ (Turquoise)
WHITE
```

### Restore the default layout

**Reset the Active Sessions panel to the default layout**

To restore the Active Sessions form to its defaults, enter the **RESet FORM** primary command from the Sessions Form panel.

**Tip:** If you want to temporarily revert to the default view in the Active Sessions panel, use the **FORM OFF** command.

## Rebalancing sessions across IMS Connect systems

The Session Message Limit option allows you to automatically respond to the failure of an IMS Connect system by balancing the number of TCP/IP sessions directed to each remaining live IMS Connect system.

Session distribution mechanisms, such as a sysplex distributor, are often used in high-availability environments to route session requests between the available IMS Connect systems. If any of the IMS Connect systems fails, the sysplex distributor routes all new sessions to the surviving IMS Connect systems. When this happens before the failed IMS Connect system can be restarted, the balance of sessions between IMS Connect systems can be disrupted. If the persistent sessions remain indefinitely, session rebalancing might take a very long time or never occur.

The Session Message Limit options in the system definition allow you to specify the maximum number of input messages for a persistent session. When the Limit threshold is reached, the session is closed by IMS Connect Extensions with the expectation that the remote client will create a new session. Session balance is gradually restored as the existing session expires and new sessions are routed to the IMS Connect with the lowest session totals.

The Message Count field on the Active Sessions dialog displays the number of input messages received so far by the session.

To implement automatic session rebalancing for an IMS Connect system, perform the following steps:

- In the system definition, select **Activate Session Message Limit** and specify a maximum number of input messages in the **Limit threshold** field.
- Ensure that remote clients are coded to detect that the socket session is being closed by IMS and to request a new socket.

To rebalance all sessions according to a schedule, use the **`DRAIN TYPE=SESSION`** host command and specify TOKEN=*. You can use automation to run a job on a scheduled basis or simply run the job as required. The member CEXRXC61 in the SCEXSAMP library shows how to drain all sessions using a REXX exec.

# Chapter 11. Automating operations tasks using REXX

The CEX host command environment for REXX enables IMS Connect Extensions commands to be embedded in REXX programs, which allows more flexible automation of IMS Connect operations.

This section documents intended Programming Interfaces that allow the customer to write programs to obtain the services of IMS Connect Extensions.

From a REXX exec you can establish connections to IMS Connects using the CONNECT command. You can issue CEX host commands and obtain command responses.

- Programs can take advantage of REXX features such as variables, conditional logic, and integration with other host environments.
- Programs can connect to multiple IMS Connects.
- Programs can be submitted interactively as well as in batch.

You can use the CEX host command environment to perform tasks such as suspending or resuming routing to a datastore, switching the active journal (equivalent to the JOURNAL SWITCH command), initiating an IMS Connect Extensions resource trace, refreshing and updating definitions (equivalent to the REFRESH command), and running commands on a target system or datastore.

**Note:** The CEX host command environment for REXX supersedes the old IMS Connect Extensions batch command utility CEXDFCMD.

**Related concepts**:

Chapter 21, "CEX host command environment: reference," on page 247
This topic describes the commands and functions that are available in the CEX host command environment for REXX.

## Connections and CONIDs

Console commands operate using a connection that has previously been established by a CONNECT command. All commands supported by the CEX host command environment are console commands except CONNECT, DELAY, OPTION, and WTO.

A CONID is a label that uniquely identifies a connection. It is established by a CONNECT command and then can be referenced by subsequent console commands. If a CONID is not explicitly specified in the CONNECT command, it defaults to the HWSID.

You can use multiple CONNECT statements in the same job and assign a different CONID to each connection. For example, you could connect to several systems and query the tracing status on each of them. If no CONID is specified on a console command then the current default connection is used. The default connection is the last connection that was specified by a CONNECT command. You can also use the OPTION command to make the connection identified by the specified CONID keyword the default connection.

## REXX variables

Each host command that is issued sets a number of REXX variables.

Variable names have the general structure *stem.fieldname*, where *stem* is a common prefix that can be specified on the command by using the STEM option. The default stem is CEX.

**Notes:**

1. *fieldname* can contain arrays of values (.*n*).
2. When an array is returned, a field with the suffix '.0' indicates the total number in the array.

There are three types of variable:

1. Top-level variables relate to the command output to CEXPRINT and MSGOUT.

*Table 4. Top-level REXX variables*

| Variable name | Description |
|---|---|
| *stem*.COMMAND | The command. |
| *stem*.MESSAGE.0 | The number of CEXPRINT messages. |
| *stem*.MESSAGE.*n* | The messages which the batch command utility would have written to CEXPRINT.<br>**Note:** If MSGLVL=VERBOSE is specified then the contents of MESSAGE.*n* will automatically appear in the output. |
| *stem*.PRINT.0 | The number of MSGOUT lines. |
| *stem*.PRINT.*n* | The lines of output which the batch command utility would have written to MSGOUT. |

2. The following response variables are common to all console commands.

*Table 5. Console command response variables*

| Variable name | Description |
|---|---|
| *stem*.RESPONSE.FIELD.0 | The number of response field variables for this request. |
| *stem*.RESPONSE.FIELDNAME.*n* | The number appended to stem.RESPONSE defines the name of a response field variable. The standard variables are: 1=RET; 2=RSN; and 3=TEXT. |
| *stem*.RESPONSE.FIELDDESC.*n* | The description of the response field variable. |
| *stem*.RESPONSE.RET | The return code associated with the response from console services. |
| *stem*.RESPONSE.RSN | The reason code associated with the response from console services. |
| *stem*.RESPONSE.TEXT | A text explanation for the reason code. |

3. Command-specific response variables contain the unique fields returned by individual console commands. They are documented in the reference topics for those commands.

   **Notes:**

   a. Variable names have the general structure *stem*.RESPONSE.*fieldname*
   b. *fieldname* can be compounded. Example: *stem*.RESPONSE.SESSION.CLIENTID.*n*

## Examples: Using the CEX host commands

These examples show the structure of a REXX exec using commands from the CEX host command environment.

## Sample REXX exec: process variables and format output

This sample exec shows how to connect to an IMS Connect system, run a query, and display the results.

```
  parse arg PARM
  address LINK "CEXRXENV INIT"                                  1

  address CEX "OPTION MSGLVL=VERBOSE"                           2

  address CEX "CONNECT HOST=SRV2,PORT=4199,HWSID=HWSOPGS1,"||,  3
                      "PASSTIK=YES,APPLID=CEXTEST"

/* Query Command */
  address CEX "QUERY TYPE=ACEE_CACHE"                           4
  call CheckCEXResponse

/* Process stem variables from Query command */
  say 'IMS Connect Extensions ACEE Cache statistics:'
  do r=4 to CEX.RESPONSE.FIELD.0                                5
    FName = CEX.RESPONSE.FIELDNAME.r
    ResponseVar = CEX.RESPONSE.fname
    rpad = 8 - length(ResponseVar)
    rpadf = ''
    if rpad > 0 then rpadf=left(' ',rpad)
    say '  ' left(CEX.RESPONSE.FIELDNAME.r,20)':'||,
                 ResponseVar||rpadf,
              ' ('CEX.RESPONSE.FIELDDESC.r')'
  end

  address LINK "CEXRXENV TERM"                                  6

return

/* ------------------------------------------------------------------
   This routine checks command responses and formats the return code,
   reason code and text when a unexpected response is encountered.
   ------------------------------------------------------------------ */
CheckCEXResponse:
  if CEX.RESPONSE.RET > 0 then do
    say 'ERROR: Unexpected response encountered'
    say '     : Command              :'CEX.COMMAND
    say '     :   Return Code =      :'CEX.RESPONSE.RET
    say '     :   Reason Code =      :'CEX.RESPONSE.RSN
    say '     :   Text        =      :'CEX.RESPONSE.TEXT
    address LINK "CEXRXENV TERM"
    exit CEX.RESPONSE.RET
  end
  if RC > 4 then do
    address LINK "CEXRXENV TERM"
    exit RC
  end
return
```

*Figure 58. Simple REXX exec that issues IMS Connect Extensions host commands*

**Notes:**

1 Define IMS Connect Extensions to the host command environment table and establish support routines.

2 Issue the OPTION command. This is an IMS Connect Extensions host command.

3 Issue a host command to CONNECT to an IMS Connect system, using a PassTicket to verify access.

**4** Issue a host command to query the state of the ACEE cache on the system identified by HWSID.

**5** Format and display the fields returned by the QUERY command.

**6** Clean up support routines and remove the IMS Connect Extensions entry from the host command environment table.

The **OPTION MSGLVL=VERBOSE** statement displays the start time and lists the host commands that were processed before the command output.

```
FUN1003I Processing started at 2014-09-04 10:37:51
CEX5043I OPTION MSGLVL=VERBOSE
CEX5043I CONNECT HOST=SRV2,PORT=4199,HWSID=HWSOPGS1,PASSTIK=YES,APPLID=CEXTEST
CEX5043I QUERY TYPE=ACEE_CACHE
IMS Connect Extensions ACEE Cache statistics:
   ELEMNUM            :0         (Number of elements in the ACEE cache)
   ELEMLEN            :152       (Length of each element in the ACEE cache)
   INITSIZE           :163840    (Initial size of the ACEE cache)
   XPNDSIZE           :1363968   (The increment by which the ACEE cache may be expanded)
   XPNDMAX            :31        (The maximum number of times the ACEE cache may be expanded)
   XPNDCNT            :0         (The number of times the ACEE cache has been expanded)
   ACEEPCT            :0         (Percentage of the ACEE cache used)
   INSERTS            :0         (Number of elements inserted in the ACEE cache)
   GETS               :0         (Number of get requests to the ACEE cache)
   GETNEXTS           :0         (Number of get next requests to the ACEE cache)
   DELETES            :0         (Number of elements deleted from the ACEE cache)
   STATS              :1         (Number of stat requests to the ACEE cache)
READY
END
```

## Sample REXX exec: use of multiple CONNECTs and STEM

This sample exec shows how to combine the results of a query run on two IMS Connect systems. The STEM keyword is used to distinguish between the variables returned from each system.

```
  parse arg PARM

  address LINK "CEXRXENV INIT"

  address CEX "OPTION MSGLVL=QUIET"

/* Define each connection with a unique CONID */
  address CEX "CONNECT HOST=SRV1,PORT=4199,HWSID=HWSOPGS1,",
                      "PASSTIK=YES,APPLID=CEXTEST,",
                      "CONID=SYS1"                             1

  address CEX "CONNECT HOST=SRV2,PORT=4299,HWSID=HWSOPGS2,",
                      "PASSTIK=YES,APPLID=CEXTEST,",
                      "CONID=SYS2"                             2

/* Query command using CONID and unique STEM name */
  address CEX "QUERY TYPE=PENDING_RESPONSES,DSLIST=RLIST1,",   3
                      "CONID=SYS1,",
                      "STEM=GS1"

  address CEX "QUERY TYPE=PENDING_RESPONSES,DSLIST=RLIST1,",   4
                      "CONID=SYS2,",
                      "STEM=GS2"
```

```
/* Produce a table of the combined Query results */
  pendingResp = 0
  say '     Datastore table:'
  say '     System    Name      Status        Pending'
  say '     --------  --------  ------------  ---------'
  if GS1.RESPONSE.DST.0 > 0 then do                   5
    do i=1 to GS1.RESPONSE.DST.0
      say '     HWSOPGS1  '||,
                left(GS1.RESPONSE.DSTNAME.I,8)'  '||,
                left(GS1.RESPONSE.STATUS.I,12)'  '||,
                right(GS1.RESPONSE.PENDING.I,9)
      pendingResp = pendingResp + GS1.RESPONSE.PENDING.I
    end
  end
  if GS2.RESPONSE.DST.0 > 0 then do
    do i=1 to GS2.RESPONSE.DST.0
      say '     HWSOPGS2  '||,
                left(GS2.RESPONSE.DSTNAME.I,8)'  '||,
                left(GS2.RESPONSE.STATUS.I,12)'  '||,
                right(GS2.RESPONSE.PENDING.I,9)
      pendingResp = pendingResp + GS2.RESPONSE.PENDING.I
    end
  end
  say '     :  --------                          ---------'
  say '     :  Total                             'right(pendingResp,9)
  say '     :  --------                          ---------'

return
```

**Notes:**

1 The first connection is assigned CONID=SYS1.

2 The second connection is assigned CONID=SYS2.

3 Issue a host command to QUERY pending responses on the IMS Connect identified by CONID=SYS1. The data returned is stored in variables using the STEM=GS1.

4 Issue a host command to QUERY pending responses on the IMS Connect identified by CONID=SYS2. The data returned is stored in variables using the STEM=GS2.

5 Format and display the fields returned by the QUERY commands.

The report combines queries on two IMS Connects HWSOPGS1 and HWSOPGS2. STEM is used to distinguish between data returned from each system.

```
     Datastore table:
     System    Name      Status        Pending
     --------  --------  ------------  ---------
     HWSOPGS1  IMSA      Normal                7
     HWSOPGS1  IMSB      Normal                3
     HWSOPGS2  IMSA      Normal                4
     HWSOPGS2  IMSB      Normal                2
     --------                          ---------
     Total                                    16
     --------                          ---------
READY
END
```

## Sample library: REXX execs

The SCEXSAMP library includes several sample REXX execs. As well as performing useful operations tasks, these samples demonstrate how to develop your own execs using the CEX host command environment.

CEXRXCSJ in the SCEXSAMP library is a batch job to execute a specified REXX sample. The JCL is parameterized. Select one of the three sample commands, depending on your security validation method:

- PassTicket
- user ID and password
- no validation

Specify the name of the REXX exec, the connection details, and an optional list of parameters. The parameters are documented in the prolog of the REXX exec.

| Sample member | Description |
|---|---|
| CEXRXC01 | Issues the SWITCH command to switch the active journal for an IMS Connect system. |
| CEXRXC02 | Coordinated journal switch and IMS switch. |
| CEXRXC11 | Activate or deactivate OTMA trace. |
| CEXRXC21 | Drain a datastore by issuing the ROUTE ACTION=DRAIN command. DRAIN temporarily suspends routing of messages to a datastore, allowing it to be 'drained' of in-progress transactions. |
| CEXRXC22 | Drain all of the datastores in a routing list (single IMS Connect system). |
| CEXRXC23 | Drain all of the datastores in a routing list (two IMS Connect systems). In this sample the associated IMS system is shut down once the drain has completed. |
| CEXRXC26 | Drain a persistent session based on criteria. |
| CEXRXC31 | Set an OTMA routing plan or ODBM routing plan. |
| CEXRXC40 | Issue IMS Connect commands and IMS Type-1 commands using the SHELL host command. |
| CEXRXC42 | Query active sessions. |
| CEXRXC44 | Query ACEE cache. |
| CEXRXC45 | Query trace. |
| CEXRXC51 | Clear ACEE cache. |
| CEXRXC61 | Update datastore capacity weights. |
| CEXRXC62 | Update the event collection level for an IMS Connect system. |
| CEXRXC63 | Update the session message limit for an IMS Connect system. |
| CEXRXC71 | Add a datastore. |

# Chapter 12. Configuring command access in IMS Connect Extensions

IMS Connect Extensions can be used to issue IMS Connect, IMS Connect Extensions, and IMS Type-1 commands.

Commands can be issued from many places in IMS Connect Extensions including the command shell and line commands in the ISPF Operations dialog, the batch interface, and the Operations Console. These topics provide information on how to configure command access:

- How to activate a specified IMS Connect system to accept commands from IMS Connect Extensions.
- How to use a SAF-compliant external security manager (ESM) such as RACF to control users' access to issue IMS Connect and IMS Connect Extensions commands.
- How to use PassTickets to validate users when they issue a command request (instead of transmitting their password).
- How to configure IMS Connect ports to accept IMS Type-1 commands.

**Related tasks**:

"APF-authorizing load libraries and the MODIFY command" on page 34
This topic explains how to APF-authorize the IMS Connect Extensions load libraries and the MVS **MODIFY** command. These steps enable IMS Connect Extensions to work with IMS Connect and to issue commands to the IMS Connect address space.

## Activating command access to an IMS Connect system

For security reasons, command access to IMS Connect systems from IMS Connect Extensions is deactivated by default.

### Procedure

To activate command access for an IMS Connect system:

1. Start the IMS Connect Extensions ISPF dialog.
2. Select the definition repository used by the IMS Connect system for which you want to activate commands.
3. Select primary menu option 1.1 **System Definitions**.
4. Enter **S** next to the system name (HWSID).
5. Select **Activate Commands**. This activates access to IMS Connect WTOR commands and IMS Connect z/OS commands, IMS Type-1 commands, and IMS Connect Extensions commands for that IMS Connect system.
6. Optionally, select **Activate Access Control** to use an external security manager (ESM) either to authorize users or to support issuing of IMS Type-1 commands from the command shell. If you activate access control, you must set a security APPLID for the system. You must then perform additional steps to configure access control. See "Controlling access to commands in your external security manager" on page 130 for details.
7. Optionally, select **Enable PassTicket Generation** to use PassTickets instead of plaintext passwords. Use of PassTickets is recommended where there is a risk

of passwords being intercepted. If you select this option, you must perform additional steps. See "Enabling PassTicket processing" on page 133 for details.

8. Restart the IMS Connect system.

### What to do next

If you need to deactivate command shell access later, clear the **Activate Commands** option. Command access becomes disabled the next time you restart your system.

## Controlling access to commands in your external security manager

To control access to IMS Connect commands and IMS Connect Extensions commands, IMS Connect Extensions uses a similar security model to IMS Type-2 commands. Using this model you can control access to each command verb and keyword combination.

Security can be implemented for IMS Type-1 commands using RACF or the Command Authorization exit routine or both. See the sections on transaction command security and authorizing commands in *IMS System Administration*.

Security checking uses the RACF OPERCMDS class, and the resource name has the following form:

```
CEX.applid.verb.keyword
```

where:

**CEX** The IMS Connect Extensions product prefix.

**applid** The application ID that you specified for the IMS Connect system in IMS Connect Extensions. The *applid* can identify a group of systems or an individual system, depending on whether you used the same *applid* for multiple IMS Connect systems or a different *applid* for each IMS Connect system.

**verb** The name of the command verb. For example, **QRY** for *query*.

**keyword**
The name of the command keyword. For example, **DS** for *datastore*.

For example, to allow user ID Sandy to have access to **REFRESH** commands on all IMS Connect Extensions systems:

```
RDEFINE OPERCMDS CEX.*.REF.* UACC(NONE)
PERMIT CEX.*.REF.* CLASS(OPERCMDS) ID(SANDY) ACCESS(UPDATE)
```

### Resource names and access authority for commands

This table lists the IMS Connect and IMS Connect Extensions commands supported by IMS Connect Extensions along with their resource name and access authority.

See "Example: typical access control configuration" on page 132 for a sample RACF job to set these profiles.

*Table 6. Resource name and RACF authority for IMS Connect and IMS Connect Extensions commands*

| Command | Type | Verb | Object | Resource name | Authority |
|---|---|---|---|---|---|
| | CEX | ADD | Datastore | CEX.*applid*.ADD.DS | UPDATE |
| | CEX | ADD | User exit | CEX.*applid*.ADD.EXIT | UPDATE |

*Table 6. Resource name and RACF authority for IMS Connect and IMS Connect Extensions commands (continued)*

| Command | Type | Verb | Object | Resource name | Authority |
|---|---|---|---|---|---|
| | CEX | CLEAR | ACEE (user profile) | CEX.*applid*.CLR.ACEE | UPDATE |
| | CEX | DELETE | User exit | CEX.*applid*.DEL.EXIT | UPDATE |
| | CEX | DRAIN | Session | CEX.*applid*.DRN.SESSION | UPDATE |
| VIEWIA | HWS | QUERY | Alias | CEX.*applid*.QRY.ALIAS | READ |
| VIEWDS | HWS | QUERY | Datastore | CEX.*applid*.QRY.DS | READ |
| VIEWIP | HWS | QUERY | IMSplex | CEX.*applid*.QRY.IMSPLEX | READ |
| VIEWMSC | HWS | QUERY | MSC physical link | CEX.*applid*.QRY.MSC | READ |
| VIEWPORT | HWS | QUERY | Port | CEX.*applid*.QRY.PORT | READ |
| VIEWRMT | HWS | QUERY | Remote IMS Connect instance | CEX.*applid*.QRY.RMT | READ |
| VIEWHWS | HWS | QUERY | System | CEX.*applid*.QRY.SYSTEM | READ |
| VIEWUOR | HWS | QUERY | Unit of recovery (UOR) | CEX.*applid*.QRY.UOR | READ |
| | CEX | REFRESH | Affinity List | CEX.*applid*.REF.AFL | UPDATE |
| | CEX | REFRESH | Application | CEX.*applid*.REF.APPL | UPDATE |
| | HWS | REFRESH | XML converter | CEX.*applid*.REF.CONVERTR | UPDATE |
| | CEX | REFRESH | Datastore | CEX.*applid*.REF.DS | UPDATE |
| | CEX | REFRESH | Datastore group | CEX.*applid*.REF.DSG | UPDATE |
| | CEX | REFRESH | Journal | CEX.*applid*.REF.JOURNAL | UPDATE |
| | CEX | REFRESH | ODBM routing rules | CEX.*applid*.REF.ODBM | UPDATE |
| | HWS | REFRESH | Cached UIDs | CEX.*applid*.REF.RACFUID | UPDATE |
| | CEX | REFRESH | Routing rules | CEX.*applid*.REF.ROUTING | UPDATE |
| | CEX | REFRESH | Security (SAF Class rules) | CEX.*applid*.REF.SECURITY | UPDATE |
| | CEX | REFRESH | System definition | CEX.*applid*.REF.SYSTEM | UPDATE |
| | CEX | REFRESH | Transaction | CEX.*applid*.REF.TRAN | UPDATE |
| | CEX | RELOAD | User exit | CEX.*applid*.RLD.EXIT | UPDATE |
| | CEX | ROUTE | Datastore | CEX.*applid*.RTE.DS | UPDATE |
| | CEX | SET | Application | CEX.*applid*.SET.APPL | UPDATE |
| | CEX | SET | Datastore | CEX.*applid*.SET.DS | UPDATE |
| | CEX | SET | Datastore group | CEX.*applid*.SET.DSG | UPDATE |
| | CEX | SET | Routing plan | CEX.*applid*.SET.PLAN | UPDATE |
| | CEX | SET | Publisher | CEX.*applid*.SET.PUB | UPDATE |
| | CEX | SET | System definition | CEX.*applid*.SET.SYSTEM | UPDATE |
| | CEX | SET | Transaction | CEX.*applid*.SET.TRAN | UPDATE |
| STARTIA | HWS | START | Alias | CEX.*applid*.STA.ALIAS | UPDATE |
| STARTDS | HWS | START | Datastore | CEX.*applid*.STA.DS | UPDATE |
| | CEX | ENABLE | User exit | CEX.*applid*.STA.EXIT | UPDATE |
| STARTIP | HWS | START | IMSplex | CEX.*applid*.STA.IMSPLEX | UPDATE |
| STAMSC | HWS | START | MSC physical link | CEX.*applid*.STA.MSC | UPDATE |
| SETOAUTO ON | HWS | START | ODBM routing | CEX.*applid*.STA.OAUTO | UPDATE |
| STARTOD | HWS | START | ODBM | CEX.*applid*.STA.ODBM | UPDATE |
| STARTPT | HWS | START | Port | CEX.*applid*.STA.PORT | UPDATE |
| SETPWMC ON | HWS | START | Mixed-case passwords | CEX.*applid*.STA.PWMC | UPDATE |
| SETRACF ON | HWS | START | RACF | CEX.*applid*.STA.RACF | UPDATE |
| STARMT | HWS | START | Remote IMS Connect instance | CEX.*applid*.STA.RMT | UPDATE |
| SETRRS ON | HWS | START | RRS | CEX.*applid*.STA.RRS | UPDATE |
| | HWS | START | System | CEX.*applid*.STA.SYSTEM | UPDATE |
| SETUIDC ON | HWS | START | ACEE cache | CEX.*applid*.STA.UIDCACHE | UPDATE |
| STOPIA | HWS | STOP | Alias | CEX.*applid*.STO.ALIAS | UPDATE |
| STOPCLNT | HWS | CANCEL | Client session | CEX.*applid*.STO.CLIENT | UPDATE |
| STOPDS | HWS | STOP | Datastore | CEX.*applid*.STO.DS | UPDATE |
| | CEX | DISABLE | User exit | CEX.*applid*.STO.EXIT | UPDATE |
| STOPIP | HWS | STOP | IMSplex | CEX.*applid*.STO.IMSPLEX | UPDATE |
| STOLINK | HWS | STOP | MSC logical link | CEX.*applid*.STO.LINK | UPDATE |
| STOMSC | HWS | STOP | MSC physical link | CEX.*applid*.STO.MSC | UPDATE |

*Table 6. Resource name and RACF authority for IMS Connect and IMS Connect Extensions commands (continued)*

| Command | Type | Verb | Object | Resource name | Authority |
|---|---|---|---|---|---|
| SETAUTO OFF | HWS | STOP | ODBM routing | CEX.*applid*.STO.OAUTO | UPDATE |
| STOPOD | HWS | STOP | ODBM | CEX.*applid*.STO.ODBM | UPDATE |
| STOPPORT | HWS | STOP | Port | CEX.*applid*.STO.PORT | UPDATE |
| | CEX | STOP | Publisher client | CEX.*applid*.STO.PUB | UPDATE |
| SETPWMC OFF | HWS | STOP | Mixed-case passwords | CEX.*applid*.STO.PWMC | UPDATE |
| SETRACF OFF | HWS | STOP | RACF | CEX.*applid*.STO.RACF | UPDATE |
| STORMT | HWS | STOP | Remote IMS Connect instance | CEX.*applid*.STO.RMT | UPDATE |
| SETRRS OFF | HWS | STOP | RRS | CEX.*applid*.STO.RRS | UPDATE |
| STOSCLN | HWS | STOP | Send client | CEX.*applid*.STO.SCLN | UPDATE |
| CLOSEHWS | HWS | STOP | System | CEX.*applid*.STO.SYSTEM | UPDATE |
| SETUIDC OFF | HWS | STOP | ACEE cache | CEX.*applid*.STO.UIDCACHE | UPDATE |
| | CEX | SWITCH | Journal | CEX.*applid*.SWI.JOURNAL | UPDATE |
| | CEX | TRACE | Port on/off | CEX.*applid*.TRC.PORT | UPDATE |
| RECORDER | HWS | TRACE | Recorder on/off | CEX.*applid*.TRC.RECORDER | UPDATE |
| | CEX | UPDATE | Datastore | CEX.*applid*.UPD.DS | UPDATE |
| | CEX | UPDATE | System | CEX.*applid*.UPD.SYSTEM | UPDATE |

## Example: typical access control configuration

IMS Connect Extensions provides a sample job, CEXCMDS, that implements IMS Connect Extensions resource profiles in RACF.

The following is the SYSIN deck for this job:

```
  /* ------------------------ */
  /* Define Resource Profiles */
  /* ------------------------ */
RDEFINE OPERCMDS CEX.<applid>.**    UACC(NONE)
RDEFINE OPERCMDS CEX.<applid>.TRC.* UACC(NONE)
RDEFINE OPERCMDS CEX.<applid>.SWI.* UACC(NONE)
RDEFINE OPERCMDS CEX.<applid>.RLD.* UACC(NONE)
  /* ---------- */
  /* Add Groups */
  /* ---------- */
  /* Operator Group */
ADDGROUP CEXOPER
  /* Operations Supervisor Group */
ADDGROUP CEXOPSUP
  /* Administrator Group */
ADDGROUP CEXADMIN
  /* ------------ */
  /* Grant Access */
  /* ------------ */
  /* Operator Group */
  /* Can only issue query commands */
PERMIT CEX.<applid>.**    CLASS(OPERCMDS) ID(CEXOPER) ACCESS(READ)
  /* Operations Supervisor Group */
  /* Can issue query, trace, journal and exit reload commands */
PERMIT CEX.<applid>.**    CLASS(OPERCMDS) ID(CEXOPSUP) ACCESS(READ)
PERMIT CEX.<applid>.TRC.* CLASS(OPERCMDS) ID(CEXOPSUP) ACCESS(UPDATE)
PERMIT CEX.<applid>.SWI.* CLASS(OPERCMDS) ID(CEXOPSUP) ACCESS(UPDATE)
PERMIT CEX.<applid>.RLD.* CLASS(OPERCMDS) ID(CEXOPSUP) ACCESS(UPDATE)
  /* Administrator Group */
  /* Can issue all commands */
PERMIT CEX.<applid>.**    CLASS(OPERCMDS) ID(CEXADMIN) ACCESS(UPDATE)
PERMIT CEX.<applid>.TRC.* CLASS(OPERCMDS) ID(CEXADMIN) ACCESS(UPDATE)
PERMIT CEX.<applid>.SWI.* CLASS(OPERCMDS) ID(CEXADMIN) ACCESS(UPDATE)
PERMIT CEX.<applid>.RLD.* CLASS(OPERCMDS) ID(CEXADMIN) ACCESS(UPDATE)
```

```
  /* Connect Users */
CONNECT <user1> GROUP(CEXOPER)
CONNECT <user2> GROUP(CEXOPSUP)
CONNECT <user3> GROUP(CEXADMIN)
```

# Enabling PassTicket processing

These topics explain why you might choose to use PassTickets instead of passwords and how to enable PassTicket support.

By default, when you use IMS Connect Extensions commands, your password is transmitted in plaintext. If this transmission occurs within a secure environment then this may not present a liability. Use PassTickets instead of passwords, if you believe there is a risk of the password being intercepted.

If you enable PassTicket support then each time the supplied IMS Connect Extensions ISPF client sends a command request to IMS Connect, it includes a PassTicket to authenticate the user making the request. The external security manager (ESM) at your installation validates the PassTicket. If the PassTicket is valid, then the command request is accepted.

To allow this authentication to occur, you need to enable PassTickets validation in your ESM, and to APF-authorize the program libraries that create PassTickets.

## Enabling PassTicket validation (RACF example)

How you enable PassTicket processing depends on which ESM your environment uses. The procedure shown here is for RACF. Other ESMs (such as CA ACF2 and CA Top Secret) have similar facilities for PassTickets. For details, see the documentation for those products.

### Procedure

To enable RACF to validate incoming PassTickets:

1. Activate the RACF general resource class PTKTDATA:
   ```
   SETROPTS CLASSACT(PTKTDATA)
   SETROPTS RACLIST(PTKTDATA)
   ```
   The resource class PTKTDATA holds the encryption key used to create and validate PassTickets.
2. Define a profile for IMS Connect Extensions:
   ```
   TSO RDEFINE PTKTDATA applid SSIGNON(KEYMASKED(password-key))
   ```
   where:

   ***applid*** Application ID you want to use for commands for this IMS Connect system.

   ***password-key***
   16-byte hexadecimal secured signon application key. For example, ABA1A2A3AFA5A6A7A8.

### What to do next

After you define or change the profile, you need to refresh the class:

```
SETROPTS RACLIST(PTKTDATA) REFRESH
```

## Authorizing the PassTicket program libraries

The programs that the IMS Connect Extensions batch and ISPF clients use to create PassTickets are authorized programs; they must reside in an APF-authorized library. In addition, the program that creates PassTickets for the ISPF client must be authorized within TSO/E.

### Procedure

To enable PassTicket processing:

1. APF-authorize the IMS Connect Extensions load libraries. (Typically, you do not need to perform this step as it is part of the IMS Connect Extensions installation.)
2. Authorize the ISPF PassTicket program, CEXPTGEN, within TSO/E. Add the program name to the AUTHPGM list of authorized programs and the AUTHTSF list of programs that are authorized when invoked through the TSO/E service facility.

   For example, edit the SYS1.PARMLIB member IKJTSO*xx*, and add CEXPTGEN to the following lists:

   ```
   AUTHPGM NAMES(

   ⋮
   CEXPTGEN
   ⋮
   )
   ```

   and:

   ```
   AUTHTSF NAMES(
   ⋮
   CEXPTGEN
   ⋮
   )
   ```

# Configuring IMS Connect ports for IMS Type-1 commands

The command shell connects to an available IMS Connect message port to submit IMS Type-1 commands to IMS via message exit HWSJAVA0.

### About this task

To establish the communication path, IMS Connect Extensions uses the local host name to derive a network address. It then issues a socket connection request to the first available message port at that address.

To verify that IMS Connect Extensions is configured correctly to issue IMS Type-1 commands, ensure the following:

- IMS Connect has an open message port defined in the configuration file. SSL ports are not supported.
- IMS Connect is listening on the message port at the local host network address.
- There are no modifications to message exit HWSJAVA0 that will affect client message flows.

# Part 4. Extending transaction processing

These topics describe how to enhance IMS Connect clients, to implement routing of IRM messages and DRDA requests, and to improve transaction performance and reliability.

**Topics:**

# Chapter 13. Providing IMS Connect clients with additional services

This topic explains how to activate the client services exit (CEXSVC01). This exit enables you to extend the capabilities of IMS Connect clients.

## About this task

IMS Connect Extensions supplies a client services exit called CEXSVC01. The exit enables IMS Connect clients to perform password changes and receive session related information. Clients receive these services by sending an IMS request message (IRM) to IMS.

The exit intercepts messages with `*CEXSVC*` in the IRM header and responds to clients with a `*REQSTS*` RSM. The message is never forwarded to IMS.

For information on the IRM format the service exit accepts, see Chapter 30, “Client services exit,” on page 425.

## Procedure

1. To use the service exit, first add CEXSVC01 to your IMS Connect (HWS) configuration member:

   ```
   EXIT=(HWSSMPL0,HWSSMPL1,HWSSOAP1,CEXSVC01)
   ```
2. The next step is to enable the service exit in IMS Connect Extensions. From the IMS Connect Extensions ISPF dialog, select primary menu option 1.2 **User Exits**.
3. In the user exit list, check that CEXSVC01 appears. If it does not appear:
   a. Enter the **LOAD** primary command.
   b. Enter **S** next to CEXSVC01, and then press Enter.
4. Restart IMS Connect.
5. Confirm the exit has started successfully, by checking the CEXPRINT ddname in the IMS Connect for this message: `CEX5040I Exit CEXSVC01 loaded at <offset>, length <length>`
6. Finally, you can verify the installation of the Service Exit by using the IMS Connect Extensions installation verification program (IVP) to run a client session that requests who-am-I information. To run the IVP:
   a. Follow the instructions in CEXIVP5, located in the SCEXSAMP sample library, to customize the client job for your site.
   b. Submit the job.
   c. View the print output for the job to see if the client received a valid response.

# Chapter 14. Routing OTMA messages and balancing workloads

OTMA routing is the process where IMS Connect Extensions dynamically alters the target datastore of an IRM message.

IMS Connect directs messages between its distributed clients and IMS resources. Request messages are passed to the datastore specified in the header, and response messages are returned to the originating TCP/IP client.

Routing allows for another level of abstraction between IMS Connect TCP/IP clients and IMS applications and data. Instead of transactions being 'hard-wired' to specific destinations, workload can be dynamically redirected among pools of suitable datastores according to routing rules that you manage within IMS Connect Extensions. This level of abstraction gives you great flexibility in managing client workload without having to change either the client or the IMS systems.

## Benefits of routing

Routing can be implemented for any of the following purposes.

**Operational adaptability and transactional continuity**

IMS Connect Extensions host command environment for REXX and configuration options allow you to automate responses to changing datastore conditions.

IMS Connect Extensions can temporarily suspend routing of messages to a datastore, allowing it to be 'drained' of in-progress transactions. This can reduce the likelihood of transactions failing or being rejected when IMS is shut down. It can also automatically exclude from routing any datastores that are reporting a flood condition.

IMS Connect Extensions can automate the closing of TCP/IP sessions based upon transaction counts. In the event of the failure of an IMS Connect system, this enables the sysplex distributor to balance the number of sessions directed to each remaining live IMS Connect system.

IMS Connect Extensions supports routing of OTMA messages using dynamically added datastores without having to restart your IMS Connect instance.

Routing plans effectively allow you to test different configurations well in advance of making a change. Routing plans can be used to reconfigure topologies for planned outages or where changes in demand require different routing behaviors, for example peak and off-peak periods.

**Performance**

IMS Connect Extensions can optimize message processing performance by always selecting the local datastore (primary) for a given destination ID.

Rules-based routing allows you to more easily introduce parallelism in OTMA by employing multiple datastores that point to the same IMS system. This type of parallelism can significantly improve performance for transactions that can utilize multiple MPP regions.

**Datastore workload balancing**

IMS Connect Extensions can redirect incoming requests among datastores according to their processing capacity. Datastores with greater capacity receive more transactions. You specify the relative weighting of each datastore in the datastore definition.

**High availability**

When a datastore is not available, IMS Connect Extensions dynamically removes it from the candidate list and routes messages to other available datastores based on their capacity weights. The client does not need to alter the destination datastore manually.

IMS Connect Extensions can improve availability for all message types, including paired Send Only and Resume TPIPE.

### Advantages of OTMA rules-based routing

There are three OTMA routing methods: rules-based routing, transaction routing, and primary datastore routing. Rules-based routing is the most powerful and flexible OTMA routing method, and effectively supersedes the older, legacy methods. It has the following advantages:

- Routing rule configuration is simpler to implement, easier to understand, and easier to maintain than transaction routing.
- It is easier to see the routing rules that are in effect on a system. The routing rules that have been applied to a system are displayed in CEXPRINT messages at startup and after an OTMA Routing Rules REFRESH command is issued.
- You can associate sets of routing rules with a routing plan. It is then possible to completely reconfigure your rules by setting or changing the routing plan.
- Routing rules enable you to safely drain and suspend datastores or groups of datastores for the purposes of planned maintenance. Draining a datastore instead of closing the datastore or stopping the IMS system allows in-flight activity to complete. Once the in-flight activity has completed it is then possible to stop the IMS system without adversely impacting IMS Connect clients. IMS Connect Extensions provides a number of REXX samples that allow you to fully automate a drain and IMS shutdown across multiple datastores and multiple IMS Connect systems.
- It supports both synchronous and asynchronous callout.
- The CEXROUTE control option allows finer control of rules-based routing behavior and routing exceptions.

## OTMA rules-based routing

Rules-based routing specifies routing behavior based on the datastore identifier (IRM_IMSDestID) supplied in a message.

You can define rules relating to a given system or system group or to all systems. Routing lists define collections of datastores. The rules allow you to specify target and fallback routing lists for different message types. Routing rules that are intended to operate at the same time can be grouped under a routing plan. You can switch to a new routing plan by issuing a Set Routing Plan command (using a REXX exec, the ISPF dialog, or the Operations Console plug-in) and then refreshing the routing rules.

Rules-based routing supports the following rule structure:

```
For the specified system or group or for all systems:
   If DestID == string & Msg_Type == {a supported message type}:
      Select datastore from the target routing list
      If none of the datastores in the target routing list are available:
         Select datastore from the fallback routing list
```

**Note:**

1. If workload balancing is active, datastores are selected from within the routing list based on their capacity weight ratings.
2. Datastores must be defined to both IMS Connect and IMS Connect Extensions to be candidates for routing.

## Features of OTMA rules-based routing

IMS Connect Extensions uses the original datastore ID and message type to work through the routing rules that are in effect and generate a list of candidate datastores.

Rules-based routing works in the following ways:

- It uses the original datastore ID to find a matching routing rule and then to determine eligible datastore candidates.
- The original datastore ID does not have to be a physical datastore defined to the IMS Connect system. Instead, it can be a generic value that represents some workload grouping such as an application, business group, or cost center. IMS Connect Extensions locates the OTMA routing rule whose key field (Original Datastore) matches that generic string (IRM_IMSDestId) specified in the message. IMS Connect Extensions processes the routing rule and replaces the generic original destination ID with a real datastore name selected from the list of datastores referenced by the routing rule.
- Routing lists are used to define the list of candidate datastores. You cannot use affinity lists or datastore groups; those are specific to transaction routing.
- For each message type you can define a target routing list and an optional fallback routing list. The fallback list contains reserve datastores that can be used if the preferred datastores in the target list are all unavailable or under load.
- It allows you to configure different candidates for different message types. For example, you could configure one list of candidates for transactional (send-receive) messages and another list of candidates for asynchronous messages.
- General routing rules can be fine-tuned by adding conditions for particular transactions. To do this, you have a master OTMA routing rule for each destination ID, and specify additional qualifying rules that route certain transaction codes to a different list of target datastores. You can specify a qualifying rule without a matching master rule in the repository. If a master rule does not already exist for that DestID, an implied routing rule will be generated internally at run time.
- IMS Connect Extensions checks for the name of the active OTMA routing plan, if there is one. Only one OTMA routing plan can be active on a system. Routing rules that are assigned to other routing plans are not considered.

  A routing rule that is not assigned to any routing plan is always in effect. Where there is a conflict between an unassigned routing rule and a routing rule that is explicitly assigned to the routing plan that is currently active, the rule that is assigned to the current routing plan takes precedence. In other words, you can define unconditional rules to specify generic routing behavior and then override that behavior using specific rules that are assigned to the current routing plan.

IMS Connect Extensions allows for a lot of flexibility in how you implement routing rules. The simplest configuration is to implement the same set of rules across all your systems. Alternatively you can implement a different set of rules for each group of systems or even for a single system. Finally, you can combine these methods. For example, you can create an "All systems" rule but then override the routing behavior for a particular message type on a given system.

Specific rules take precedence over more general rules. That is:

- A qualifying rule will override an explicit or implicit master rule.
- A rule that is specific to an IMS Connect system will override a rule with the same condition and message type that is specified for a group or for all systems
- A rule specified for a group will override a rule that applies to all systems.

## Sample OTMA routing rule descriptions

When a system starts that has OTMA routing rules activated, a series of messages is written to CEXPRINT. These messages summarise the routing rules, routing lists, and routing plan that are in effect.

```
07.07.47 CEX5091I ---Begin OTMA routing rules descriptions---, refresh
                  index=1---
07.07.47 CEX5091I OTMA Routing plan is WEEKENDS 1
07.07.47 CEX5091I OTMA routing rule for Destid=DEPS
07.07.47 CEX5091I Message             --Routing  Lists--
07.07.47 CEX5091I type      Source    Target    Fallback
07.07.47 CEX5091I --------  --------  --------  --------
07.07.47 CEX5091I TRANSACT  PLNRULEW  RLPLAW
07.07.47 CEX5091I SENDONLY  PLNRULEW  RLPLAW
07.07.47 CEX5091I RTPIPE    PLNRULEW  RLPLAW
07.07.47 CEX5091I SYNCREQ   PLNRULEW  RLPLAW
07.07.47 CEX5091I SYNCRESP  PLNRULEW  RLPLAW
07.07.47 CEX5091I End of entries for OTMA routing rule
07.07.47 CEX5091I OTMA routing rule for Destid=PROD 2
07.07.47 CEX5091I Message             --Routing  Lists--
07.07.47 CEX5091I type      Source    Target    Fallback
07.07.47 CEX5091I --------  --------  --------  --------
07.07.47 CEX5091I TRANSACT  OTMARULE  RLIST1    RLIST2
07.07.47 CEX5091I SENDONLY  OTMARULE  RLIST1    RLIST2
07.07.47 CEX5091I RTPIPE    OTMARULE  RLIST1    RLIST2
07.07.47 CEX5091I SYNCREQ   OTMARULE  RLIST1    RLIST2
07.07.47 CEX5091I SYNCRESP  OTMARULE  RLIST1    RLIST2
07.07.47 CEX5091I End of entries for OTMA routing rule
07.07.47 CEX5091I Implied OTMA routing rule for Destid=PRODJLM 3
07.07.47 CEX5091I Message             --Routing  Lists--
07.07.47 CEX5091I type      Source    Target    Fallback
07.07.47 CEX5091I --------  --------  --------  --------
07.07.47 CEX5091I There are no active entries
07.07.47 CEX5091I OTMA routing rule for Destid=PRODJLM 4
07.07.47 CEX5091I   Rule qualifier list=JLMTRANS
07.07.47 CEX5091I     Transaction qualifier=JLM*
07.07.47 CEX5091I Message             --Routing  Lists--
07.07.47 CEX5091I type      Source    Target    Fallback
07.07.47 CEX5091I --------  --------  --------  --------
07.07.47 CEX5091I TRANSACT  JLMRULE   RLIST1
07.07.47 CEX5091I SENDONLY  JLMRULE   RLIST1
07.07.47 CEX5091I End of entries for OTMA routing rule
07.07.47 CEX5091I ---End OTMA routing rules descriptions---
07.07.47 CEX5093I ---Begin OTMA routing list description---
07.07.47 CEX5093I Routing             Weight
07.07.47 CEX5093I lists     Datastore rating
07.07.47 CEX5093I --------  --------  ------
07.07.47 CEX5093I RLIST1 5   IMSA          1
07.07.47 CEX5093I           IMSB         1
07.07.47 CEX5093I RLIST2    IMSC        10
07.07.47 CEX5093I           IMSD        30
07.07.47 CEX5093I RLPLAW    IMSA         1
07.07.47 CEX5093I           IMSB         1
07.07.47 CEX5093I ---End routing list description---
```

*Figure 59. Sample CEXPRINT messages showing OTMA routing descriptions*

This describes how the routing rules are configured on this system. For each message type it shows the source or name of the definition it has determined the rule from, and the target and fallback routing lists that the rule will use. Only eligible datastores for this system are displayed for each routing list.

Note that many more OTMA routing plans, routing rules, routing lists, and datastores are likely to be defined in the repository. The OTMA routing rules descriptions that are written to CEXPRINT show the in-memory rules and objects that are currently in effect.

**1** WEEKENDS is the active OTMA routing plan on this system. The OTMA routing rules description block displays the routing rules that are active because they are either assigned to the active routing plan or unassigned.

**2** This routing rule for Destid=PROD is named OTMARULE. It can potentially route transactions to datastores in routing lists named RLIST1 and RLIST2.

**3, 4** A qualifying routing rule is defined for a subset of transactions addressed to Destid=PRODJLM ( **4** ). Because no master routing rule is already defined for Destid=PRODJLM, an implied rule was generated ( **3** ).

**5** The qualifying routing rule for Destid=PRODJLM refers to the OTMA routing list RLIST1 as a target. The attributes of RLIST1 and other routing lists that are referred to by active OTMA routing rules are displayed in the OTMA routing list description block.

## Implementing OTMA rules-based routing

Implementing OTMA rules-based routing involves configuring routing plans, routing lists (lists of potential datastore candidates), routing rules, and optional transaction lists, and then activating OTMA routing on required systems.

### Procedure

1. Analyze the message traffic in your organization to find out what datastore identifiers (DestIDs) and message types you need to cater for. Decide if you can apply the same set of rules across all systems, or if you want to organize systems into groups and implement a different set of rules for each system group.
2. If you want different sets of routing rules to take effect at different times, define routing plans (see "Defining routing plans" on page 229). When you define routing rules you can assign them to a routing plan.
3. Define OTMA routing lists. These specify lists of potential datastore candidates. Only datastores defined to a system are eligible routing candidates.
4. Define optional transaction lists. You can use a transaction list in an OTMA routing rule to route selected transactions to a different routing list from the one specified in the master routing rule.
5. Define OTMA routing rules for each DestID string you intend to route (see "Defining OTMA routing rules" on page 219). Each routing rule maps messages to target and fallback routing lists based on the message type and the IMS Connect system processing the message.
6. Review the INELIGIBLEIF, RBR_NODEST, and RBR_FAILURE keywords on the CEXROUTE control option to confirm whether the default behavior matches your requirements.

### What to do next

Rules-based routing can be enabled on a per-system basis. To toggle OTMA rules-based routing, select or clear **Activate OTMA rules** in the Systems Definition panel.

To activate a routing plan using the Operations dialog, first use the **Routing Plan** option in the Set Commands panel to set the plan name, and then use the **OTMA routing rules** option in the Refresh Commands panel to load the new set of rules.

**Tip:** You can also activate a routing plan from a REXX exec using the SET PLAN and REFRESH host commands.

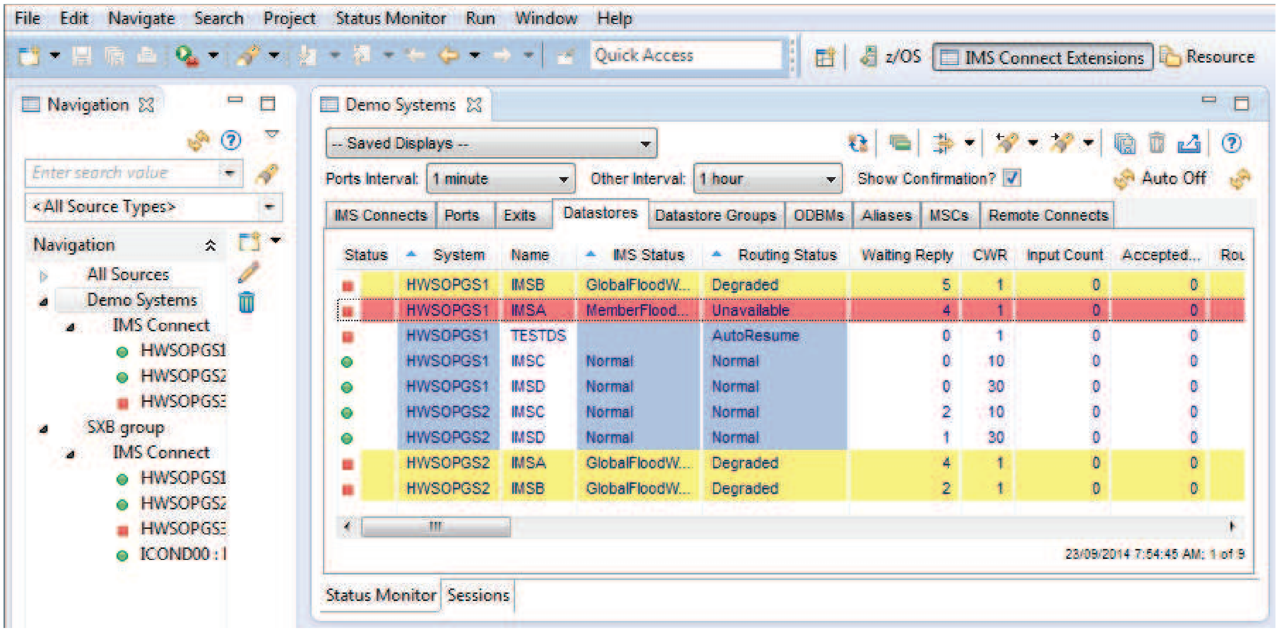
## Rules-based routing: flood-control processing

Rules-based routing includes logic designed to avoid individual datastores being flooded with OTMA messages.

For OTMA rules-based routing, the following flood-control processing is performed in sequence:

1. The Target list is checked for an available datastore. In this first pass, datastores in Degraded (flood warning) state are always excluded as candidates.
2. If no available datastore is found in the Target list, an attempt is made to find an available datastore in the Fallback list.
3. If no Fallback list is present or no available candidates can be found in the Fallback list, the Target list is searched again for available datastores, this time including those in Degraded state.
4. If no candidates can be found in the Target list, the Fallback list, if present, is searched for an available datastore including those in Degraded state.

**Note:** If the CEXROUTE INELIGIBLEIF=GLOBALFLOODWARNING parameter is set in the CEXCTLIN data set, any datastore that is in Degraded state is treated as Unavailable and therefore cannot be a candidate for routing.



Ports Interval: 1 minute
Other Interval: 1 hour
Show Confirmation? ☑

| Status | System | Name | IMS Status | Routing Status | Waiting Reply | CWR | Input Count | Accepted... | Rou |
|---|---|---|---|---|---|---|---|---|---|
| ■ | HWSOPGS1 | IMSB | GlobalFloodW... | Degraded | 5 | 1 | 0 | 0 | |
| ■ | HWSOPGS1 | IMSA | MemberFlood... | Unavailable | 4 | 1 | 0 | 0 | |
| ■ | HWSOPGS1 | TESTDS | | AutoResume | 0 | 1 | 0 | 0 | |
| ● | HWSOPGS1 | IMSC | Normal | Normal | 0 | 10 | 0 | 0 | |
| ● | HWSOPGS1 | IMSD | Normal | Normal | 0 | 30 | 0 | 0 | |
| ● | HWSOPGS2 | IMSC | Normal | Normal | 2 | 10 | 0 | 0 | |
| ● | HWSOPGS2 | IMSD | Normal | Normal | 1 | 30 | 0 | 0 | |
| ■ | HWSOPGS2 | IMSA | GlobalFloodW... | Degraded | 4 | 1 | 0 | 0 | |
| ■ | HWSOPGS2 | IMSB | GlobalFloodW... | Degraded | 2 | 1 | 0 | 0 | |

23/09/2014 7:54:45 AM: 1 of 9

*Figure 60. Status Monitor in Operations Console highlights datastores in Degraded and Unavailable state*

### Using OTMA Global Flood Warning support to respond to message flood conditions

OTMA monitors IMS system resources that are used to process OTMA transactions. To avoid a buildup of messages that could result in a message flood condition, OTMA clients are notified when the number of messages in the IMS system has reached a maximum allowable number.

If an OTMA client receives a notification that the IMS system is not processing OTMA messages normally, the OTMA client can then take appropriate action, such as rerouting OTMA transaction messages to a different IMS system.

The overall state of the IMS system is categorized by using the following states.

**Normal state (X'03')**
IMS is available and is processing OTMA messages normally.

**Degraded state (X'02')**
The number of messages in the IMS system has reached 80% of the maximum number defined either for an OTMA client or for all OTMA clients. A X'2D' IMS Connect event is produced with a status of Degraded.

**Unavailable state (X'01')**
The number of messages in the IMS system has exceeded the maximum number defined either for an OTMA client or for all OTMA clients. OTMA rejects all subsequent input messages from the OTMA client until the message flood condition is resolved. A X'2D' IMS Connect event is produced with a status of Unavailable.

Reporting of degraded service for OTMA was enhanced in IMS V11 to include global-level reporting. This new type of reporting is called global reporting because it is triggered by the total number of unprocessed messages across all OTMA transaction members (tmembers) for an IMS system. The issue is that this support gives an IMS Connect datastore a status of Degraded with a Global Flood Warning condition when the requirements are met, but unlike the Member level of reporting, a status of Unavailable for the Global Flood Severe condition is never reported. For member reporting, it is permissible to send messages to a datastore that is in Member Flood Warning condition. IMS Connect Extensions will send messages to a datastore in Flood Warning status if that is the only option available. If the datastore changes to the Member Flood Severe condition, IMS Connect Extensions will not send messages to the datastore. Since the condition of Global Flood Severe is never obtained, IMS Connect Extensions may continue to send messages to a datastore that has a Global Flood Warning condition. To do so may subject the IMS system to an abend due to out of storage conditions.

If you wish IMS Connect Extensions to take the necessary actions to protect the target IMS systems, you must be using some form of CEX routing and you must specify the INELIGILBLEIF=GLOBALFLOODWARNING parameter on the CEXROUTE command in the CEXCTLIN data set. When this parameter is specified, IMS Connect Extensions will treat a datastore with a Global Flood Warning condition as if it had a Global Flood Severe condition and stop sending messages to the datastore. If the Global Flood Warning condition results in no available datastores, IMS Connect Extensions routing will reject the message with an RSM indicating that there are no available datastores. This is done because if the message is returned to IMS Connect, it will be sent to the target datastore and might compromise the availability of the IMS system.

## Enabling message flood detection in IMS Connect

To enable message flood detection for an individual OTMA client, specify the maximum allowable number of input messages from the OTMA client in one or more of the following places:

- The INPT= parameter of the OTMA client descriptor for the OTMA client
- The INPUT parameter of the /START TMEMBER command
- The client-bid request from the OTMA client in the two-byte field at byte 62 of the client-bid command message

To enable message flood detection globally for OTMA clients combined, use the following type-1 command:

```
/START TMEMBER ALL INPUT nmbr
```

Values specified in the OTMA client descriptor can be overridden by the /START TMEMBER command. If the client-bid request specifies a smaller number, the client-bid request can also override the value specified in the OTMA client descriptor.

For more information about message flood detection, see the section on “Enabling and using OTMA” in *IMS Communications and Connections*.

## Supported message types

IMS Connect Extensions converts the DestID string in a transaction to a collection of datastores that you specify. How particular messages are handled by rules-based routing depends on the request type.

Here are the supported request types:

**Send-Receive Transactions**
Standard IMS Connect transactions.

This message type can also be used to route IMS commands. CEXROUTE IMSCMDROUTE=YES must be specified in the control input data set.

**Send-Only requests**
IMS Connect Send Only requests.

**Resume TPIPE (RTPIPE) requests**
Resume TPIPE requests do not have a transaction code, and may be paired with Send Only messages. If you are pairing Send Only with Resume TPIPEs, use the same collection of target and fallback datastores for both message types and ensure all datastores are part of the same shared queue. If the messages can be routed to more than one IMS Connect system, then all IMS Connect systems that will handle a Send Only with Resume TPIPE transaction must refer to the same Super Member.

**Note:** Routing Resume TPIPE requires an OTMA Super Member and shared queues.

**Synchronous callout Resume TPIPE requests**
Routing for Resume TPIPE messages marked as synchronous callout requests.

**Synchronous callout responses**
Routing for synchronous callout responses.

## Pre-routing user exit

You can use the pre-routing exit to influence the selection of datastores as candidates for OTMA rules-based routing.

IMS Connect can determine whether a datastore and a target IMS system are available for routing, but it cannot tell whether a particular transaction is available on a specific IMS system.

If you have processes that can determine the status of a transaction on an IMS system, you can use the pre-routing exit to influence which IMS systems are to be candidates for routing. You pass a list of candidate tmembers to the exit. (Because IMS Connect maintains its knowledge of IMS systems using the XCF tmember name, all references to IMS systems are by XCF tmember name.) When a tmember is disqualified, a reason code can be returned to the client explaining why.

## When the pre-routing exit is called

The exit is only called when there is a routing decision to be made. It is not called for cases such as ACK responses or IMS conversation integrations which must be returned to the same IMS system as the original message or conversation.

It is important to understand that this is a pre-routing exit and at this point the list of destinations has not yet been verified as actually being available. The exit is called under the following circumstances:

- The exit is not called unless a valid IMS transaction code has been found.
- The exit is not called if the resulting tmember list does not contain any members.
- The exit is only called for the following message types:
  - Transaction
  - Send-Only or Send-Only with Ack
- The routing candidates for a specific routing case have been determined using a specific candidate list, which is normally the target list.
- The exit can be called multiple times for a transaction.
  - Once for each target list.
  - Once for the fallback list if there is one, and only if the fallback list is actually used.

  Only the last call to the exit is used to determine if a reason code has been returned.

## Parameter list

When the exit is called it is passed a parameter list in general register 1. The actual contents of the list are shown in macro CEXRBEPL. The following items are expected to be present in the list, either within the list or by address pointer references:

- Call function:
  - INIT. The exit is always loaded during IMS Connect Extensions initialization. A dummy exit is provided that when called for initialization, returns that the exit is to be inactive.
  - TERM. Termination.
  - ROUT. Pre-routing processing. The exit is called when rules-based routing has determined that an input message is going to be routed but before the routing decision is made.
- Pointer to a 1024-byte work area.
- Original DestID.
- IMS transaction code.
- Alternate transaction code, if one is present.
- An 8-byte field containing the reason code for OTMA client return.
- Pointer to a list of candidate tmembers.
- Flag byte containing information useful to the exit, including flag bits for the following cases:
  - Client has provided OTMA headers.

  - Exit has returned a reason code matching the client type.
- Flag byte containing a message type identifier such as Transaction or Send-Only.
- Pointer to the first application segment.

## Handling the “No valid datastores” case

Consider the case where all datastores in the target and fallback lists could be excluded based upon the target IMS systems. If this is possible, you could do one of the following:

- Ensure that the original DestID that is sent to IMS Connect is not a valid datastore. That way if no datastores are selected and the original datastore is used as the final datastore target, it will fail due to missing datastore.
- Use the CEXROUTE control option in the CEXCTLIN data set to request that the message be rejected if there are no datastores selected by routing. The command format is:

  ```
  CEXROUTE     RBR_FAILURE=REJECT
  ```

  If the client has provided OTMA headers, return an 8-character return code that will be stored in the OMUSR_RESCODE field of the returned response.

  If the client did not provide OTMA headers, the reason code is a 4-byte hexadecimal number that, when converted to decimal, represents the reason code to be included in the RSM sent by IMS Connect Extensions.

## Exit requirements and specifications

The pre-routing exit must adhere to the following standards:

- The exit is called in Key 7 and might also be in supervisor state.
- Link the exit as reentrant and AMODE/RMODE 31.
- The exit must not issue an OS WAIT or do anything that causes the TCB to be suspended. (This is for performance reasons, as the exit is in the message input path.)
- The exit is not protected by an ESTAE when called.
- The exit must execute in TCB mode only.
- The exit does not use private PC instructions.
- The exit cannot change the message or application data in any manner.

## Return codes

A message can influence the selection of a target IMS system by checking the return code:

| RC | Function | Meaning |
|---|---|---|
| 0 | INIT | The exit has initialized and wishes to receive pre-routing requests. |
| 0 | TERM | The exit has completed termination and released all resources. |
| 0 | ROUT | All of the tmember names provided to the exit are acceptable. |

| RC | Function | Meaning |
|---|---|---|
| 4 | ROUT | All of the tmember names are acceptable; do not call the exit again for this specific transaction.<br>If there are no restrictions on the transaction, this might be a better return code than 0 so that if the fallback list is used the exit will not be called again.<br>**Important:** There is a slight but unavoidable risk that a transaction could be marked as restricted to a specific IMS system during the routing process and would not be detected. |
| 8 | INIT | The exit does not wish to receive pre-routing calls. |
| 8 | ROUT | At least one of the tmember names provided to the exit is not to be a candidate. |
| 16 | INIT | The exit encountered issues during initialization and does not wish to receive pre-routing calls.<br>Return and reason codes will be reported in a CEX5620E message. |
| 16 | TERM | The exit encountered issues during termination.<br>Return and reason codes will be reported in a CEX5620E message. |
| 16 | ROUT | The exit encountered issues during pre-routing processing and is to be deactivated until IMS Connect is restarted.<br>Return and reason codes will be reported in a CEX5620E message. |

### Example

The CEXRTGEX member in the SCEXSAMP library is a skeleton pre-routing exit. If called it returns RC=8 for the INIT call and RC=16 for any other call type. You can use it as the basis for your own pre-routing exit.

## Suspending routing to datastores: Drain/Resume

The Drain/Resume feature temporarily suspends routing of messages to a datastore, allowing it to be 'drained' of in-progress transactions. This can reduce the likelihood of transactions failing or being rejected when IMS is shut down.

When IMS is shut down using a /CHE command, it can result in IMS Connect transactions being sent to the IMS system that is being stopped. The problem is that IMS does not notify IMS Connect that the IMS system is unavailable until later in the termination process. IMS Connect might not recognize that IMS is not really available until IMS disconnects from the XCF group. It can take several seconds before IMS disconnects from the XCF group. Many transactions might have been routed to the IMS system that will not be processed or be rejected by IMS, resulting in timeouts or RSM messages for the remote client.

If the process is to shut down IMS using the /CHE command, a STOP datastore command could be issued first, but that might result in transaction responses being trapped within the IMS system as they would not have the datastore path available for them to be returned to IMS Connect. This case can also result in the remote client getting a time-out response to the stranded IMS response. What is needed is a way to ‘drain’ a datastore pipe before issuing the /CHE command.

The Drain/Resume feature allows the routing status of one or more datastores to be changed. The Drain process works for any form of OTMA routing in IMS

Connect Extensions, as the Suspended status applies to the selection of routing candidates only. Note that the Drain/Resume feature only affects routing done by IMS Connect Extensions. It has no effect on routing that might be done by the IMS Connect message exit.

The ROUTE host command has two modes:

- `ROUTE ACTION=DRAIN` changes the routing status of the datastore in IMS Connect Extensions from Normal to Suspended and withdraws it as a candidate for routing. The optional keyword AUTORESUME indicates that the datastore is to be automatically resumed when an IMS Datastore Available event record (X'10') is received.
- `ROUTE ACTION=RESUME` changes the routing status of the datastore from Suspended to Normal and restores it as a candidate for routing.

You can use the QUERY host command to help automate the process needed to safely shut down an IMSsystem without potential loss of inbound or outbound messages. `QUERY TYPE=PENDING_RESPONSES` returns the number of messages that are pending a response from IMS (that is, messages that are in a Waiting for Datastore state).

The Drain and Resume actions can also be initiated from the Status Monitor or Datastore Monitor dialogs. The Waiting Reply column contains the count of pending responses for each datastore.

**Normal**
The datastore is potentially a candidate for routing.

**NormalWLB**
The datastore is potentially a candidate for routing, and workload balancing is active.

**Degraded**
The datastore might be a candidate for routing but only if there are no other candidates in Normal state. This could be due to a warning-level IMS status.

**Unavailable**
The datastore is not a candidate for routing. This could be due to a severe-level IMS status.

**SusCWRZero**
The datastore has been suspended as a candidate for routing. Workload balancing is active and the datastore has a capacity weight ratio of zero.

**Suspended**
The datastore has been suspended as a candidate for routing. The datastore has been drained without the autoresume option. The datastore might be stopped - see ICON Status.

**SusAutoRes**
The datastore has been suspended as a candidate for routing. It will autoresume when the datastore has been stopped and then becomes active again.

**AutoRes**
The datastore is stopped but if it is started it will automatically become a candidate for routing.

## Implementing Drain/Resume example: IMS /CHE case

When it is intended to stop an IMS system using the /CHE command, you could manually perform the following actions in the Operations Console:

1. Use the Status Monitor to view the datastores across your plex.
2. Drain the datastores associated with the IMS system you wish to take offline.
3. Use the Active Session summary keyed on datastore to monitor existing connections until they have all completed.

   The Waiting Reply column contains the count of pending responses for each datastore.
4. Use the Status Monitor to stop the datastores and then issue the /CHE command.

You could use automated operations to achieve the same result.

1. Issue ROUTE commands to DRAIN all datastores associated with the IMS system that is being shut down.
2. Use QUERY TYPE=PENDING_RESPONSES against each datastore. When the number of pending responses falls to zero, use SHELL to issue the UPDATE DATASTORE IMS Connect command to stop the datastore.
3. When it is apparent that no transaction responses are left on the IMS system to be stopped, issue the /CHE command to stop the IMS system.
4. Use automatic operations to detect that the datastores have been disconnected from IMS Connect.
5. Issue ROUTE commands to RESUME all of the stopped datastores that are associated with the IMS system, so that they will be routing candidates when IMS restarts and joins the XCF group.

   **Tip:** An alternative is to specify AUTORESUME on the ROUTE ACTION=DRAIN command.

## Sample REXX execs:

See the following members in the SCEXSAMP library for detailed working examples of REXX execs that demonstrate these options:

**CEXRXC21**

1. Drain a datastore by issuing the ROUTE ACTION=DRAIN command.
2. Use QUERY TYPE=PENDING_RESPONSES to check the number of pending responses.

   If the timeout value is exceeded and there are still in-progress transactions then a WTO message is issued.
3. Use SHELL to issue the UPDATE DATASTORE command.

**CEXRXC22**

Drain all of the datastores in a routing list (single IMS Connect system). This exec is an extension of CEXRXC21.

**CEXRXC23**

Drain all of the datastores in a routing list (two IMS Connect systems). This exec is an extension of CEXRXC22.

The datastores are monitored until there are no in-progress transactions remaining. At this point a /CHE FREEZE command is issued against the IMS system.

## Dynamically adding datastores as targets for routing

IMS Connect Extensions supports rules-based routing of OTMA messages to dynamically added datastores without restarting IMS Connect.

### Before you begin

If you plan to add a datastore dynamically to IMS Connect, it is preferable to add the datastore definition in IMS Connect Extensions *first*. This allows it to be used in OTMA rules-based routing and viewed in the Status Monitor immediately.

### About this task

There are two scenarios in which a datastore might be defined to IMS Connect but not available for OTMA routing in IMS Connect Extensions:

1. The datastore was specified in the IMS Connect configuration member. IMS Connect and IMS Connect Extensions were restarted, but the datastore has not been defined in the IMS Connect Extensions repository.

   In this scenario, add the repository definition and then use the ADD DS host command to add the datastore to the in-memory datastore table. The member CEXRXC71 in the SCEXSAMP library contains a detailed example based on the ADD DS host command.
2. The datastore was dynamically added to IMS Connect. IMS V13 introduced a Type-2 command that allows you to add IMS Connect datastore definitions during runtime processing without having to restart your IMS Connect instance. The general format of the Type-2 command is:

   ```
   CREATE IMSCON TYPE(DATASTORE) keywords
   ```

IMS Connect Extensions does not automatically add a datastore definition when it detects that a new datastore has been created in IMS Connect. However, you can predefine a datastore definition in IMS Connect Extensions before creating the datastore itself using the CREATE IMSCON Type-2 command. This then allows you to refresh the in-memory routing rules to use the new datastore for OTMA rules-based routing.

The amount of room reserved for dynamically adding entries to the table is controlled by the CEXCTLIN option AUTOADD_DATASTORE. Reserve enough space to add the maximum number of datastores needed for the life of the IMS Connect region.

### Procedure

1. Add the datastore definition in the IMS Connect Extensions repository.
2. Issue the CREATE IMSCON IMS Type-2 command to dynamically add the datastore to IMS Connect.
3. Add this datastore definition to one or more OTMA routing lists in IMS Connect Extensions.
4. Refresh the routing rules using the Refresh Commands dialog or REFRESH host command in IMS Connect Extensions.

### What to do next

Messages can now be routed to the datastore. It is now visible in the Datastore Monitor and Status Monitor so that you can view its status and activity.

**Related reference**:

"ADD" on page 248
The **ADD** command adds an entry to the in-memory datastore table for a datastore that is defined to IMS Connect but was not defined to IMS Connect Extensions at startup.

"REFRESH" on page 259
The **REFRESH** command rebuilds the in-memory copy of the selected definitions for the target IMS Connect system. IMS Connect Extensions reads the definitions from the repository and rebuilds the in-memory values.

## Balancing workloads based on processing capacity of datastores

Workload balancing allows IMS Connect Extensions to redirect incoming transaction requests among IMS datastores according to their capacity, thereby balancing the transaction workload across multiple IMS systems and ensuring both availability and responsiveness for the remote client.

Workload balancing is accomplished using a weighted rotate algorithm that considers the processing capacity of a datastore. Datastores with greater capacity receive more transactions. You can specify the relative weighting of each datastore during the datastore definition process. Basic rotation, in which each datastore is deemed to have the same capacity, can be achieved by assigning each datastore an equal weighting value, or by deactivating workload balancing. You can keep a datastore in reserve by assigning it a capacity of zero, making it ineligible as a routing candidate.

IMS Connect Extensions will not route messages to datastores that are unavailable or which have a capacity of zero, and will instead use an eligible candidate regardless of whether workload balancing is active. If a datastore is no longer available, the candidate list is automatically rebalanced across the remaining candidate datastores.

Workload balancing takes effect when the Activate Workload Balancing option is selected in the System Definition and there is more than one eligible candidate for routing. Workload balancing can be activated for both rules-based routing and transaction routing.

Datastore capacities can be dynamically changed by IMS Connect Extensions clients. This is useful for redistributing workload in response to changing periods of demand. The member CEXRXC61 in the SCEXSAMP library shows how to make these changes using a REXX exec.

## Legacy routing methods

| Transaction routing and primary datastore routing are legacy OTMA routing
| methods. They have effectively been superseded by rules-based routing.

### Transaction routing

Transaction routing is the process by which IMS Connect Extensions selects a destination datastore for an incoming IRM message based on the transaction code and destination ID specified by the message.

You can use transaction routing to ensure that certain IMS Connect systems only use specific datastores and no others, regardless of the destination ID specified on the incoming messages. This can be useful for security and system testing.

Rules-based routing takes precedence over transaction routing. If rules-based routing is active and the DestID specified in the message matches a rule, then that rule will be used (see "OTMA rules-based routing" on page 140). If rules-based routing is not active or if no match can be made on a rule, the message follows the process for transaction routing, as shown in the following figure:

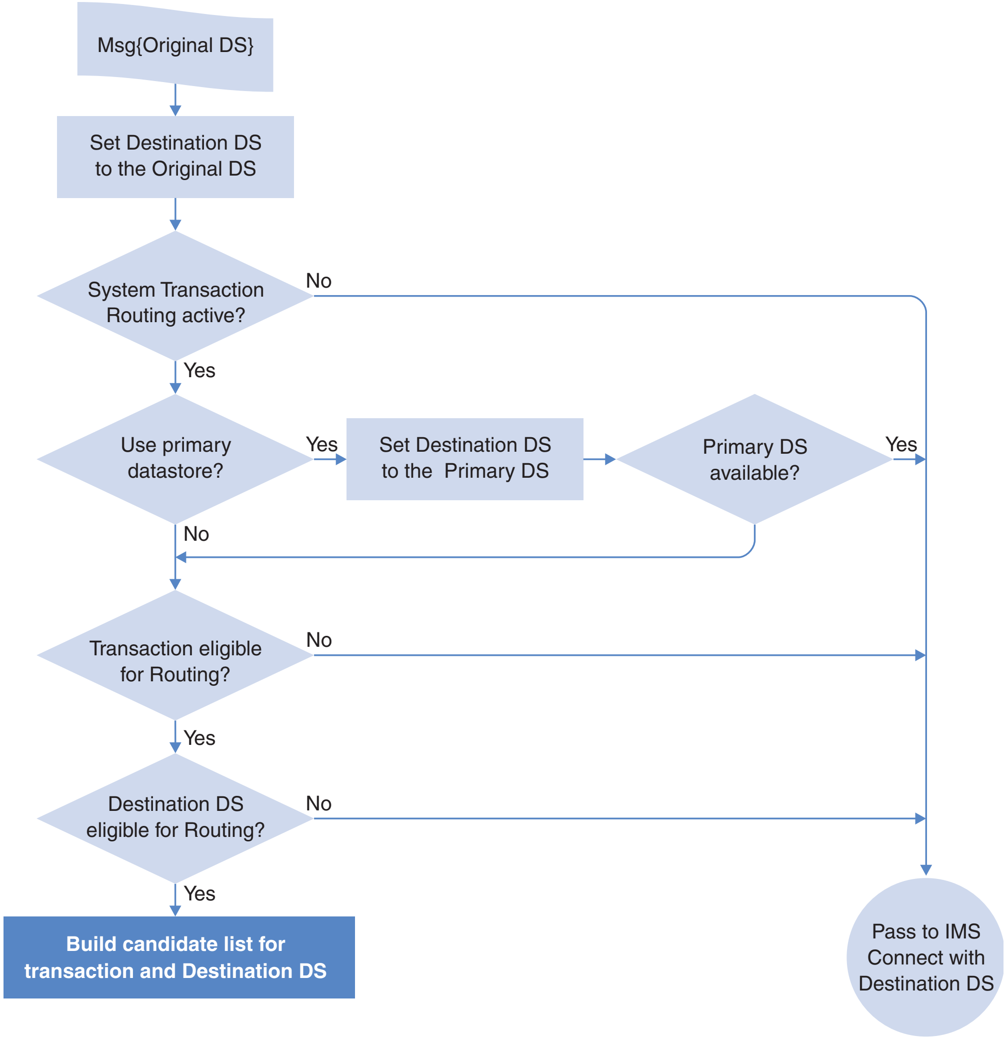

Msg{Original DS}
Set Destination DS to the Original DS
System Transaction Routing active?
No
Yes
Use primary datastore?
Yes
Set Destination DS to the Primary DS
Primary DS available?
Yes
No
Transaction eligible for Routing?
No
Yes
Destination DS eligible for Routing?
No
Yes
Build candidate list for transaction and Destination DS
Pass to IMS Connect with Destination DS

*Figure 61. Determining if a transaction is eligible for transaction routing*

Transaction routing can be set on several levels:

1. If system transaction routing is inactive, the message is passed to IMS Connect with the datastore determined by the incoming transaction request header (Original DS). If system transaction routing is active, processing continues.

   To toggle system transaction routing, select or clear **Activate Transaction Routing** in the Systems Definition panel.

2. If the primary datastore should be used, and the primary datastore is available, the message is passed to IMS Connect with the primary datastore rather than the original destination. Otherwise, processing continues.

   For more information on using primary datastores, see "Primary datastore routing" on page 163.

3. If the transaction is not eligible for routing, the message is passed to IMS Connect with either the original destination or the primary datastore. Otherwise, processing continues.

   To make a transaction eligible for routing, you need to enable transaction routing for both the transaction code and the associated Application. Alternatively, override the Application options for a specific transaction code. If you override the application options, you only need to enable or disable Transaction routing on the transaction level.

   To determine the eligibility of Transactions, grouped in an Application, for routing, select or clear **Activate Transaction Routing** in the Application definition panel.

   To determine the eligibility of a specific transaction for routing, select or clear **Activate Transaction Routing** in the Transaction definition panel. To override the application options for a specific transaction code, select **Override Application Options** in the Transaction definition panel.
4. If the destination datastore is not eligible for routing, the message is passed to IMS Connect with the destination datastore. Otherwise, processing continues. The destination datastore is either the original datastore or the primary datastore, if the primary datastore option is active.

   To determine the eligibility of datastore for routing, select or clear **Activate Transaction Routing** in the Datastore definition panel.
5. A list of candidate datastores is generated by IMS Connect Extensions. Candidate datastores are those that are associated with both the transaction and the destination datastore. The destination datastore is either the original or primary datastore. For more information on how the candidate list is constructed, see “Candidate datastore list” on page 158.

If a candidate list is built for the transaction, IMS Connect Extensions will dynamically select a destination datastore.

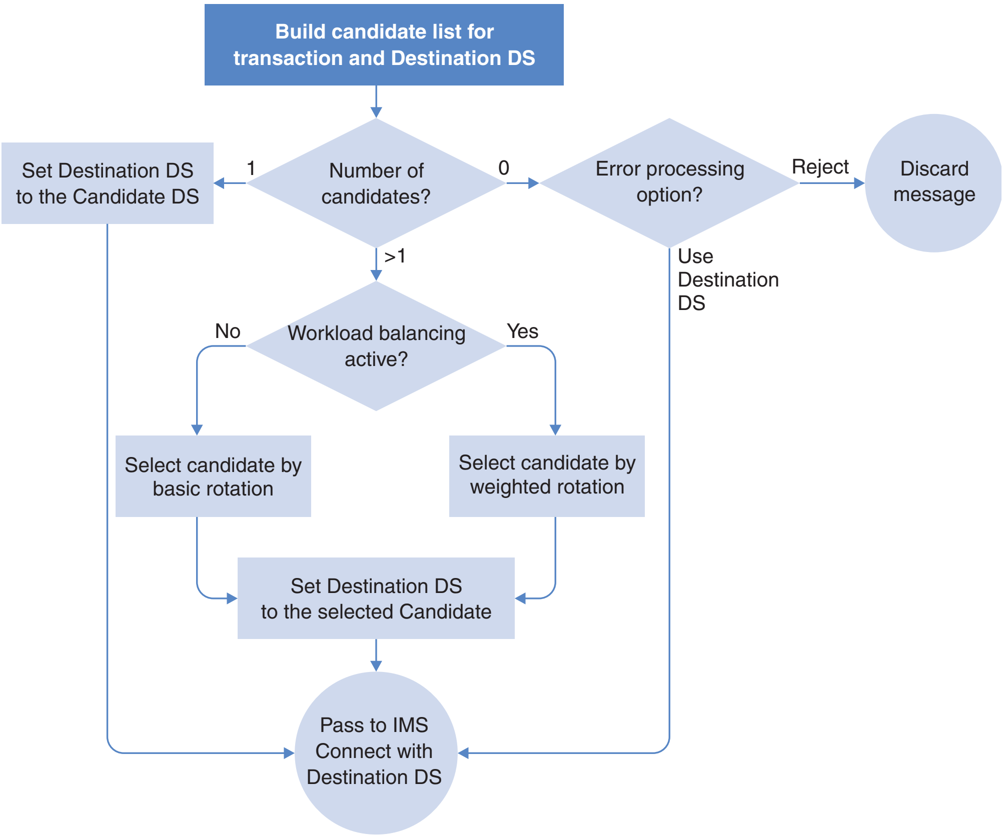

Build candidate list for transaction and Destination DS
Set Destination DS to the Candidate DS
1
Number of candidates?
0
Error processing option?
Reject
Discard message
>1
Use Destination DS
No
Workload balancing active?
Yes
Select candidate by basic rotation
Select candidate by weighted rotation
Set Destination DS to the selected Candidate
Pass to IMS Connect with Destination DS

*Figure 62. Determining a destination datastore with transaction routing*

Figure 62 shows how a destination datastore is selected if transaction routing is used:

- If there are no eligible candidates, the transaction is either discarded or passed to IMS Connect with the destination datastore. This outcome depends on the error processing option. The error processing option is set at the Application level, but you can override error processing for specific transaction codes.

  To select a routing error processing option on the Application level. See “Defining applications” on page 211.

  To override the routing error processing option on the Application level, and specify routing error processing for the transaction, see “Defining transactions” on page 213.
- If the candidate list contains only one candidate, the transaction is passed to IMS Connect with the destination datastore set to the candidate datastore.
- If the candidate list contains more than one candidate, the destination datastore passed to IMS Connect is selected using either basic rotation or weighted rotation.

  To use weighted rotation, you need to select **Activate Workload Balancing** in the System Definition panel.

## Transaction routing: flood-control processing

Transaction datastore routing includes logic designed to avoid individual datastores being flooded with OTMA messages.

For transaction routing, the following additional flood-control processing is performed:

1. IMS Connect Extensions first attempts to find a candidate datastore that is not in Degraded (flood warning) state.
2. If no available datastore can be found, the search is expanded to include datastores in Degraded state.

   **Note:** If the CEXROUTE INELIGIBLEIF=GLOBALFLOODWARNING parameter is set in the CEXCTLIN data set, any datastore that is in Degraded state is treated as Unavailable and therefore cannot be a candidate for routing.
3. If no available candidate can be found, the standard process for handling an empty candidate list is followed, as shown in Figure 62 on page 157.

## Supported message types

Transactions can be routed based on the transaction code and destination ID specified by the incoming message. How particular messages are handled by transaction routing depends on the request type.

Here are the supported request types:

**Transactional**
IMS Connect Extensions can dynamically alter the target datastore for conversational and non-conversational transactions based on routing and workload balancing rules. The first message in a conversational transaction is dynamically routed, and then subsequent messages in the conversational transaction are routed to the same datastore.

**ACK, NAK, or DEALLOC**
Positive acknowledgments (ACKs), negative acknowledgments (NAKs), and deallocate (DEALLOC) requests, whether in a conversation or non-conversational context, are always routed to the same datastore as the message that initiated the transaction.

**Send Only**
IMS Connect Extensions can route Send Only request types like other message types. However, you can choose to disable Send Only routing, without affecting the routing of other message types.

Although Resume TPIPE (RTPIPE) requests do not have a transaction code they can be routed based on the destination datastore. See “Routing Resume TPIPE messages with transaction routing” on page 162.

## Candidate datastore list

When transaction routing is activated, IMS Connect Extensions builds a list of candidate datastores. The list contains those datastores that have affinity with both the transaction and the destination datastore. Datastores that are unavailable are not included in the candidate list.

You need to define the affinity of transactions and datastores.

**Transaction affinity**
Define transaction affinity for either individual transaction codes or applications. A transaction can have affinity with all datastores, a single datastore, a datastore group, or an affinity list.

**Datastore affinity**
If you use transaction routing, you must define a datastore affinity for each datastore. A datastore can have affinity with all datastores, a single datastore, a datastore group, or an affinity list.

The following are some examples of how a candidate list is constructed.

### Example: candidate list with single entry

As shown in the following figure, an incoming message enters an IMS Connect system containing eight datastores. IMS Connect Extensions determines the affinity list and finds a single datastore that shares affinity with both the transaction and the destination datastore. IMS Connect Extensions then routes the message to the candidate.

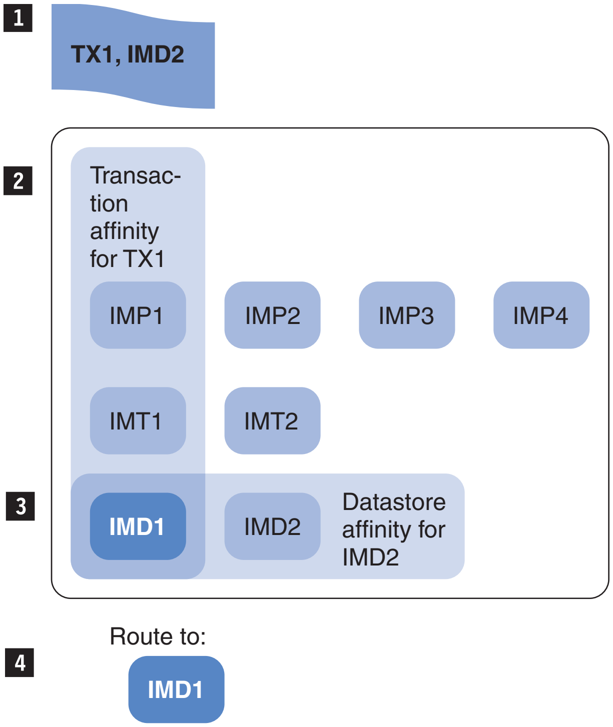


*Figure 63. Example: Candidate list with single entry*

**1** When IMS Connect Extensions constructs the candidate list, the incoming message has the transaction set to TX1 and the destination datastore set to IMD2.

**2** The transaction TX1 has Transaction Affinity with datastores IMP1, IMT1 and IMD1

**3** The destination datastore IMD2 has Datastore Affinity with datastores IMD1 and IMD2

**4** Only the datastore IMD1 shares affinity with both the transaction and the destination datastore.

### Example: empty candidate list, error routing

As shown in Figure 64 on page 160, an incoming message enters an IMS Connect system containing eight datastores. IMS Connect Extensions determines the affinity list and finds that there are no eligible candidates, as the transaction and the datastore do not have a shared affinity. IMS Connect Extensions then uses the error processing option set in the transaction or application definition to determine whether to discard the message or pass it to IMS Connect.

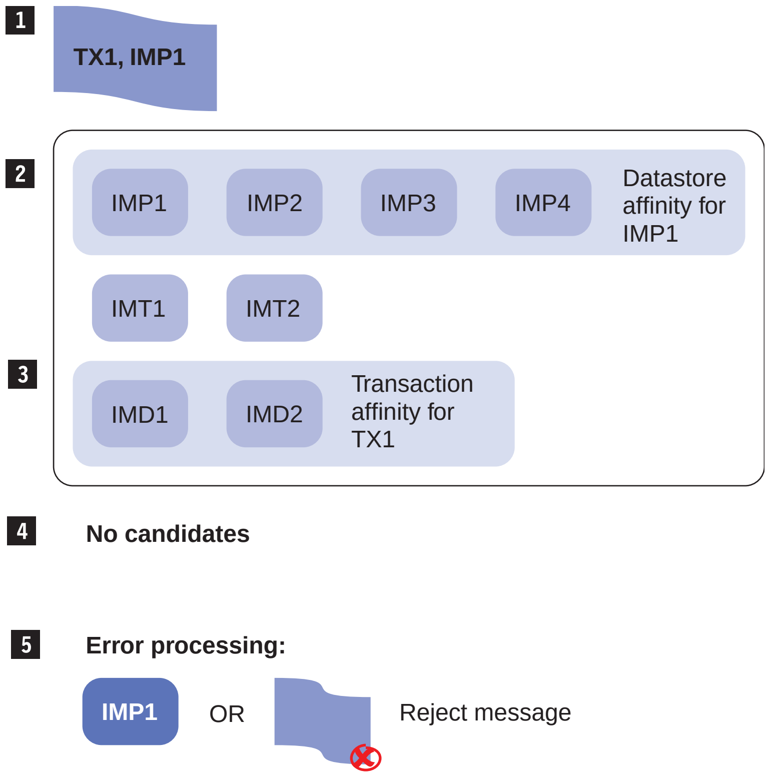


*Figure 64. Example: Candidate list with error routing option*

**1** When IMS Connect Extensions constructs the candidate list, the incoming message has the transaction set to TX1 and the destination datastore set to IMP1.

**2** The destination datastore IMP1 has Datastore Affinity with datastores IMP1, IMP2, IMP3, and IMP4.

**3** The transaction TX1 has Transaction Affinity with datastores IMD1 and IMD2.

**4** None of the datastores have a common affinity, so there are no eligible candidates.

**5** Depending on the error processing option, the message is either discarded or passed to IMS Connect with the destination datastore.

## Routing messages based on the transaction code and destination ID

Implementing transaction routing involves configuring datastores and datastore groups, affinity lists, applications, and transactions, and then linking the applications to the system definition and activating transaction routing on the required systems.

### Procedure

1. The first step is to determine how your datastores can share workloads and therefore what logical groups of datastores you need to define. Typically you will think of your datastores as being grouped by some of the following:
   - Lifecycle (Development, Test and Production)
   - Function (Finance, Sales, Payroll)
   - Geographic location (London, New York, Paris)

- Service level (Gold, Silver, Bronze)
- Department or business unit (Head Office, Factory, Branches)

2. Define the logical groups. The next step is to define your logical groups of datastores to IMS Connect Extensions. Your primary logical group should be defined as Datastore Groups.
   a. Datastore Groups
      1) Select option 1.4 **Datastore Groups** and create a Datastore Group.
      2) Press the Exit function key (F3) to save and exit.
      3) Repeat for all Datastore Groups.

      **Note:** See "Primary datastore routing" on page 163 for information on using datastore groups for primary datastore routing.
   b. Next you define the datastores to IMS Connect Extensions. Define every datastore that is defined in your IMS Connect configuration member.
      1) Select option 1.3 **Datastores**, and then create a datastore definition.
      2) Specify a Datastore Group name (optional). This defines the Datastore Group for this datastore.
      3) Press the Exit function key (F3) to save and exit.
      4) Repeat for all datastores.
   c. If you have more than one logical group of datastores you need to define affinity lists. For example, one affinity list to support your Payroll transactions, another for your Finance transactions.
      1) Select option 1.5 **Affinity Lists**, and then create an Affinity List definition.
      2) Add datastores to the list.
      3) Press the Exit function key (F3) to save and exit.
      4) Repeat for all Affinity Lists.
3. Datastore Affinity. For every datastore you must define the Transaction Routing options.
   a. Select option 1.3 **Datastores**, and then create a datastore definition.
   b. Ensure **Transaction Routing** is activated.
   c. Ensure the Routing Option is defined: select an option 1-4.
   d. Press the Exit function key (F3) to save and exit.
   e. Repeat for all datastores.
4. Applications. The next step is to define at least one Application. Applications serve two purposes:
   - To define the default transaction routing options for all transactions that are part of it: every transaction must belong to an Application
   - To associate transaction definitions to IMS Connect Extensions

   a. Select option 1.6 **Applications** and then create an application definition.
   b. Ensure **Transaction Routing** is activated.
   c. Ensure the **Route transactions** option is defined.
   d. Ensure the **Routing Error** option is defined.
   e. Press the Exit function key (F3) to save and exit.
   f. Repeat for all Applications.
5. Transactions.
   a. Create a Transaction or a Transaction prefix (Option 1.7 **Transactions**). A transaction prefix allows you to define a set of rules for a group of similarly

named transactions; for example PAY* for all payroll transactions. If you only want to route messages based on the original datastore, you can define a transaction '*' as the default definition for all transactions.

   b. Ensure Application is defined.
   c. Ensure **Transaction Routing** is activated.
   d. If this transaction requires different options to the parent Application:
      1) Select **Override Application options**.
      2) Ensure the **Route transactions** option is defined.
      3) Ensure the **Routing Error** option is defined.
   e. Press the Exit function key (F3) to save and exit.
   f. Repeat for all transactions.
6. System Definitions. The next step is to link the Applications to the System Definition.
   a. Select option 1.1 **System Definitions** and then create or edit a system definition.
   b. Ensure Advanced Features is activated.
   c. Ensure **Transaction Routing** is activated.
   d. Define Applications to the System Definition.
   e. Press the Exit function key (F3) to save and exit.
7. Finally, restart the IMS Connect system.

## Routing Send Only messages with transaction routing

IMS Connect Extensions can route Send Only messages, like other request types, using transaction routing.

### About this task

You can choose to enable or disable routing of Send Only messages for each datastore. A Send Only message that is not routed is sent to the original datastore specified by the DESTID.

If you are using Resume TPIPE messages to retrieve the output of Send Only messages, see“Routing Resume TPIPE messages with transaction routing.”

### Procedure

To route Send Only messages:

1. Define transaction routing rules. See “Routing messages based on the transaction code and destination ID” on page 160.
2. Activate Send Only routing for those datastores for which you want to route Send Only messages.

## Routing Resume TPIPE messages with transaction routing

Resume TPIPE messages do not specify a transaction code so you can only determine which datastores to route this request type based on the original datastore (DestID).

### About this task

Unlike other request types, to route Resume TPIPE requests you do not need to associate transactions, applications, and datastores. Instead, affinity lists are used to associate candidate datastores with the original datastore.

If you use Resume TPIPE to retrieve the output of a Send Only message, you need to ensure that the candidate datastores for both the Resume TPIPE messages and the Send Only transactions are the same and that those candidates are on the same shared queue. Further, the IMS Connect system processing the Resume TPIPE has to be part of the same OTMA Super Member as the IMS Connect system that processed the Send Only request.

### Procedure

To route Resume TPIPE requests:

1. Activate transaction routing for the IMS systems that process the Resume TPIPE requests.
2. Create an affinity list with the candidate datastores.
3. Activate Resume TPIPE routing for the datastore definitions for which you want to route Resume TPIPE and select the candidate affinity list.

## Primary datastore routing

Primary datastore routing lets you nominate a datastore as a primary for an IMS Connect system. Messages are routed to the primary unless it is unavailable.

If the datastore is unavailable, you can automatically revert to transaction routing as a backup mechanism (see Figure 61 on page 155).

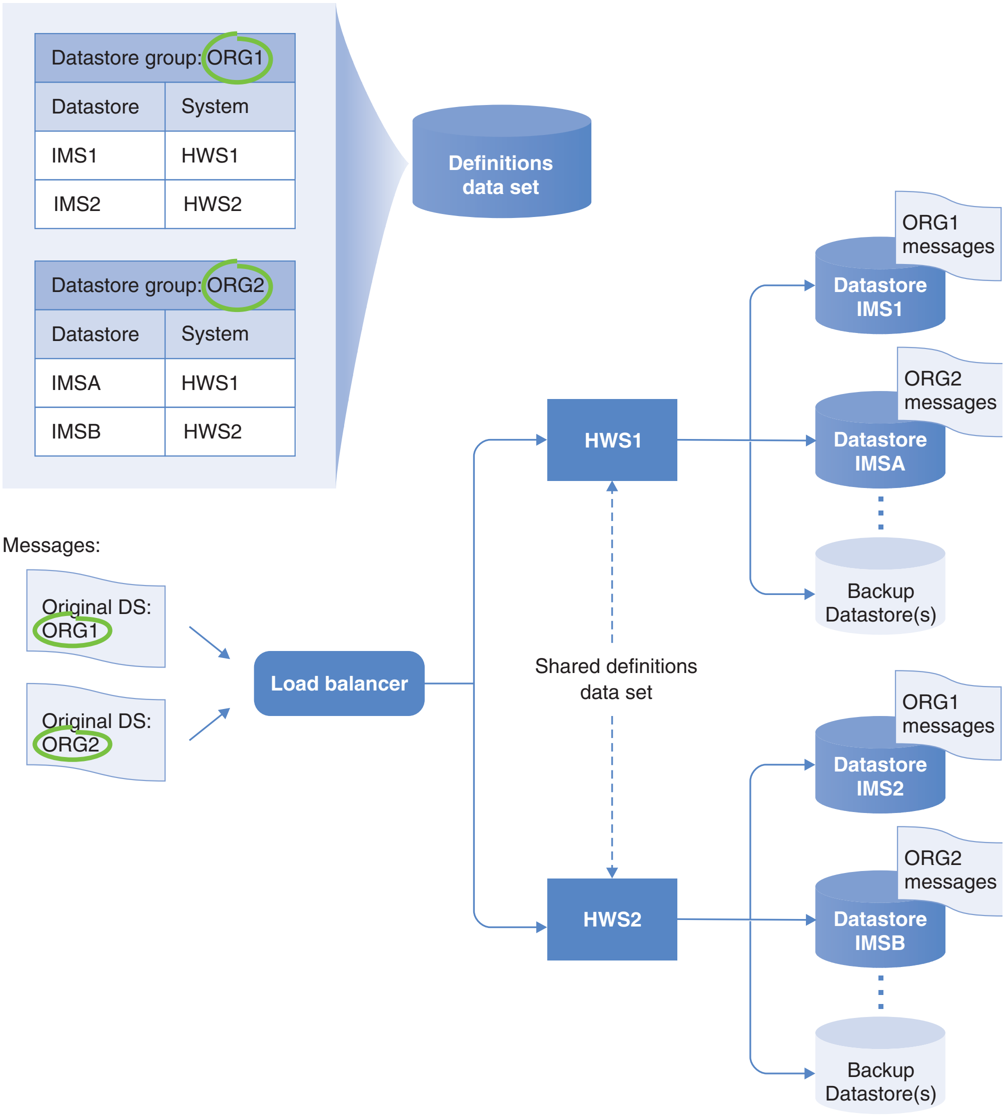
You can use primary datastore routing to handle the following scenarios:

- You want to route to a specified datastore but have backup routing available if this datastore is unavailable.
- You want to use a generic destination datastore name and route to the appropriate datastore for the IMS Connect system handling the message.

  For example (shown in Figure 65 on page 164), route all messages with the destination datastore ORG1 to the datastore IMS1 if passing through the IMS Connect system HWS1, and to the datastore IMS2 if passing through the IMS Connect system HWS2.

  This scenario can be used even if sharing a repository with multiple sites.

These scenarios could also be implemented through rules-based routing. However, if the datastore is unavailable under rules-based routing, you cannot revert to transaction routing as a backup mechanism.



| Datastore group: ORG1 | |
|---|---|
| Datastore | System |
| IMS1 | HWS1 |
| IMS2 | HWS2 |

| Datastore group: ORG2 | |
|---|---|
| Datastore | System |
| IMSA | HWS1 |
| IMSB | HWS2 |

*Figure 65. Primary datastore routing uses the datastore group to route messages*

Figure 65 shows that primary datastores are defined in datastore groups. The datastore group contains a datastore and the IMS Connect system for which it is the primary.

As shown in Figure 66 on page 165:

- If the destination datastore is defined as a datastore in IMS Connect Extensions, a primary datastore is not used.
- If the destination datastore is not defined, IMS Connect Extensions checks if a datastore group with the same name as the destination exists. If one does not exist, the message is passed to IMS Connect with the original datastore as the destination. If the datastore group exists, IMS Connect Extensions checks the datastore group to find which datastore is configured as the primary, and sets the destination to the primary datastore.

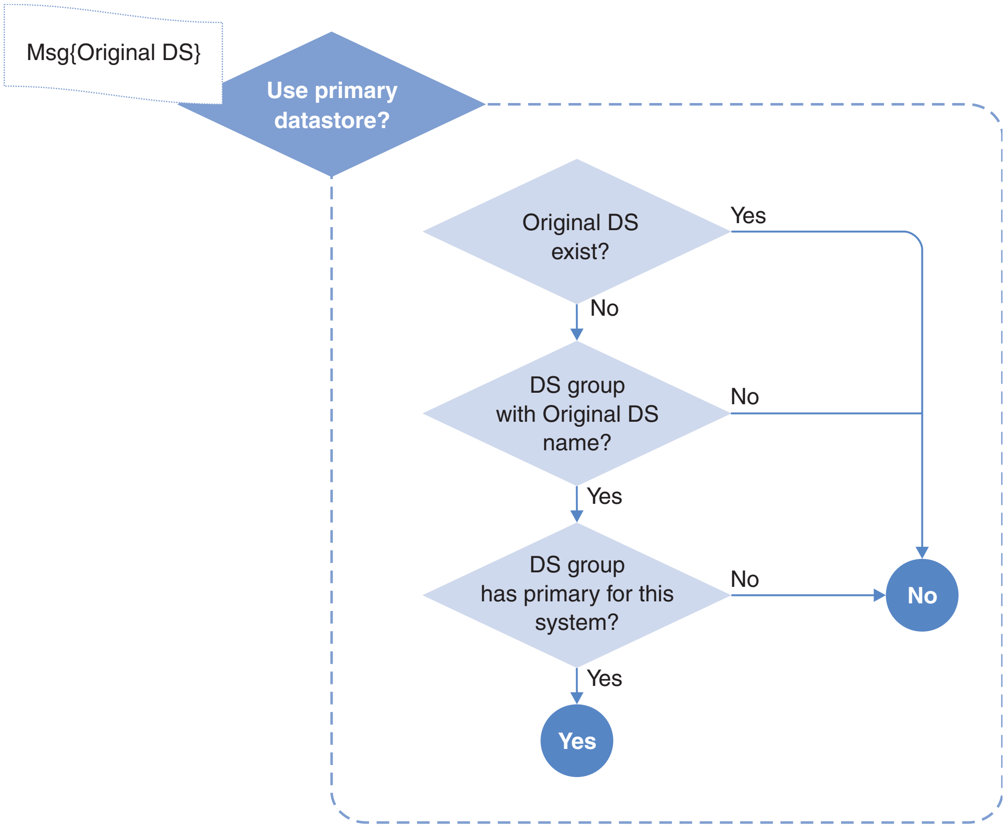


*Figure 66. Determining whether to route a transaction to a primary datastore*

## Routing messages to a primary datastore

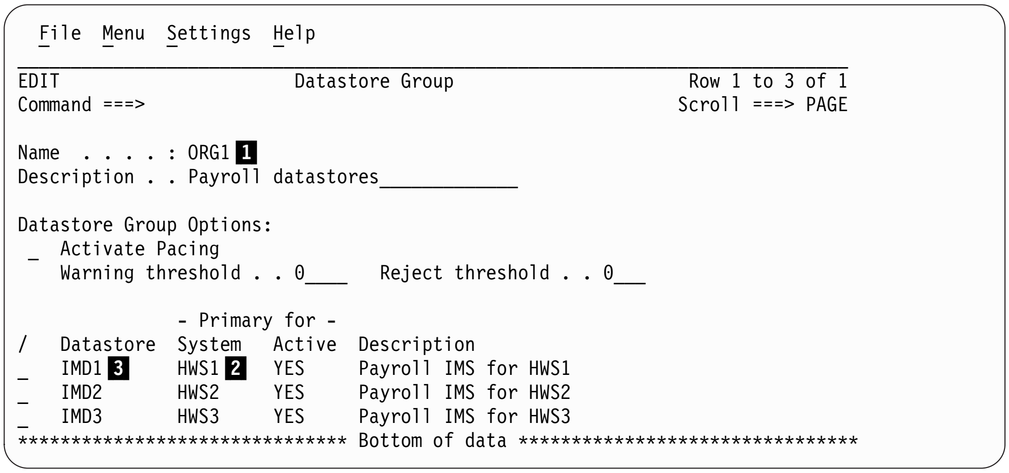
Implementing primary datastore routing involves configuring one or more datastore groups whose name corresponds to a destination ID set by clients in the incoming message.

### Before you begin

For primary datastore routing to take effect, transaction routing must be activated and an application must be defined in the system definition.

### Procedure

1. Define a datastore group for each generic destination datastore name you want to use in messages. The name of the datastore group must be exactly the same as the destination ID set by clients in the incoming message.
2. For each datastore that you want to use for primary datastore routing, set the IMS Connect system and select one of the virtual datastores as a datastore group. As you build your datastore group, it will look similar to the datastore group in the following figure.



```
  File  Menu  Settings  Help
──────────────────────────────────────────────────────────────────────────────
EDIT                       Datastore Group                      Row 1 to 3 of 1
Command ===>                                                  Scroll ===> PAGE

Name  . . . . : ORG1 1
Description . . Payroll datastores_____________

Datastore Group Options:
 _  Activate Pacing
    Warning threshold . . 0____   Reject threshold . . 0___

               - Primary for -
/   Datastore  System   Active  Description
_   IMD1 3     HWS1 2   YES     Payroll IMS for HWS1
_   IMD2       HWS2     YES     Payroll IMS for HWS2
_   IMD3       HWS3     YES     Payroll IMS for HWS3
****************************** Bottom of data ********************************
```

*Figure 67. An example datastore group used for primary routing*

In this example, an incoming message with the destination ID ORG1 **1**, which passes through the IMS Connect system HWS1 **2**, is routed to IMD1 **3**.

## Primary datastore routing: flood-control processing

Primary datastore routing includes logic designed to avoid individual datastores being flooded with OTMA messages.

For primary datastore routing the following flood-control processing is performed:

1. The primary datastore is never used when it is in Degraded (flood warning) or Unavailable state. IMS Connect Extensions reverts to transaction routing as a backup mechanism.
2. If no backup datastore is available, one of the backup datastores can be used if it is in Degraded state.

   **Note:** If the CEXROUTE INELIGIBLEIF=GLOBALFLOODWARNING parameter is set in the CEXCTLIN data set, any datastore that is in Degraded state is treated as Unavailable and therefore cannot be a candidate for routing.

## Supported message types

How particular messages are handled by primary datastore routing depends on the request type.

Messages whose original datastore name is the same as the datastore group name are routed to the primary datastore in that group, if handled by the selected system. Here is additional information on how certain request types are handled by primary datastore routing:

**Transactional**
: IMS Connect Extensions can dynamically alter the target datastore for a conversational transaction based on routing and workload balancing rules. The first message in a conversational transaction is dynamically routed, and then subsequent messages in the conversational transaction are routed to the same datastore.

**ACK, NAK, or DEALLOC**
: Positive acknowledgments (ACKs), negative acknowledgments (NAKs), and deallocate (DEALLOC) requests, whether in a conversation or non-conversational context, are always routed to the same datastore as the message that initiated the transaction.

# Chapter 15. Routing Open Database requests

ODBM routing allows IMS Connect Extensions to alter the target IMS system that will process incoming database requests, by dynamically changing the target alias used for IMS ODBM communication. This improves availability and performance.

An alias name within the incoming DRDA request sent by the client application normally determines the target IMS alias for processing. However, IMS Connect Extensions provides a mechanism for dynamically routing requests by allowing the target IMS alias to be substituted with an alternative according to a specified set of rules. These rules define a collection of target and fallback ODBM lists to use for a given alias name and optionally a PSB name supplied by a request.

The routing rule definition allows you to define rules for ODBM routing. These rules convert the original input alias on the client request to an ODBM name and alias defined in the ODBM target. Routing rules that are intended to operate at the same time can be grouped under a routing plan. You can switch to a new routing plan by issuing a Set Routing Plan command (using a REXX exec, the ISPF dialog, or the Operations Console plug-in) and then refreshing the routing rules.

The IMS Connect Extensions routing exit CEXROUT0 replaces the IMS Connect user-written DB Routing Exit routine (HWSROUT0).

## How ODBM routing rules are processed

The following steps are used to find the routing target for a DRDA request.

1. IMS Connect Extensions checks for the name of the active ODBM routing plan, if there is one. Only one ODBM routing plan can be active on a system. Routing rules that are assigned to other routing plans are not considered.

   A routing rule that is not assigned to any routing plan is always in effect. Where there is a conflict between an unassigned routing rule and a routing rule that is explicitly assigned to the routing plan that is currently active, the rule that is assigned to the current routing plan takes precedence. In other words, you can define unconditional rules to specify generic routing behavior and then override that behavior using specific rules that are assigned to the current routing plan.
2. The candidates in the target list are checked to find which are active and can be considered as potential candidates for routing.
   - If there is at least one active candidate in the resulting list, one of the targets is selected from the list of active candidates. If workload balancing is active, the relative weight for the target candidates is used in the selection formula.
   - If no active candidates are found in the target list, the process is repeated using the candidates in the fallback list.
   - If there are no active candidates in either the Target or Fallback lists, “normal default processing” is used to select the ODBM name and alias target returned by CEXROUT0. See “Normal default processing” on page 170.
3. The selected candidate is checked to see if the target alias name selected for the ODBM name is set to blanks. If it is, CEXROUT0 attempts to find the target candidate for the ODBM name returned.
4. If the ODBM name is active, a candidate is selected from the target list of active alias entries for the ODBM name.

If a candidate can not be selected for some reason (for example, the ODBM name selected is not active, or no active alias name exists) “normal default processing” is used.

## Sample ODBM routing rules descriptions

When a system starts that has ODBM routing rules activated, messages like the following are written to CEXPRINT:

```
08.10.47 CEX5050I SYB1 request: Set ODBM plan active
08.10.47 CEX5599I Routing plan status at IMS Connect start
08.10.47 CEX5603I There is no active OTMA routing plan.
08.10.47 CEX5598I The ODBM routing plan name PEAK is active
08.10.47 CEX5050I SYB1 action : Set ODBM plan active - successful
08.22.08 CEX5050I SYB1 request: Refresh ODBM routing rules

04.40.16 CEX5191I ---Begin ODBM routing rules descriptions---, refresh
                  index=1---
04.40.16 CEX5191I ODBM Routing plan is PEAK 1
04.40.16 CEX5191I ODBM routing rule for alias=DETS
04.40.16 CEX5191I PSB name=*
04.40.16 CEX5191I Request             --Routing  Lists--
04.40.16 CEX5191I type      Source    Target    Fallback
04.40.16 CEX5191I --------  --------  --------  --------
04.40.16 CEX5191I DRDARQST  PLNRULEW  ODLSTB
04.40.16 CEX5191I End of entries for ODBM routing rule
04.40.16 CEX5191I ODBM routing rule for alias=PROD
04.40.16 CEX5191I PSB name=AUTPSB11
04.40.16 CEX5191I Request             --Routing  Lists--
04.40.16 CEX5191I type      Source    Target    Fallback
04.40.16 CEX5191I --------  --------  --------  --------
04.40.16 CEX5191I DRDARQST  PHONRULE  PHONLIST
04.40.16 CEX5191I End of entries for ODBM routing rule
04.40.16 CEX5191I ODBM routing rule for alias=PROD
04.40.16 CEX5191I PSB name=DFS*
04.40.16 CEX5191I Request             --Routing  Lists--
04.40.16 CEX5191I type      Source    Target    Fallback
04.40.16 CEX5191I --------  --------  --------  --------
04.40.16 CEX5191I DRDARQST  DFSRULE 2 DFSLIST
04.40.16 CEX5191I End of entries for ODBM routing rule
04.40.16 CEX5191I ---End ODBM routing rules descriptions---
04.40.16 CEX5193I ---Begin ODBM routing list description---
04.40.16 CEX5193I Routing             ODBM      IMS    Weight
04.40.16 CEX5193I lists     Target    name      alias  rating
04.40.16 CEX5193I --------  --------  --------  -----  ------
04.40.16 CEX5193I DFSLIST 3 DFSTARG   DCA1OD    ODP1        1
04.40.16 CEX5193I ODLSTB    ODTARGB   DCA1OD    ODP2        1
04.40.16 CEX5193I PHONLIST  PHONETGT  DCA2OD    ODP2        1
04.40.16 CEX5193I ---End ODBM routing list description---
```

*Figure 68. Sample CEXPRINT messages showing ODBM routing descriptions*

This describes how the routing rules are configured on this system. For each alias it shows the source or name of the definition it has determined the rule from, and the target and fallback routing lists that the rule will use. Only eligible datastores for this system are displayed for each routing list.

Note that many more ODBM routing plans, routing rules, routing lists, and datastores are likely to be defined in the repository. The ODBM routing rules descriptions that are written to CEXPRINT show the in-memory rules and objects that are currently in effect.

**1** PEAK is the active ODBM routing plan on this system. The ODBM routing rules description block displays the routing rules that are active because they are either assigned to the active routing plan or unassigned.

2 The routing rule for alias=PROD and PSB names starting with DFS is named DFSRULE. The ODBM routing list DFSLIST defines the primary ODBM targets for requests that match this routing rule.

3 The attributes of DFSLIST and other routing lists that are referred to by active ODBM routing rules are displayed in the ODBM routing list description block.

## Implementing ODBM routing

Implementing ODBM routing involves analyzing your DRDA request traffic, configuring routing plans, routing rules, targets, and PSB name lists, and then activating ODBM routing on required systems.

### Before you begin

1. Ensure that CEXROUT0 is specified as the ODBM Routing Exit in the BPE configuration member.
2. Review the ODBMDEFAULTS and CEXROUTE NOALIAS control options to confirm whether the default behavior matches your requirements (see Chapter 32, “Control input data set options,” on page 443).

### Procedure

To configure ODBM routing:

1. Analyze the DRDA request traffic in your organization to find out what input aliases and PSB names you need to cater for. Decide if you want to apply the same set of rules across all systems, or to organize systems into groups and implement a different set of rules for each system group.
2. If you want different sets of routing rules to take effect at different times, define routing plans (see “Defining routing plans” on page 229). When you define routing rules you can assign them to a routing plan.
3. Define ODBM targets (see “Defining ODBM targets” on page 223). These specify ODBM names and aliases defined in the ODBM CSL PROCLIB member. They are the building blocks for ODBM routing lists. They also contain capacity weights that influence target selection when workload balancing is active.
4. Define collections of PSB names on which ODBM routing rules can be predicated (see “Defining PSB name lists” on page 227).
5. Define collections of ODBM targets to use as candidates in ODBM routing rules (see “Defining ODBM routing lists” on page 226).
6. Define ODBM routing rules to convert an input alias into a real database target from a collection of ODBM targets that you specify (see “Defining ODBM routing rules” on page 224).

### What to do next

ODBM routing can be enabled on a per-system basis.

- To toggle ODBM routing, select or clear **Activate ODBM rules** in the Systems Definition panel.

- To activate a routing plan using the Operations dialog, first use the **Routing Plan** option in the Set Commands panel to set the plan name, and then use the **ODBM routing rules** option in the Refresh Commands panel to load the new set of rules.

  **Tip:** You can also activate a routing plan from a REXX exec using the SET PLAN and REFRESH host commands.

**Related tasks**:

"Configuring the ODBMROUT exit member" on page 30
If you want to use IMS Connect Extensions to route DRDA request traffic, specify CEXROUT0 as the ODBMROUT exit member in the BPE configuration member.

## Default processing

This topic covers the processing performed by the CEXROUT0 exit in situations that make it impossible to perform ODBM routing.

### Normal default processing

Normal default processing behavior is defined by specifying one of the following ODBMDEFAULTS options in the control input data set (see Chapter 32, "Control input data set options," on page 443):

**INPUTALIAS**

The input alias is copied as the output alias and the ODBM name is set to blanks. This is also the behavior of the HWSROUT0 exit supplied with IMS Connect.

**BLANKALIAS**

The output alias and the ODBM name are both set to blanks.

This is the processing used by the routing exit when it cannot make a routing decision for either of the following reasons:

- The input alias name does not match any ODBM routing rule. This includes the case where no ODBM routing rules are specified.
- An ODBM routing rule is found, but none of the ODBM targets for the rule are active.

### Exit default processing

The input alias is copied as the output alias and the ODBM name is set to blanks. This is also the behavior of the HWSROUT0 exit supplied with IMS Connect.

This is the processing used by the routing exit when any of the following exceptional conditions is detected:

- CEXROUT0 is configured as the ODBM routing exit, but IMS Connect Extensions is not active.
- CEXROUT0 is configured as the ODBM routing exit and IMS Connect Extensions is active, but the system definition does not specify that ODBM Routing is to be active.
- CEXROUT0 could not connect to a critical IMS Connect Extensions resource or service during initialization.
- A processing error occurs within CEXROUT0 that is not covered in "Normal default processing."

- During routing processing, a required routing rule could not be found.

## Balancing workloads based on processing capacity of ODBM targets

Workload balancing allows IMS Connect Extensions to redirect incoming database requests to one of multiple ODBM instances, thereby balancing the workload across multiple IMS systems and ensuring both availability and responsiveness for the remote client.

Workload balancing is accomplished using a weighted rotate algorithm that considers the processing capacity of an ODBM target. Targets with greater capacity receive more requests. You can specify the relative capacity weighting of each ODBM target during the definition process. Basic rotation, in which each target is deemed to have the same capacity, can be achieved by assigning each target an equal weighting value, or by deactivating workload balancing. You can keep a target in reserve by assigning it a capacity of zero, making it ineligible as a routing candidate.

IMS Connect Extensions will not route messages to targets that are unavailable or which have a capacity of zero, and will instead use an eligible candidate regardless of whether workload balancing is active. If a target is no longer available, the candidate list is automatically rebalanced across the remaining candidate targets.

Workload balancing takes effect when the Activate Workload Balancing option is selected in the System Definition and there is more than one eligible candidate for routing.

A target's capacity can be dynamically changed by IMS Connect Extensions clients. This is useful for redistributing workload in response to changing periods of demand. The member CEXRXC61 in the SCEXSAMP library shows how to make these changes using a REXX exec.

# Chapter 16. Enhancing IMS Connect security

IMS Connect Extensions enhances the security features of IMS Connect. These security enhancements are specified and activated in the Advanced Features section of the system definition.

If you activate IMS Connect Extensions security, then you should disable IMS Connect security by specifying RACF=N in the configuration member.

**Note:** These topics refer to RACF functions and terminology but IMS Connect Extensions security features are not restricted to RACF. External security managers that support the SAF interface, such as CA ACF2 and CA Top Secret, may be used.

## Caching user credentials to improve performance

When security management is activated in the system definition, IMS Connect Extensions performs user ID and password validation by issuing a RACROUTE VERIFY request. If caching of user credentials is active the accessor control environment element (ACEE) created by the RACROUTE VERIFY is saved to improve performance.

On subsequent calls for the same user ID, password, and group, IMS Connect Extensions uses the cached ACEE and so avoids having to issue another RACROUTE VERIFY request.

Caching of user credentials is activated by specifying the **Activate ACEE Cache** and **Ageing interval** fields in the system definition.

### Lifecycle of a user ID in the ACEE cache

When IMS Connect Extensions receives ENF 71 notification of a security profile change that might affect a user ID's authorization to use resources, it purges the ACEE from the cache. Otherwise ACEEs are kept in the cache for a period defined by the Ageing Interval in the system definition. The ACEE is cleared after this time interval has passed. ACEEs can also be purged from the cache using the Security Commands dialog or the CLEAR host command. IMS Connect Extensions will make a new RACROUTE VERIFY request to perform user ID and password verification in the following situations:

- The password in the message differs from that used to create the ACEE in the cache.
- The group has changed.
- The ACEE has been in the cache for longer than the ageing interval.

### Managing the ACEE cache

The ACEE cache control blocks are located in a storage pool managed by IMS Connect Extensions in the extended private area. Each ACEE cache element is 52 bytes. The ACEE Cache pool starts at a size of 81,920 bytes. The pool can expand if needed in 81,920 byte increments to a maximum size of approximately 2.5 MB. Since some of the pool space is required for cache control block management, the pools will hold approximately 48,000 ACEE cache entries.

The SECURITY CACHED_USER_MAX control option specifies the maximum number of ACEEs in the cache. The number specified is converted to the initial size of the ACEE cache pool. The ACEE cache pool is allowed to expand beyond the initial value if needed to accommodate additional user IDs.

You can display statistics relating to the ACEE cache using the **QUERY TYPE=ACEE_CACHE** host command or by issuing the **AC** line action in the Operations - Systems View dialog.

You can activate or deactivate ACEE caching dynamically by updating the ACEE settings in the system definition and then refreshing them.

**Related reference**:

"System Definition" on page 190
Use the System Definition panel to define the options and features for an IMS Connect system. You must create one system definition in IMS Connect Extensions for every IMS Connect system that you wish to manage.

"CLEAR" on page 249
The **CLEAR** command clears one or more users from the ACEE cache.

"Changing definitions dynamically with REFRESH commands" on page 234
The **REFRESH** command causes IMS Connect Extensions to read the definitions from the repository and rebuild the in-memory values.

## Validating an incoming OTMA transaction or DRDA request

If security validation is activated, IMS Connect Extensions will check whether the user ID associated with an incoming message or DRDA request is authorized.

**Note:** To enable validation of DRDA requests, you must first configure CEXAUTH0, the IMS Connect Extensions ODBM security validation exit.

If **Activate Security** and **Activate Validation** are selected in the system definition, IMS Connect Extensions will check the user ID associated with an incoming OTMA transaction or DRDA request. If RACF returns an invalid status the message request is rejected.

The authority of the user ID associated with an incoming message request can be checked in two ways:

- Is the user ID authorized to use the IMS Connect?
- Is the user ID authorized to use the IMS Connect from this IP address and listener port?

Which validation type you select determines the format and maximum length of the resource entity used on the RACROUTE FASTAUTH request.

### Validation type: IMS Connect only

IMS Connect Extensions checks whether the user ID associated with an incoming message request is authorized to use the IMS Connect system.

IMS Connect Extensions performs this validation against the security class that is specified in the system definition. The profile name is the name of the IMS Connect system, so the security class must support resource profiles up to eight bytes in length. User IDs associated with the incoming request must have read access to be authorized.

### RACF example

In this example the user IDs in group CEXGRPI are authorized to use IMS Connect system ZZSEC01.

```
RDEFINE FACILITY ZZSEC01 UACC(NONE)
⋮
/* ----------------------------------- */
/* PERMIT ACCESS */
/* ----------------------------------- */
PERMIT ZZSEC01 -
CLASS(FACILITY) ID(CEXGRPI) ACCESS(READ)
```

## Validation type: IMS Connect + IP Address + Port

This validation type allows for an additional level of security by providing a means to control access based on the client IP address and IMS Connect port number. IMS Connect Extensions checks whether the user ID associated with an incoming message request is authorized to use the IMS Connect system from the IP address and listener port associated with the incoming message request.

IMS Connect Extensions performs this validation against the security class that is specified in the system definition. The security class must support resource profiles up to sixty-four bytes in length.

The security class can be a predefined RACF class or any user-defined class in the RACF class descriptor table. The predefined FACILITY security class is, by default, limited to 44 characters; therefore, you can use it only for IPV4 addresses. For IPV6 addresses, the security class must be able to support longer resource strings. You can use the predefined XFACILIT security class. It supports up to 246 characters.

If the installation is using an IPv4 network address scheme, the profile string has the following structure:

```
CEX.IPV4.ICONname.nnn.nnn.nnn.nnn.zzzzz
```

If the installation is using an IPv6 network address scheme, the profile string has the following structure:

```
CEX.IPV6.ICONname.xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.zzzzz
```

The components of the previous two profile strings are described below:

- CEX, IPV4, and IPV6 are literals.
- *ICONname* is the name of the IMS Connect system. Do not pad the name with spaces if it is less than eight characters.
- *nnn.nnn.nnn.nnn* is the client IP address (if IPv4 is used). Nodes that are shorter than 3 digits must be padded with leading zeros.
- *xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xxxx* is the client IP address (if IPv6 is used). Nodes that are shorter than 4 characters must be padded with leading zeros.

  Consecutive nodes of zero value can be replaced with two colons (::). For example, the following addresses are equivalent:

  ```
  CEX.IPV6.ZZSEC01.AB12.0000.0000.0000.DDAA.0000.3333.4444.05678
  CEX.IPV6.ZZSEC01.AB12::DDAA.0000.3333.4444.05678
  ```
- *zzzzz* is the IMS Connect port number.
- The port number must be padded with leading zeros to the full five characters.

## RACF examples

```
RDEFINE XFACILIT CEX.IPV4.ZZSEC01.129.042.060.025.08801  UACC(NONE)
RDEFINE XFACILIT CEX.IPV4.ZZSEC01.129.042.060.030.*      UACC(NONE)
RDEFINE XFACILIT CEX.IPV4.ZZSEC01.*.*.*.*.48801          UACC(NONE)
...
   /* ---------------------------------- */
   /* PERMIT ACCESS                      */
   /* ---------------------------------- */
PERMIT  CEX.IPV4.ZZSEC01.129.042.060.025.08801           -
        CLASS(XFACILIT) ID(CEXGRPI) ACCESS(READ)
PERMIT  CEX.IPV4.ZZSEC01.129.042.060.030.*               -
        CLASS(XFACILIT) ID(CEXGRPI) ACCESS(READ)
PERMIT  CEX.IPV4.ZZSEC01.*.*.*.*.48801                   -
        CLASS(XFACILIT) ID(USRID01) ACCESS(READ)
```

The first RDEFINE statement creates a rule to block access to the IMS Connect system ZZSEC01 on port 08801 from the specified IP address.

The second RDEFINE statement creates a rule to block access to the IMS Connect system ZZSEC01 on any port ( * ) from the specified IP address.

The third RDEFINE statement creates a rule to block access to the IMS Connect system ZZSEC01 on the specified port from any IP address.

The first PERMIT statement grants read access by members of the RACF group CEXGRPI to the IMS Connect system ZZSEC01 from the specified port and IP address.

The second PERMIT statement grants read access by members of the RACF group CEXGRPI to the IMS Connect system ZZSEC01 on any port from the specified IP address.

The third PERMIT statement grants read access by the user USRID01 to the IMS Connect system ZZSEC01 on the specified port from any IP address.

The following examples are **incorrect**:

```
CEX.IPV4.ZZSEC01.192.168.1.10.1234
CEX.IPV6.ZZSEC01.AB12.0.0.0.DDAA.0.3333.4444.5678
```

Leading zeros are omitted from some address nodes in both examples and from the port number. The examples should be specified as follows:

```
CEX.IPV4.ZZSEC01.192.168.001.010.01234
CEX.IPV6.ZZSEC01.AB12.0000.0000.0000.DDAA.0000.3333.4444.05678
```

## Control options that affect validation

The OMUSR_TRSTUSR flag in the IRM header can be used by a client to identify certain OTMA messages as coming from a trusted user. You can use the SECURITY control option to tell IMS Connect Extensions whether or not to perform IP validation for messages that have the trusted user flag set. When SECURITY VALIDATE_TRUSTED=NO is specified, messages from trusted users are not validated. When SECURITY VALIDATE_TRUSTED=YES is specified and IP validation (type 2) is in effect, messages are validated even if the OMUSR_TRSTUSR flag is set.

The SECURITY PWCASE control option determines how mixed-case passwords are handled. You can use this to specify that passwords are to be translated to uppercase, or left as mixed case, or else the setting from the external security manager is to be used. By default, the Activate Mixed Case Passwords setting in the system definition is used.

**Related tasks**:

"Configuring the Open Database security exit" on page 31
CEXAUTH0 is the IMS Connect Extensions ODBM security exit. When installed it enables for Open Database workloads the same access control options as for OTMA workloads.

## Security preprocessing exit

The purpose of a security preprocessing exit is to parse an incoming IMS request message and return the following fields, if specified: user ID, password or PassTicket, SAF group name, and optional application name.

These fields are used by the SAF-enabled external security manager to perform user ID and password verification.

A security preprocessing exit is required if these fields are not located at fixed offsets within the IRM or if the incoming message uses encrypted passwords. This topic describes the requirements for creating a security preprocessing exit for IMS Connect Extensions.

To invoke a security exit, edit the definition of the user message exit that processes the message and specify the name of the executable that returns the IRM offsets.

- Using the ISPF dialog: In the User Exit definition panel, select option **3. Use supplementary exit named below** and then enter the name of the executable in the **Pre-Preprocessing Exit** field.
- Using the definition maintenance utility: In the **DEFN EX ADD** command, specify:

  ```
  IRM_OFFSETS=(EXIT,EXIT_NAME=name)
  ```

### Exit restrictions

The exit must adhere to the following rules:

1. The exit must be written in High Level Assembler (HLASM)
2. The exit must be reentrant
3. The exit must be able to execute in AMODE 31, RMODE ANY

**Note:** The exit is invoked for IRM requests from IMS Connect clients. The exit is **not** invoked for IRM requests from IMS Connect Extensions clients or IWEB clients (that is, clients that build the OTMA structure).

### Processing

The security preprocessing exit must return the data elements necessary to provide security checking, a return code indicating the User ID is to be considered 'trusted', or a return code indicating the data elements could not be returned.

The return element list contains a flag byte that the exit uses to indicate which elements in the returned list are valid. The flag byte also contains an indicator to use if no translation to EBCDIC is needed.

For each incoming message, extract the following fields from the IRM:

- The User ID, or blanks if none

- The Password or PassTicket field, or blanks if none
- The SAF Group name, or blanks if none
- Optionally, the application name when a PassTicket is used for the password field

Set the appropriate fields in the 'Security preprocessing exit return element list' including flag indicators.

## Input parameters

At entry to the exit, general register one (R1) contains the address of a parameter list defined by macro CEXSEEL.

The exit is passed the following information:

1. The address of the standard IMS Connect read message exit input parameter list
2. The address of a list of values that must be returned by the exit. The list is mapped by macro CEXSREL.

General Register contents at entry to exit are as follows:

- R1 contains the parameter list address
- R13 contains a standard register save area
- R14 contains the calling programs return address
- R15 contains the exit's entry address

## Output parameters

General registers at return from the exit:

- R0 can contain a return reason code if desired
- R1 can contain any value
- R2 - R12 must not be changed
- R14 contains the caller's return address
- R15 contains the exit's return codes (decimal)
  - 0 = security data returned, perform security check
  - 4 = security data not returned, consider user trusted
  - 8 = security data not returned, password is not valid or has expired
  - 12 = security element not returned, reject message with security error
  - 16 = exit encountered an error, reject message with security error

## Samples and user macros

The SCEXSAMP library provides a sample exit, CEXSPESP, that can be used as a guide in creating your own security preprocessing exit. The sample exit supports the IBM default IRM structure and returns the appropriate security fields from the message IRM. The sample data set also includes the member CEXSPASM, which contains sample JCL to assemble and link CEXSPESP.

The data set CEX.V2R4M0.USERMAC provides macros to map the input and output of the exit. This data set contains these two members:

**CEXSEEL**
The entry parameter list

**CEXSREL**
The exit parameter list

# Chapter 17. Managing IMS Connect transaction options

The IMS Connect messaging protocol allows clients to specify options in the IRM that can potentially improve transaction performance and reliability. IMS Connect Extensions allows you to modify these options at the transaction code level.

## About this task

An example of a transaction option is the IRM transaction timer. This allows clients to set how long they want to wait for a response from IMS. Too long a wait could lead to unresponsiveness at the client end, too short a wait may lead to unnecessary timeouts and a processing overhead in IMS.

Because these options are managed by the client, they are hard to centrally manage or modify at runtime. To overcome this, IMS Connect Extensions allows you to specify these options at the transaction code level rather than at each individual client.

**Notes:**

1. You must add the application that the transaction belongs to each system in which you want these options to apply.
2. If the message transaction code matches with more than one transaction in IMS Connect Extensions the options from the most specific match apply. For example: if you define two transactions A* and AB*, then, for a message that specifies the transaction code ABC, the options in AB* apply.
3. You can define default behavior by creating a transaction called "*".

## Procedure

To configure additional transaction processing options:

1. Start the IMS Connect Extensions ISPF dialog.
2. Select option 1.7 **Transactions**.
3. Enter **S** next to the transaction code for which you want to enable additional transaction processing options.
4. Select one or more of the following options:

   **Override Transaction Timer**

   The transaction timer feature allows you to set a timeout value in the IRM header of messages for this transaction or group of transactions. Activating **Override Transaction Timer** means that the timeout values specified in the transaction definition will override the IRM setting.

   When you override the transaction timer, you must also specify a message timeout value and optionally an ACK/NAK timeout value in IRM timer format. The acceptable field values are:

   `00` Default

   `E9` No timer

   `FF` Wait indefinitely

   *`xx`* A hexadecimal representation of the timeout value in minutes, seconds, or hundredths of a second

For a full description of how to specify values in IRM timer format, refer to the Transaction Timer panel in the ISPF Help or see the "Timer interval specifications" topic in the *IMS Communications and Connections Guide*.

**MSGTO**=*xx*
Message timeout value.

**ACKTO**=*xx*
ACK/NAK timeout value (Optional).

**Override Transaction Expiration**
IRM_F1_TRNEXP is a flag on the IRM that determines whether the transaction expiration feature is active or inactive. If active, then IMS Connect sets the expiration time for the input transaction. Activating **Override Transaction Expiration** means that the value specified for F1_TRNEXP in the transaction definition (either ON or OFF) will override the IRM setting.

**F1_TRNEXP=ON | OFF**
ON indicates that transaction expiration is enabled for this transaction.

**Override Client ID Cancellation**
IRM_F3_CANCID is a flag on the IRM that determines whether the Client ID Cancellation feature is active or inactive. If it is active, then a message that specifies the same client ID as that of an active session will cancel the original session and then run. Activating **Override Client ID Cancellation** means that the value specified in the transaction definition (either ON or OFF) will override the IRM setting.

**F3_CANCID=ON | OFF**
ON indicates that cancellation of duplicate sessions is enabled for this transaction.

5. Press the Exit function key (F3) to save the definition.
6. Use the Refresh command to reload the transaction definition, or restart IMS Connect.

# Chapter 18. Customizing IMS Connect message translation

IMS Connect includes exits that translate messages sent and received by clients between EBCDIC and ASCII. If translation errors occur because clients or the mainframe environment use different character sets from those implemented by the exit, IMS Connect Extensions can perform custom translation.

## About this task

IMS Connect includes exits that translate messages sent by clients in ASCII (Latin 1) to EBCDIC and then translate the EBCDIC responses back to ASCII. If your clients or mainframe environment use different character sets from those implemented by the exit, then you may encounter translation errors. Because many character sets are similar to one another, translation errors can sometimes be hard to identify. For example, you may find that the sample exit correctly translates alphanumeric characters, but incorrectly translates the Euro sign (€). This would indicate that your system character set is different from the one used by the exit, but the differences between the two only appear when certain characters require translation.

To help you address such issues, you can bypass the message exit and have IMS Connect Extensions perform custom translation. You specify the character sets for the client and the server to IMS Connect Extensions and it will perform the appropriate translation, instead of the IBM-supplied exit.

IMS Connect Extensions translation is compatible with any of the IBM-supplied sample message exits. It supports customized exits that use a similar translation mechanism to the IBM samples.

IMS Connect Extensions translation relies on z/OS Unicode Services. It is available for any single-byte character set that is supported by z/OS Unicode Services on your system. To view all the conversions that are available on your system, use the following z/OS command:

```
DISPLAY UNI,ALL
```

**Important:**

1. Do not delete or comment-out the translation routines from your message exits. IMS Connect Extensions needs to identify your translation routine before it can perform custom translation.
2. z/OS Unicode Services must contain both a conversion to and a conversion from the client and server character sets you require. If the conversions do not appear, see *z/OS Support for Unicode: Unicode Services* or consult your system administrator.

## Procedure

To configure custom message translation in IMS Connect Extensions:

1. Start the IMS Connect Extensions ISPF dialog.
2. Select option 1.2 **User Exits**.
3. Enter **S** next to the exit for which you want to perform custom translation.
4. Select **Activate translation** and set values for the following fields:

**client and server CCSID**
Specify the Coded Character Set Identifier (CCSID) for your client and server character sets.

**On error option**
Specify whether you want IMS Connect to continue using this exit if IMS Connect Extensions identifies an exceptional condition while attempting to set up or perform translation. You can either:

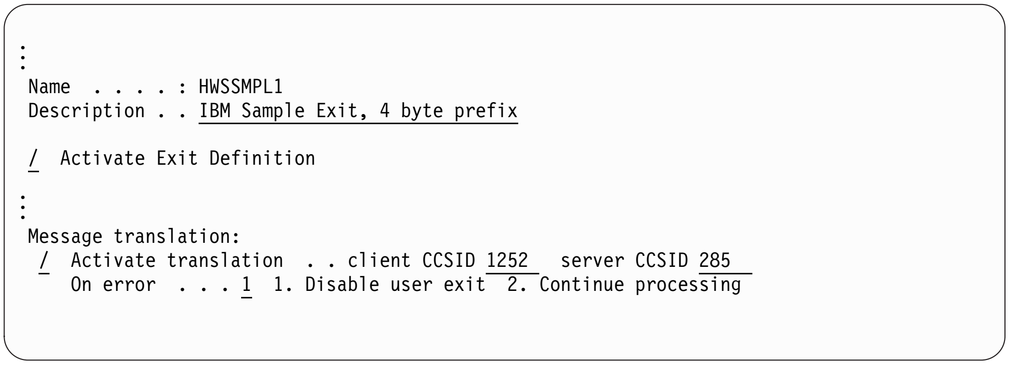
**Disable the user exit**
Clients will not be able to use the exit until you re-enable it.

**Continue processing**
Continue using the exit even though IMS Connect Extensions is no longer performing custom translation; the translation routine in the user exit is used instead.

5. Press the Exit function key (F3) to save the definition.
6. Reload the exit or restart IMS Connect.

## Example

The following exit definition for HWSSMPL1 overrides the translation map for this exit, and instead translates messages between Windows-1252 (CCSID 1252) encoding and UK EBCDIC (CCSID 285) encoding:



```
⋮
 Name  . . . . : HWSSMPL1
 Description . . IBM Sample Exit, 4 byte prefix

 /  Activate Exit Definition

⋮
 Message translation:
  /  Activate translation  . . client CCSID 1252   server CCSID 285
     On error  . . . 1  1. Disable user exit  2. Continue processing
```

*Figure 69. Translating messages between Windows 1252 encoding and UK EBCDIC encoding.*

# Part 5. Administering definitions

These topics describe how to specify IMS Connect Extensions definitions and how to dynamically change the in-memory values for definitions using the ISPF dialog.

**Topics:**

# Chapter 19. Administering definitions with the ISPF dialog

This section describes how to specify IMS Connect Extensions definitions using the ISPF dialog.

For each definition type there is typically a list panel, which lists existing definitions, and an edit panel, which you can use to specify the attributes of a new or existing definition.

## Definitions by routing method

This table shows which definitions apply to rules-based routing (RBR), transaction routing, primary datastore routing (PDR), and workload balancing.

| Definition | Routing methods | Workload balancing |
|---|---|---|
| System Definitions | All | ✓ |
| User Exits | All | |
| Datastores | OTMA RBR, Transaction, PDR | ✓ |
| Datastore Groups | Transaction, PDR | |
| Affinity Lists | Transaction, PDR | |
| Applications | Transaction | |
| Transactions | Transaction | |
| System Groups | All | |
| OTMA Routing Rules | OTMA RBR | |
| OTMA Routing Lists | OTMA RBR | |
| ODBM Targets | ODBM RBR | ✓ |
| ODBM Routing Rules | ODBM RBR | |
| ODBM Routing Lists | ODBM RBR | |
| PSB Name Lists | ODBM RBR | |
| Transaction Lists | OTMA RBR | |
| Routing Plans | OTMA RBR, ODBM RBR | |

**Related concepts**:

"Repository architecture" on page 17
Each instance of IMS Connect with IMS Connect Extensions refers to a repository, which contains configuration data for the IMS Connect Extensions components that start with IMS Connect.

**Related tasks**:

"Creating an initial configuration for IMS Connect Extensions" on page 31
Use the Definitions Setup dialog to take up definitions from the IMS Connect configuration member into a repository. This provides you a basis from which you can start customizing IMS Connect Extensions.

# Definition list panels: common features

Some commands and fields are common to definition list panels in the ISPF dialog.

## Name list

The list panel is a scrollable display of definitions that can be selected for editing. The information for each entry includes the definition name and description, and statistics fields showing who last created or modified the definition. You can edit the definition to modify the description.

The number of definitions in the list is shown in the message area after the panel title. You can reduce the number of entries that are shown by entering a pattern in the **Filter** field.

## Filter field

Only definitions that match the pattern in the **Filter** field are displayed, as shown in the following examples:

***** displays all definitions

**AB*** displays all definitions starting with AB.

**ABC*** displays all definitions starting with ABC.

**%%C*** displays all definitions with C in the 3rd position.

**%B%D***
displays all definitions with B in the 2nd position and D in the 4th position.

**%B%D%**
displays all 5 character definitions with B in the 2nd position and D in the 4th position.

If a filter is active, the status message changes to `Filter Mode`.

## Primary commands in definition list panels

**NEW name MODEL modelname**
The **`NEW`** command creates a definition. The name must be unique for this definition type. If it is omitted or invalid, you are prompted to specify a valid name and optional model.

If you specify MODEL with the name of an existing definition, IMS Connect Extensions uses the values of the modelname to populate the new definition. You can also select New from File in the action bar.

**LOCATE**
The **`LOCATE`** command (which can be entered as LOC or L) scrolls to the entry you specify or the one preceding it on the display if the entry is not found. The syntax is:

```
LOCATE string
```

where *string* can be any character string of ten or less characters. It is compared with the values of the field on which the member list is currently sorted. If the string is found, that entry is scrolled to the top of the display. If it is not found, the entry preceding it in the current sort order is scrolled to the top of the display.

**REFRESH**
The **`REFRESH`** command refreshes the member list.

It adds new members, deletes members that have been removed from the list, and resets line actions.

**RESET**
Resets line actions.

**SORT** The `SORT` command sorts a member list by one or two fields displayed on the definition list. The field names are the column headings: Name, Description (DESC), Changed (CHA) and ID.

### Line actions in definition list panels

**D** Delete this definition

**E or S** Edit this definition

**V** View this definition

## Definition edit panels: primary commands

The following primary commands can be entered in the command line on definition edit panels in the ISPF dialog. Additional primary commands are recognized for some definition types.

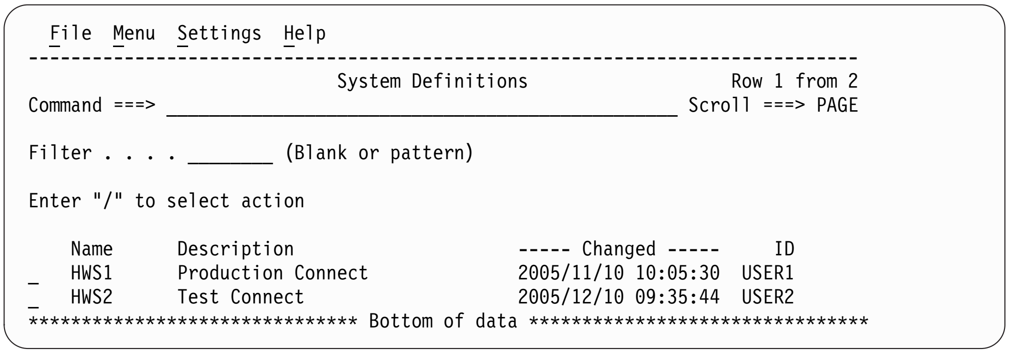
**SAVE** Save the current definition.

**SAVEAS name**
Save the current definition under the specified name. If you do not specify a name, IMS Connect Extensions will prompt you. If a definition of that name already exists in the repository, it will be overwritten. If it does not exist, it will be created.

**CANcel**
Cancel request. All updates are discarded on panel exit.

## Defining IMS Connect systems

Use the System Definitions list panel to create a new definition for an IMS Connect or to update an existing definition.



```
  File  Menu  Settings  Help
------------------------------------------------------------------------------
                             System Definitions                   Row 1 from 2
Command ===> ________________________________________________ Scroll ===> PAGE

Filter . . . . _______ (Blank or pattern)

Enter "/" to select action

    Name      Description                     ----- Changed -----     ID
_   HWS1      Production Connect              2005/11/10 10:05:30  USER1
_   HWS2      Test Connect                    2005/12/10 09:35:44  USER2
****************************** Bottom of data ********************************
```

*Figure 70. System Definitions list panel*

Enter **S** in front of a definition name to edit the definition, or enter the `NEW` primary command to create a new system definition.

## System Definition

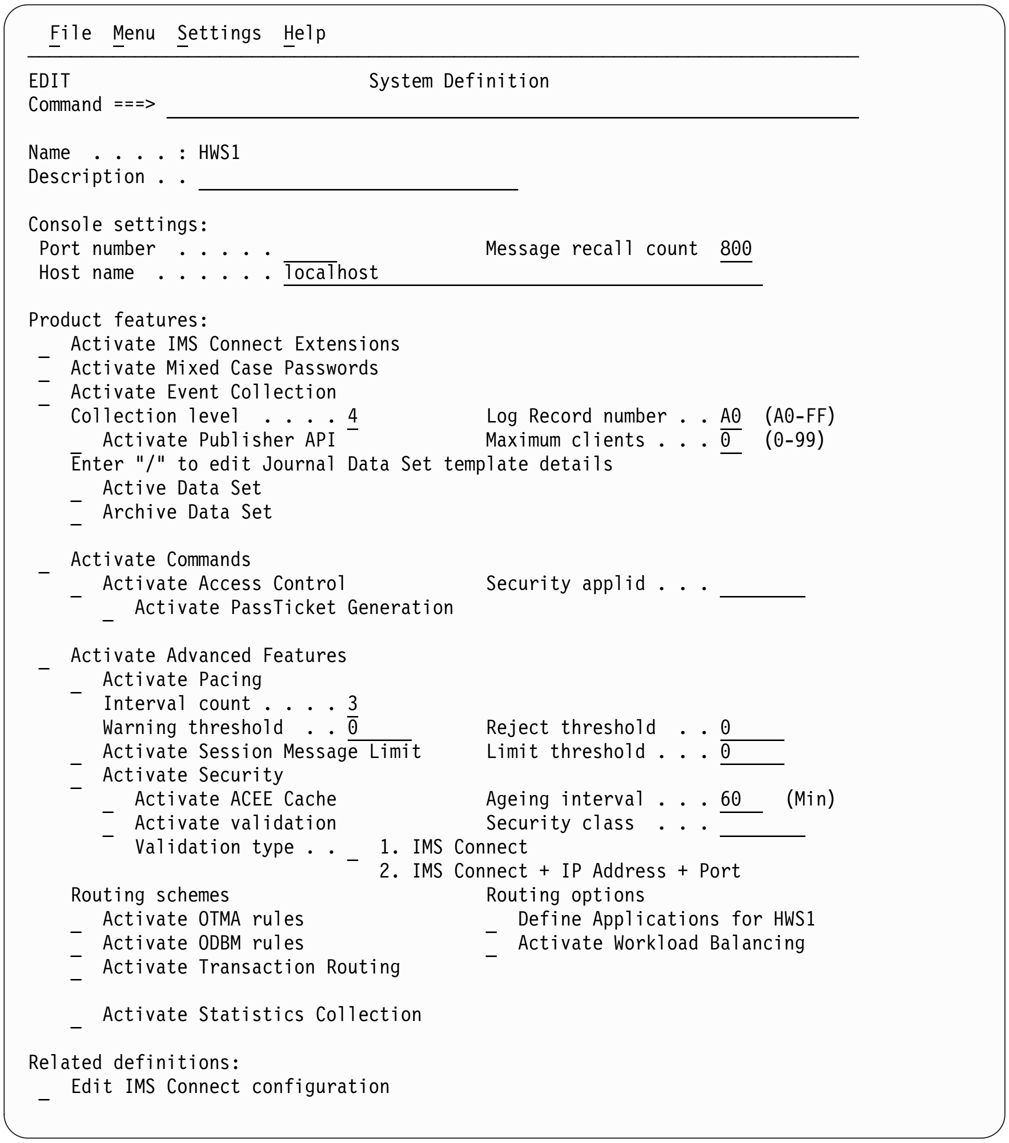
Use the System Definition panel to define the options and features for an IMS Connect system. You must create one system definition in IMS Connect Extensions for every IMS Connect system that you wish to manage.

The System Definition name **must** be the same as the HWS ID parameter in the IMS Connect configuration member. If the System Definition name does not match the HWS ID parameter, IMS Connect Extensions will not activate.

**Notes:**

1. A System Definition cannot be shared by multiple IMS Connect systems.
2. The **Warning override** profile setting determines whether a pop-up window is displayed when a warning-level error is detected during definition maintenance. When this option is set, you will be prompted to accept or reject warnings such as `TCP/IP address invalid` or `Archive JCL data set not cataloged`. When this option is not set, you must correct warning-level errors to proceed.

   You can set the warning override on or off from option 0 **Profile**.



```
  File  Menu  Settings  Help

EDIT                            System Definition
Command ===> ________________________________________________________________

Name  . . . . : HWS1
Description . . ______________________________

Console settings:
 Port number  . . . . . _____              Message recall count  800
 Host name  . . . . . . localhost_____________________________

Product features:
 _  Activate IMS Connect Extensions
 _  Activate Mixed Case Passwords
 _  Activate Event Collection
    Collection level  . . . . 4            Log Record number . . A0  (A0-FF)
    _  Activate Publisher API              Maximum clients . . . 0   (0-99)
    Enter "/" to edit Journal Data Set template details
    _  Active Data Set
    _  Archive Data Set

 _  Activate Commands
    _  Activate Access Control             Security applid . . . ________
       _  Activate PassTicket Generation

 _  Activate Advanced Features
    _  Activate Pacing
       Interval count . . . . 3
       Warning threshold  . . 0            Reject threshold  . . 0
    _  Activate Session Message Limit      Limit threshold . . . 0
    _  Activate Security
       _  Activate ACEE Cache              Ageing interval . . . 60    (Min)
       _  Activate validation              Security class  . . . ________
          Validation type . . _  1. IMS Connect
                                 2. IMS Connect + IP Address + Port
    Routing schemes                        Routing options
    _  Activate OTMA rules                 _  Define Applications for HWS1
    _  Activate ODBM rules                 _  Activate Workload Balancing
    _  Activate Transaction Routing

    _  Activate Statistics Collection

Related definitions:
 _  Edit IMS Connect configuration
```

*Figure 71. System Definition detail panel*

**Description**

A description of this System Definition.

**Console Settings**

The console listener settings determine how ISPF Operations dialog clients, Operations Console clients, REXX execs, and batch clients establish connections to IMS Connect Extensions.

**Port number**

This field sets the port number that will be used when clients connect to the console listener.

Enter a valid port number for your installation in the range 1 - 65535.

**Restriction:** Do **not** choose an existing port number listed in the IMS Connect configuration member. The IMS Connect Extensions console port must not be used by any other job or subsystem in the LPAR on which this IMS Connect system runs.

**Message Recall Count**

This field sets the number of IMS Connect Extensions messages that can be recalled on the IMS Connect Extensions message log.

Enter a value in the range 10 - 800.

**Host Name**

The IP address (IPv4 or IPv6) or DNS name of the server running IMS Connect.

**Note:** The **Warning override** profile setting determines what happens if an invalid host name or TCP/IP address is detected during definition maintenance. When this option is set, a pop-up window will be displayed to ask you whether to override the warning and continue. When this option is not set, you must correct the value to proceed.

**Activate IMS Connect Extensions**

Enter / (slash) to activate IMS Connect Extensions for this IMS Connect system.

**Activate Mixed Case Passwords**

This field allows you to activate or deactivate mixed case passwords for this IMS Connect system.

The acceptable field values are:

**/** Mixed case passwords are active. Passwords are treated as mixed case and no case translation is performed.

**blank** Mixed case passwords are not active for this IMS Connect system. Passwords are converted to uppercase before any security processing is performed.

**Note:** The SECURITY PWCASE control option determines how mixed-case passwords are handled. You can use this to specify that passwords are to be translated to uppercase, or left as mixed case, or else the setting from the external security manager is to be used.

If PWCASE=SYSTEMDEFINITION, the setting from the Activate Mixed Case Passwords option in the system definition is used. If PWCASE is set to anything other than SYSTEMDEFINITION, the Activate Mixed Case Passwords option in the system definition is ignored.

**Activate Event Collection**

This field allows you to activate and deactivate the Event Collection feature for IMS Connect Extensions. Event Collection provides the data for you to run performance and availability reports.

**Note:** To display active session details for a system, event collection must be active and the collection level must be greater than 1.

The acceptable field values are:

**/** Event Collection is active

**blank** Event Collection is not active. No event records are collected and no data is available for reporting purposes

**Collection Level**

This field allows you to set the Collection Level. The acceptable field values are:

**0** Minimum level. Collects startup and shutdown events along with some infrequent error events.

**1** Accounting level. Collects Return from Exit events, OTMA timeout and session error events, and ODBM registration and routing events. This level provides accounting information in terms of the number of messages by Transaction, User Exit, and so on.

**2** Transit time reporting. Collects the minimum number of records to run simple transit time reports.

**3** Comprehensive performance analysis. Collects all TCP/IP read and write events which provides for analysis of TCP/IP activity and remote ICON connectivity.

**4** Maximum level. Collects **all** event records.

**Log Record number**

Specify a log record prefix to identify IMS Connect records created by IMS Connect Extensions.

Acceptable values are in the range X'A0' - X'FF'. Unless you have user log records that already use the `A0` prefix, use the default value, `A0`.

**Tip:** Do not use an existing IMS log record number. A unique log record number will make IMS PA reporting easier.

**Activate Publisher API**

The Publisher API provides an interface that publishes captured data to an external application.

Enter / (slash) to activate the Publisher API.

**Note:** You must restart IMS Connect to apply changes to this option.

**Maximum clients**

Sets the initial value for the maximum number of Publisher API clients. If you change the maximum number of clients through the Publisher API console, then the number will revert to this value when IMS Connect Extensions is restarted.

**Active Data Set**

This field allows you to specify allocation information for the Active Journal data sets and any other parameters needed at initialization time.

Enter / (slash) to edit or create the Active Journal data set template. See "Active Journal" on page 197.

**Archive Data Set**

Used by the Archive Manager to determine data set usage when copying data from the Active Journal data sets to the Archive Journal data sets; and any other parameters needed at initialization time.

Enter / (slash) to edit or create the Archive Journal data set template. See "Archive Journal" on page 200.

**Activate commands**

The commands feature allows you to send IMS Connect WTOR commands and IMS Connect z/OS commands, IMS Type-1 commands, and IMS Connect Extensions commands using the command shell. The options are:

**Activate Access Control**

Select whether to restrict access to commands by using RACF or an equivalent external security manager to authorize users or to control issuing of IMS Type-1 commands from the command shell. If you activate access control you must also specify a security APPLID. This APPLID is also used for PassTicket generation.

**Security applid**

Specify the Application ID (APPLID) used to form the SAF resource profile for validation of access to commands in IMS Connect and IMS Connect Extensions. The Security Administrator at your installation defines the resource profiles in the format **CEX**.*applid.verb.object* in the SAF resource class OPERCMDS.

When PassTicket generation is selected, this APPLID is also used as the resource profile within resource class PTKTDATA to authorize and facilitate the generation process.

**Activate PassTicket Generation**

Specify whether to use PassTickets for security authentication. Use PassTickets if you do not want user passwords to be transmitted.

See Chapter 12, "Configuring command access in IMS Connect Extensions," on page 129 for details.

**Activate Advanced Features**

The Advanced Features of IMS Connect Extensions include OTMA and ODBM rules-based routing, Transaction Routing, Workload Balancing, Transaction Pacing, Security, and Statistics Collection. Enter / (slash) to activate these Advanced Features.

**Note:** You must restart IMS Connect to apply changes to this option.

**Activate Pacing**

This field allows you to activate and deactivate the Transaction Pacing feature for this IMS Connect system.

Transaction Pacing monitors incoming transaction request rates according to thresholds assigned to:

- IMS Connect system
- Datastore Groups
- Datastores

Based upon these thresholds, IMS Connect Extensions either issues warning messages or rejects transactions. The acceptable field values are:

**/** Transaction Pacing is active. Where pacing threshold settings have been defined, these will be honored. Warning messages or transaction rejections may occur.

**blank** Transaction Pacing is not active

Transaction Pacing allows IMS Connect Extensions to reject incoming transaction requests if predetermined threshold values are exceeded. IMS Connect Extensions monitors transaction arrival rates against your user-defined thresholds for identifying unusual transaction arrival rates.

Transaction Pacing is defined using a number of controls:

**Interval count**

This field allows you to specify the number of consecutive 20 second intervals in which the number of incoming messages must exceed threshold values before warning messages are sent or transactions are rejected.

The acceptable field values are in the range: 2 - 9. The default value is 3.

**Warning threshold**

This field defines the Warning threshold. At this threshold, IMS Connect Extensions issues a warning message to the IMS Connect Extensions message log and z/OS operator console indicating this threshold level has been exceeded.

The acceptable field values are any number.

A value of 0 means that Warning processing is inactive at the IMS Connect system level.

**Reject threshold**

This field defines the Reject threshold. At this threshold IMS Connect Extensions rejects incoming messages. A message is issued to the IMS Connect Extensions message log and z/OS operator console indicating maximum pacing rates have been reached.

The acceptable field values are any number.

A value of 0 means that Reject processing is inactive at the IMS Connect system level.

**Activate Session Message Limit**

This field specifies whether the Session Message Limit feature is active. If it is active, a persistent session will close when it has received the number of input messages specified in the Limit threshold field.

The expectation is that the remote client will create a new session. This is useful in environments where a session distribution mechanism such as sysplex distributor is used to route session requests between available IMS Connect systems. Session balance is gradually restored as the existing session expires and new sessions are routed to the IMS Connect with the lowest session totals.

**Limit threshold**

This field specifies the number of input messages a persistent session can receive before IMS Connect Extensions will close the session. This value must be between 1 and 999999 if Activate Session Message Limit is activated.

**Activate Security**

This field allows you to activate and deactivate security management, which enhances the security features of IMS Connect. When security management is active IMS Connect Extensions performs user ID and password validation by making a security call.

Enter / (slash) to activate security management.

If you activate IMS Connect Extensions security, then you should disable IMS Connect security by specifying RACF=N in the configuration member.

Security is controlled by the following options:

**Activate ACEE Cache**

This field allows you to specify whether IMS Connect Extensions saves the ACEE security control block. On subsequent calls for the same user ID, password, and group, IMS Connect Extensions does not reissue the security call, saving valuable system overhead. If this option is not selected, IMS Connect Extensions discards the control blocks and revalidates the user ID and password each time it is used.

Enter / (slash) to activate caching of ACEE structures.

**Ageing Interval**

This field allows you to specify the maximum age for ACEE structures before they are cleared from the cache.

Enter a value in the range 0 - 1440 minutes. An ageing interval of 0 minutes indicates the ACEEs are never deleted based on their age.

**Note:** An ACEE may remain in the cache for up to 6.25% longer than you specify.

**Activate validation**

This field allows you to specify whether IMS Connect Extensions checks that the user ID or group associated with the incoming message or DRDA request is authorized to access IMS Connect.

Enter / (slash) to activate validation.

**Validation type**

Use this field to specify the type of validation that is to be performed. Note that this option is ignored unless both Activate Security and Activate Validation are active.

The acceptable field values are:

**1** Validate using the IMS Connect name only.

**2** Validate using the IMS Connect name, the IP address, and the IMS Connect port number.

**Security class**

The security class can be a predefined RACF class such as FACILITY or any user-defined class in the RACF class descriptor table. Within the specified security class, each IMS Connect system is defined as a resource by using the name from the HWS ID parameter of the IMS Connect configuration member. This is the name that appears on this panel.

To gain access to any given IMS Connect system, users require READ authority for the relevant resource.

**Activate OTMA rules**

OTMA rules-based routing allows IMS Connect Extensions to route IRM messages to collections of datastores called routing lists. Routing lists are selected based on the message type and the original datastore ID (DESTID). Send-Receive and Send-Only message types can be further qualified by transaction.

Enter / (slash) to activate OTMA rules-based routing for this IMS Connect system.

**Define Applications for *system***

This option allows you to specify the applications that are associated with this System Definition. You need to include applications containing all the transaction codes you want to route with Transaction Routing or override transaction processing options.

Enter / (slash) to display the Applications pop-up window.

**Activate ODBM rules**

ODBM routing allows IMS Connect Extensions to route DRDA requests to collections of ODBM names and alias targets called routing lists, based on the input alias and PSB names.

Enter / (slash) to activate ODBM rules-based routing for this IMS Connect system.

**Activate Workload Balancing**

Workload balancing is accomplished using a weighted rotate algorithm. In the case of OTMA processing, datastores with greater processing capacity receive more transactions. In the case of ODBM processing, ODBM targets with greater processing capacity receive more requests.

Enter / (slash) to activate workload balancing for this IMS Connect system.

**Activate Transaction Routing**

Transaction routing is an alternative routing method to OTMA rules-based routing. It allows IMS Connect Extensions to alter the target IMS datastores that will process the incoming messages. Transaction affinity and datastore affinity rules are used to determine how individual incoming messages are rerouted.

Enter / (slash) to activate transaction routing for this IMS Connect system.

**Activate Statistics Collection**

Statistics collection must be active for information to be displayed on the Status Monitor.

Enter / (slash) to activate statistics collection for this IMS Connect system.

**Edit IMS Connect Configuration**

This option allows you to view or edit the IMS Connect Configuration member for this IMS Connect system.

Enter / (slash) to display the IMS Connect Configuration panel. See “IMS Connect configuration” on page 203.

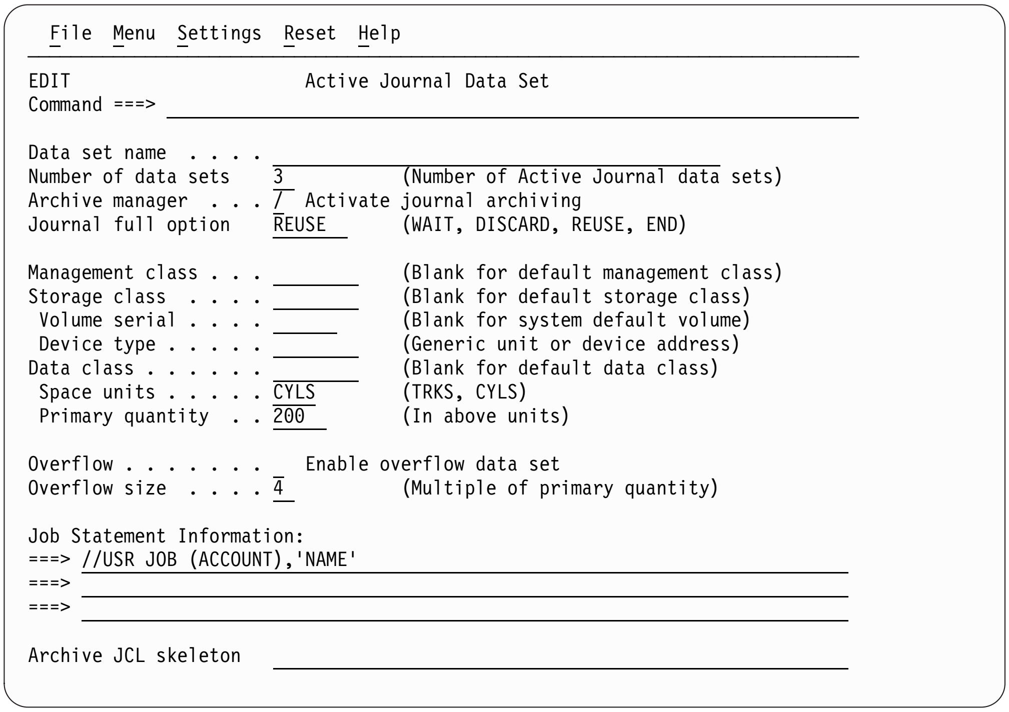
**Related concepts**:

“Configuring event collection and journals: overview” on page 61
To configure event collection on a system you must activate event collection in the system definition and enter details of the active and archive journal data sets.

## Active Journal

If you have activated Event Collection, IMS Connect Extensions writes events to the Active Journal data sets.

As part of Event Collection you must define your Active Journal data set processing options, as shown in the following figure.



```
  File  Menu  Settings  Reset  Help
─────────────────────────────────────────────────────────────────
EDIT                       Active Journal Data Set
Command ===> ____________________________________________________

Data set name  . . . . __________________________________________
Number of data sets    3          (Number of Active Journal data sets)
Archive manager  . . . /  Activate journal archiving
Journal full option    REUSE      (WAIT, DISCARD, REUSE, END)

Management class . . . ________   (Blank for default management class)
Storage class  . . . . ________   (Blank for default storage class)
 Volume serial . . . . ______     (Blank for system default volume)
 Device type . . . . . ________   (Generic unit or device address)
Data class . . . . . . ________   (Blank for default data class)
 Space units . . . . . CYLS       (TRKS, CYLS)
 Primary quantity  . . 200        (In above units)

Overflow . . . . . . . _  Enable overflow data set
Overflow size  . . . . 4          (Multiple of primary quantity)

Job Statement Information:
===> //USR JOB (ACCOUNT),'NAME'
===>
===>

Archive JCL skeleton   ____________________________________________
```

Figure 72. Active Journal template

**Data Set Name**

Specify the data set name prefix used to create the Active Journal data sets. The Active Journal is a series of permanent data sets that are used in rotation. The data sets are allocated by IMS Connect Extensions, with the suffix P*nn* where *nn* represents the Active Journal number, 01 - 32. For example:

```
'YOUR.HWSP.ACTIVE.JOURNAL.P01'
'YOUR.HWSP.ACTIVE.JOURNAL.P02'
'YOUR.HWSP.ACTIVE.JOURNAL.P03'
```

You can use the &ID symbol in the data set name to use the System Definition name when the data set is physically allocated. For example:

```
'YOUR.&ID..ACTIVE.JOURNAL'
```

**Important:** Ensure that either the user running IMS Connect or the IMS Connect task has RACF ALTER authorization for the active and archive journal data sets.

**Number of Data Sets**

Specify the number of data sets in the Active Journal.

Enter a value in the range 1 - 32. The default value is 3.

**Note:** If you specify one data set you must also specify the REUSE option in the Action flag.

**Archive Manager**

This field allows you to specify whether the Archive Manager will copy the Active Journals to the Archive Journal data sets.

**Journal full option**

This field allows you to determine what happens if all active data sets and their overflow data set (if specified) are full:

**Wait** Stop reading event records until a data set becomes available. This option will cause IMS Connect to stop processing incoming messages.

**Attention:** Using this option can reduce transaction response times.

**Discard**
Continue to create the event records but discard the records until a data set becomes available.

**Reuse** Overwrite the data sets.

**End** Shut down the Journal Manager. Event collection stops and cannot be restarted until IMS Connect is restarted.

**Management Class**
The system-managed storage (SMS) management class for active data sets. Leave blank for the default management class for your site. In most cases you do not need to specify this field. If uncertain, consult your site's storage administrator.

**Storage Class**
The SMS storage class for active data sets. Leave blank to use the default storage class for your site.

**Volume Serial**
The volume serial number of the direct access storage device for the active journal data sets. Leave blank for the default volume serial.

**Device Type**
Specify the device type, such as TAPE or SYSDA, for the active data sets.

**Data Class**
The data class for active data sets. In most cases you do not need to specify this field.

**Space Units**
Enter one of the following:

**TRKS** Express the data set size in Tracks

**CYLS** Express the data set size in Cylinders

**Primary Quantity**
The primary allocation quantity in tracks or cylinders as indicated in the Space Units field. Enter a numeric value greater than zero.

**Overflow**
Determine whether IMS Connect Extensions will use an overflow data set if the Archive Manager falls behind and all Active Journal data sets become full. If the overflow data set also becomes full then the action indicated in the Action Flag will be taken.

**Overflow Size**
The size of the overflow data set as a multiple of the Primary Quantity. Enter a value in the range 1 - 99. The default value is 4 (that is, 4 times the Primary Quantity).

**Job Statement Information**
Specify the Job Card to be used when the Archive Manager is submitted.

Lines 2 and 3 are optional and are not used if blank. The ISPF interface will not validate the job statement information.

**Archive JCL skeleton**

Specify the member name that contains the skeleton JCL to execute the Archive Manager. The name must be entered as a fully qualified data set and member name, all enclosed in quotation marks.

### Primary command

**RESET**

Reset the Active Journal.

Occasionally the Archive utility may become out of sync with the Active Journal and be unable to archive records to the Archive Journal data sets. The RESET command will resync the Archive utility and archiving of records will start again.

**Notes:**

1. You can only run the **RESET** command when IMS Connect is not running.
2. Any event records that have not been archived will be discarded.

**Related concepts**:

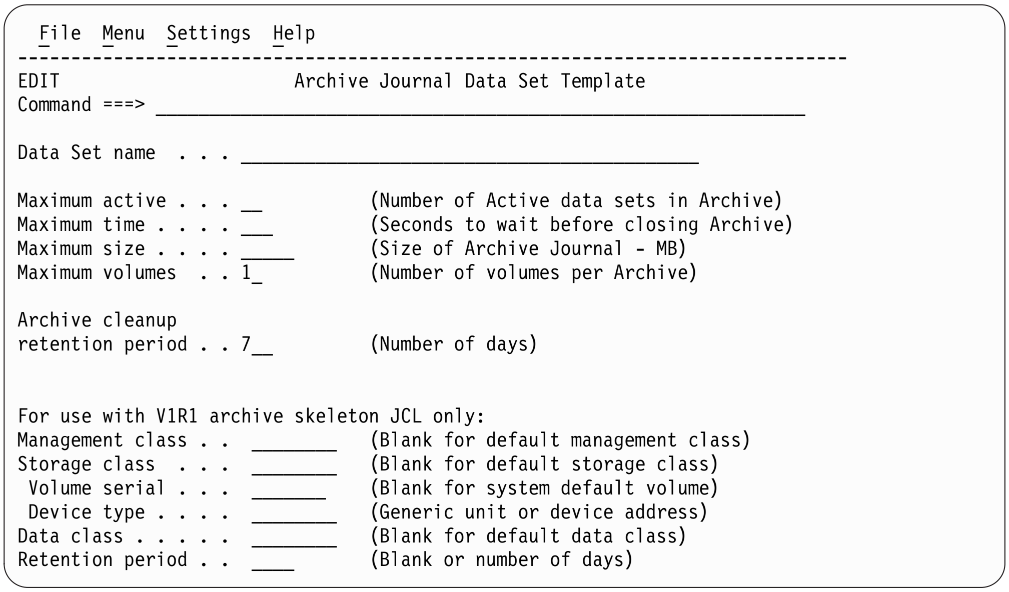
"**Active journals and archive journals**" on page 55
Event records are first written to active journal data sets and subsequently copied by the Archive Manager task to archive journal data sets.

**Related reference**:

"Archive Manager utility" on page 337
The journal archive task submits archive jobs when the active journal is full or when a command is issued to switch it. You can create a JCL skeleton to generate an archiving job using CEXJARC, or you can manually submit a complete archive job.

## Archive Journal

If Archive Manager is enabled, IMS Connect Extensions copies event records from Active Journal data sets when they become full to Archive Journal data sets on tape or DASD.



```
  File  Menu  Settings  Help
------------------------------------------------------------------------------
EDIT                      Archive Journal Data Set Template
Command ===> ______________________________________________________________

Data Set name  . . . ____________________________________________

Maximum active . . . __         (Number of Active data sets in Archive)
Maximum time . . . . ___        (Seconds to wait before closing Archive)
Maximum size . . . . _____      (Size of Archive Journal - MB)
Maximum volumes  . . 1_         (Number of volumes per Archive)

Archive cleanup
retention period . . 7__        (Number of days)


For use with V1R1 archive skeleton JCL only:
Management class . .  ________  (Blank for default management class)
Storage class  . . .  ________  (Blank for default storage class)
 Volume serial . . .  _______   (Blank for system default volume)
 Device type . . . .  ________  (Generic unit or device address)
Data class . . . . .  ________  (Blank for default data class)
Retention period . .  ____      (Blank or number of days)
```

*Figure 73. Archive Journal Data Set Template panel*

**Data Set name**

Specify the archive data set name. To generate a unique data set name, enter either a generation data group (GDG) data set or a name containing symbols.

- For a GDG data set, use the name of an existing GDG base and "(+1)" to indicate a new generation. For example:

  ```
  JOURNAL.GDGBASE(+1)
  ```

- For a name containing symbols such as date and time stamps or the IMS Connect system ID, see "Naming archive journal data sets" on page 62.

**Note:** The **Warning override** profile setting determines what happens if an invalid archive JCL data set is detected during definition maintenance. When this option is set, a pop-up window will be displayed to ask you whether to override the warning and continue. When this option is not set, you must correct the value to proceed.

**Maximum active**

The maximum number of Active Journals that can be written to an Archive data set. When the number is reached, the Archive data set will be closed and a new Archive data set will be started.

You can specify 2 - 32 data sets. The default is 3 Active Journal data sets.

**Maximum time**

The maximum time, in seconds, that the Archive Manager waits before checking to see if more Active journals become available for archiving.

Enter 0 - 120 seconds or leave blank.

**Maximum size**

The maximum size, in megabytes (MB), of an Archive data set. When the size is reached, the Archive data set will be closed and a new Archive data set will be written.

Enter 0 - 32767 megabytes or leave blank for unlimited.

**Maximum volumes**

The number of tape volumes that an Archive data set can span. This value is ignored if Archive data sets are written to DASD devices.

If the specified number of volumes has been reached, the Archive data set will be closed and a new Archive data set will be created.

Enter 1 - 99 volumes. The default is 1 volume.

**Archive cleanup retention period**

The number of days before an archive data set can be deleted or uncataloged by the archive cleanup utility and its name deleted from the repository.

Specify 1 - 365 days; the default is 7 days.

## — Legacy (V1.1) skeleton JCL only —

IMS Connect Extensions refers to the following fields only if you use an archive JCL skeleton from IMS Connect Extensions V1R1. It is recommended that you update JCL skeletons from V1R1 rather than use these fields.

**Management Class**

The system-managed storage (SMS) management class for archive data sets. Leave blank for the default management class for your site. In most cases you do not need to specify this field. If uncertain, consult your site's storage administrator.

**Storage Class**

The SMS storage class for archive data sets. Leave blank to use the default storage class for your site.

**Volume Serial**

The volume serial number of the direct access storage device for the archive journal data sets. Leave blank for the default volume serial.

**Device Type**

Specify the device type, such as TAPE or SYSDA, for the archive data sets.

**Data Class**

The data class for archive data sets. In most cases you do not need to specify this field.

**Retention Period**

Specify the retention period for the archive data set. The retention period is the number of days from when the archive data set is created to its expiration date. This field only applies to tape data sets. Enter 1 - 9999 days. Leave blank to set no expiration.

**— End of Legacy (V1.1) skeleton JCL only —**

**Related concepts**:

“**Active journals and archive journals**” on page 55
Event records are first written to active journal data sets and subsequently copied by the Archive Manager task to archive journal data sets.

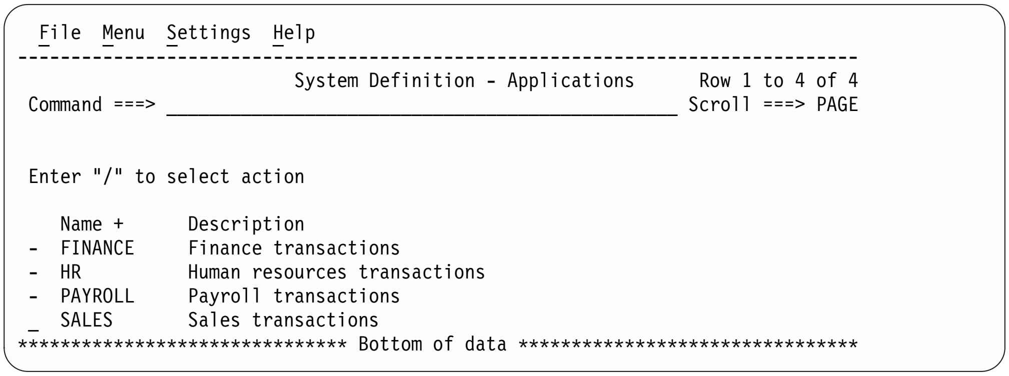
**Related reference**:

“Archive Journal cleanup utility” on page 335
When an archive data set is created, IMS Connect Extensions enters its name in the repository. The archive journal cleanup utility deletes from the repository archive data sets that have expired.

## System Definition - Applications

When you select **Define Applications for** ***sysname*** from the System Definitions panel, you are prompted to list the applications controlled by the system.

Applications associate the IMS Connect system with the transaction codes it can route, and are used for transaction routing and other transaction level features.



```
  File  Menu  Settings  Help
--------------------------------------------------------------------------------
                          System Definition - Applications      Row 1 to 4 of 4
 Command ===> ________________________________________________ Scroll ===> PAGE


 Enter "/" to select action

    Name +      Description
 -  FINANCE     Finance transactions
 -  HR          Human resources transactions
 -  PAYROLL     Payroll transactions
 _  SALES       Sales transactions
******************************* Bottom of data ********************************
```

*Figure 74. System Definition - list of applications*

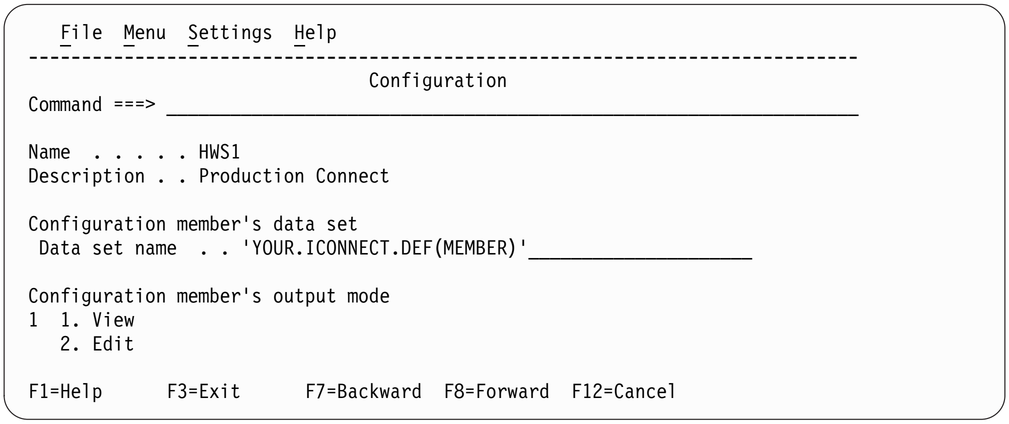
Use the **I** line action to insert a new application. In the Name field, enter a valid application name or use the Prompt function key (F4) to select from a list of defined applications.

Use the **D** line action to delete an application from the list.

Press the Exit function key (F3) to save and return to the System Definition screen.

## IMS Connect configuration

The Configuration panel allows you to view and edit the IMS Connect configuration member.



```
   File  Menu  Settings  Help
 -------------------------------------------------------------------------------
                                 Configuration
 Command ===> _________________________________________________________________

 Name  . . . . . HWS1
 Description . . Production Connect

 Configuration member's data set
  Data set name  . . 'YOUR.ICONNECT.DEF(MEMBER)'_____________________

 Configuration member's output mode
 1  1. View
    2. Edit

 F1=Help      F3=Exit      F7=Backward  F8=Forward  F12=Cancel
```

*Figure 75. IMS Connect configuration*

**Description**
: The description of the System Definition associated with this IMS Connect Configuration member.

**Configuration member's data set**

This field displays the IMS Connect Configuration data set that is currently selected for editing or browsing.

The Configuration data set is displayed as a fully qualified data set name and member, enclosed in quotation marks.

**Output Mode**

Enter 1 to view the configuration member.

Enter 2 to edit the configuration member.

## Defining user exits

The User Exits Definition allows you to configure user message exits. You can define the standard IBM-supplied exits to IMS Connect Extensions using a **LOAD** command.

**Description**

A description of this User Exit.

**Activate Exit Definition**

This field allows you to activate this user exit. If the user exit is not active (or not defined) then no IMS Connect Extensions features can execute for incoming messages processed by this user exit. Acceptable field values are:

**/** The user exit is active

**blank** The user exit is not active. IMS Connect Extensions services cannot execute for incoming messages processed by this exit. Services such as Event Collection, Transaction Routing, Workload balancing, Pacing and Security will not function.

**Choose how IRM offsets are determined**

Specifies how to obtain the offsets within the user portion of the IRM header of the following fields: user ID, password or PassTicket, SAF group name, and optional application name. IMS Connect Extensions passes these fields to the external security manager to perform user ID verification. The acceptable values are:

**1** For IMS Connect sample exits supplied by IBM, use the default offsets referenced in the exit.

**2** Use this option to specify custom offsets in the following cases:

- If you use your own message processing exits.
- If you use the supplied exits but have changed offsets for any of the fields that are required for security processing.

Offsets are expressed in decimal, start at zero, and do not include the 4-byte IRM prefix. The offsets have to be separated by at least 8 bytes. Overlapping offsets are not allowed.

**3** Specify the name of a user exit that returns the security-related IRM fields. The security exit is called as each message is received and processed by IMS Connect Extensions. Use this option in the following cases:

- The security fields are not located at fixed offsets within the IRM. For example, if the field values are generated dynamically.
- The field values are not in a form your security manager can recognize. For example, if you use your own password encryption scheme.

For more information on creating a security preprocessing exit, see "Security preprocessing exit" on page 177.

**Use a 4-byte length prefix for messages instead of 2 bytes**
The acceptable values are:

**/** The user exit processes messages with a 4-byte length prefix

**blank** The user exit processes messages with a 2-byte length prefix

**Extend length of RSM for error information**
Specify whether IMS Connect Extensions can append enhanced information on some errors to the Response Status Message (RSM) that is returned to the remote client. For example, for a password error, the additional information might indicate whether the password was incorrect or expired.

This option is only available to user exits that support the RSM protocol and build messages with a 4-byte length prefix.

**Note:** Activating this feature changes the length of the RSM. Some remote clients may have processing that depends on the RSM length.

**Message ID support**
These fields allow you to specify whether MSG ID1 and MSG ID2 for this user exit are in ASCII or EBCDIC.

The acceptable values are:

**1** ASCII

**2** EBCDIC

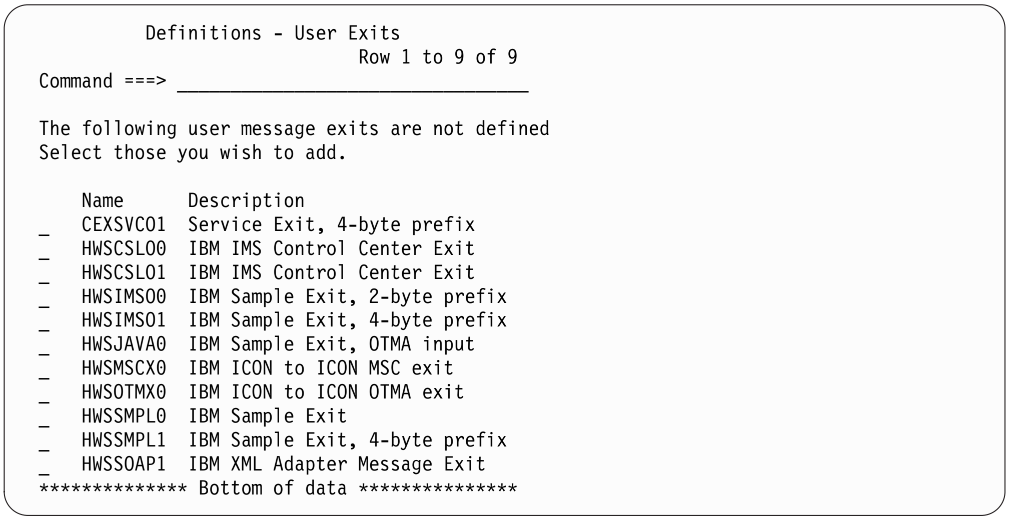
**Message translation**
These fields allow you to specify custom code page translation for the exit. See Chapter 18, "Customizing IMS Connect message translation," on page 183 for details.

## Loading IBM user exits

You can invoke the `LOAD` from the User Exits list panel as a primary command or from the Action Bar.

**Command Line**
In addition to the standard commands supported on the List panel (see "Defining IMS Connect systems" on page 189, the User Exit list panel recognizes the following command:

**LOAD**
The `LOAD` command invokes the user exit load facility, as shown in the following panel.



```
          Definitions - User Exits
                              Row 1 to 9 of 9
Command ===> _________________________________

The following user message exits are not defined
Select those you wish to add.

    Name      Description
_   CEXSVC01  Service Exit, 4-byte prefix
_   HWSCSLO0  IBM IMS Control Center Exit
_   HWSCSLO1  IBM IMS Control Center Exit
_   HWSIMSO0  IBM Sample Exit, 2-byte prefix
_   HWSIMSO1  IBM Sample Exit, 4-byte prefix
_   HWSJAVA0  IBM Sample Exit, OTMA input
_   HWSMSCX0  IBM ICON to ICON MSC exit
_   HWSOTMX0  IBM ICON to ICON OTMA exit
_   HWSSMPL0  IBM Sample Exit
_   HWSSMPL1  IBM Sample Exit, 4-byte prefix
_   HWSSOAP1  IBM XML Adapter Message Exit
************** Bottom of data ***************
```

*Figure 76. Load IBM User Exits panel*

**Action Field**
The action field allows you to select a User Exit from the list of IBM user message exits and create a User Exit definition.

The acceptable values are:

**S** Select this entry and create a User Exit definition

When all required user exits are selected, press Enter.

## Defining datastores

The Datastore Definition allows you to define the datastores controlled by IMS Connect.

**Description**
A description of this datastore.

**Datastore Group**
The name of a datastore group that this datastore belongs to. Use datastore groups for primary datastore routing, pacing and reporting.

This field is required if using the datastore as a primary for an IMS Connect system. Use the Prompt function key (F4) to get a list of defined Datastore Groups.

**Use as Primary Datastore for IMS Connect System**
Set the datastore as a Primary Datastore for a selected IMS Connect system. Messages are routed to this datastore if the original datastore set in the message is the same as the name of the datastore group and the message is handled by the selected system.

**Note:** Within a datastore group, only one datastore can be the active primary for an IMS Connect system.

**Activate Pacing**
This field allows you to activate and deactivate the Transaction Pacing feature for this datastore.

Transaction Pacing monitors incoming transaction request rates according to thresholds assigned to this datastore. Based upon these thresholds, IMS Connect Extensions either issues warning messages or rejects transactions. The acceptable field values are:

**/** Pacing is activated for this datastore. Where pacing threshold settings have been defined, these will be honored. Warning messages or transaction rejections may occur.

**Note:** Pacing must also be active at the System Definition level.

**blank** Transaction Pacing is not active for this datastore

**Warning Threshold**

This field defines the Warning Transaction threshold for this datastore. At this threshold, IMS Connect Extensions issues a warning message to the IMS Connect Extensions message log and z/OS operator console indicating this threshold level has been exceeded.

At the datastore level, warning thresholds use the Interval Count as defined in the System Definition.

The acceptable field values are any number less than 999999. A value of 0 deactivates warning processing for this datastore.

**Reject Threshold**

This field defines the Reject Transaction rate for this datastore. At this threshold IMS Connect Extensions rejects incoming messages. A message is issued to the IMS Connect Extensions message log and z/OS operator console indicating maximum pacing rates have been reached.

At the datastore level, reject thresholds use the Interval Count as defined in the System Definition.

The acceptable field values are any number less than 999999. A value of 0 deactivates reject processing for this datastore.

**Activate Transaction Routing**

This field allows you to activate and deactivate the Transaction Routing feature for this datastore.

Transaction Routing allows IMS Connect Extensions to alter the target IMS datastore that will process the incoming transaction request. See “Candidate datastore list” on page 158 for detailed examples of how Transaction Routing options work.

If transaction routing is inactive then all transaction affinity and datastore affinity rules are bypassed and the original target datastore defined on the incoming transaction request is used.

The acceptable field values are:

**/** Transaction Routing is active for this datastore. IMS Connect Extensions will alter the target datastore if the combinations of transaction affinity and datastore affinity allow.

**blank** Transaction Routing is not active for this datastore.

**Route transactions to**

This field allows you to define which datastores (if any) can share work with the datastore if Transaction Routing is active for this datastore.

The acceptable field values are:

**1** All datastores in this IMS Connect system can support this datastore.

**2** One datastore only. You must supply the name of a defined Datastore. Use the Prompt function key (F4) to get a list of defined datastores.

**3** Any datastore defined in this Datastore Group. You must supply the name of a defined Datastore Group. Use the Prompt function key (F4) to get a list of defined Datastore Groups.

**4** Any datastore defined in this Affinity List. You must supply the name of a defined Affinity List. Use the Prompt function key (F4) to get a list of defined Affinity Lists.

**Also route Send Only requests**

These fields determine how messages of type Send Only are routed. See "Routing Send Only messages with transaction routing" on page 162 for details.

The acceptable field values are:

**/** Allow routing of Send Only requests for this datastore. Send Only requests with this datastore as the original destination ID are candidates for routing. The actual decision to route depends on the transaction associated with the Send Only request. Transaction Routing must be activated for this datastore for the routing of Send Only requests to be actioned.

**Note:** Routing of related Resume TPIPE requests relies on Super Member Routing.

**blank** Send Only requests are not candidates for routing.

**Activate Super Member Routing**

Select this feature to allow routing of Resume TPIPE (RTPIPE) requests. It applies only to IMS Connect systems in a Super Member configuration (defined with SMEMBER=*xxxx*). See "Routing Send Only messages with transaction routing" on page 162 for details.

The acceptable field values are:

**/** Allow routing of RTPIPE requests to an Affinity List of eligible datastores. The routing of RTPIPE requests relies on Super Member routing. Rules-based routing, primary datastore routing, and transaction routing do not apply.

**blank** RTPIPE requests are not candidates for routing.

**Route Resume TPIPE requests to Affinity List**

Specify the name of an Affinity List containing the datastores that are candidates for routing of Resume TPIPE (RTPIPE) requests when Super Member routing is activated for this datastore. When pairing Send Only and Resume TPIPE requests, ensure that the Affinity List for Super Member routing is compatible with that for transaction routing.

**Capacity Weight Rating**

This field allows you to define the Capacity Weighting for this datastore. The Rating is used by workload balancing to reroute incoming messages to alternate datastores.

Acceptable field values are numbers in the range 1 - 100 and 0. A value of zero has a special meaning. It indicates that the datastore is not a candidate for routing.

The higher the rating number, the more workload this datastore can handle. Weightings are relative to other datastore definitions.

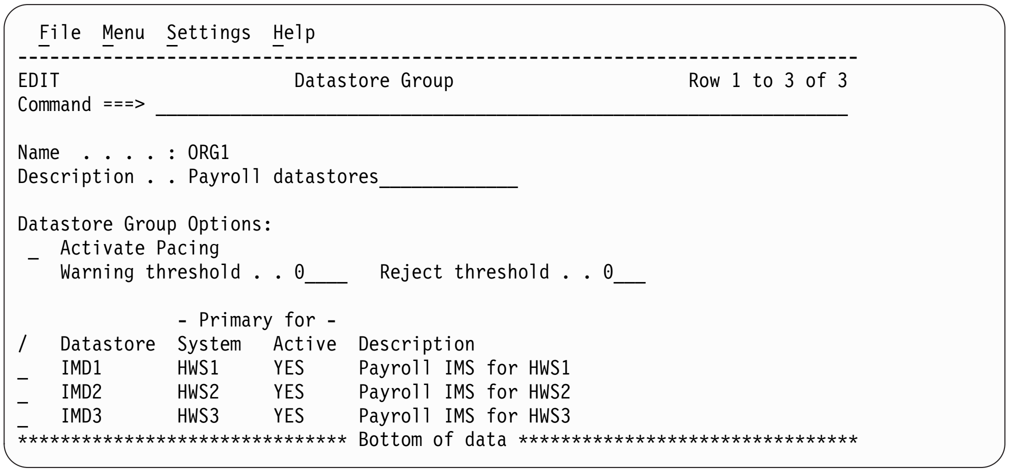
**Examples:**

1. If you have 3 datastores of equal capacity you could define them all with the same rating number (for example, 1 or 50 or 99). IMS Connect Extensions would apportion incoming messages equally amongst all 3 datastores.
2. If you have 2 datastores and one has twice the capacity as the other, you could define them with weightings of 1 and 2, or 10 and 20, or 40 and 80, and IMS Connect Extensions will apportion twice as much work to the datastore with the higher weighting.

## Defining datastore groups

Datastore Group Definition allows you to group datastores controlled by IMS Connect Extensions.

The panel lists all datastores that are part of the datastore group.



```
  File  Menu  Settings  Help
-------------------------------------------------------------------------------
EDIT                      Datastore Group                      Row 1 to 3 of 3
Command ===> _________________________________________________________________

Name  . . . . : ORG1
Description . . Payroll datastores_____________

Datastore Group Options:
 _  Activate Pacing
    Warning threshold . . 0____   Reject threshold . . 0___

           - Primary for -
/   Datastore  System   Active  Description
_   IMD1       HWS1     YES     Payroll IMS for HWS1
_   IMD2       HWS2     YES     Payroll IMS for HWS2
_   IMD3       HWS3     YES     Payroll IMS for HWS3
****************************** Bottom of data ********************************
```

*Figure 77. Datastore Group detail panel*

**Command Line**

The Datastore Group Edit panel recognizes the following additional primary commands:

**LOCATE**

Locate datastore definition.

**NEW** Create a new datastore using the current datastore group.

**REFRESH**
Refresh list of datastores.

**RESET**
Clear line actions.

**SORT** ***column***
Sort datastores by column.

**Description**
A description of this Datastore Group.

**Pacing**

This field allows you to activate and deactivate the Transaction Pacing feature for this Datastore Group.

Transaction Pacing monitors incoming transaction request rates according to thresholds assigned to this Datastore Group. Based upon these thresholds, IMS Connect Extensions either issues warning messages or rejects transactions.

The acceptable field values are:

**/** Transaction Pacing is activated for this Datastore Group. Where pacing threshold settings have been defined, these will be honored. Warning messages or transaction rejections may occur.

**Note:** Pacing must also be active at the System Definition level.

**blank** Transaction Pacing is not active for this Datastore Group.

**Warning Threshold**
This field defines the Warning Transaction threshold for this Datastore Group. At this threshold, IMS Connect Extensions issues a warning message to the IMS Connect Extensions message log and z/OS operator console indicating this threshold level has been exceeded.

At the Datastore Group level, warning thresholds use the Interval Count as defined in the System Definition.

The acceptable field values are any number less than 999999. A value of 0 deactivates warning processing for this Datastore Group.

**Reject Threshold**
This field defines the Reject Transaction rate for this Datastore Group. At this threshold IMS Connect Extensions rejects incoming messages. A message is issued to the IMS Connect Extensions message log and z/OS operator console indicating maximum pacing rates have been reached.

At the Datastore Group level, reject thresholds use the Interval Count as defined in the System Definition.

The acceptable field values are any number less than 999999. A value of 0 deactivates reject processing for this Datastore Group.

Here are the line actions you can use for the list of datastores in this datastore group:

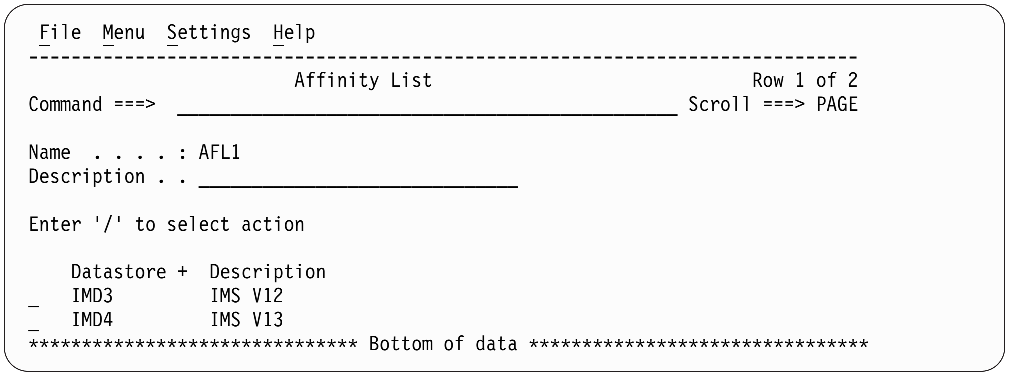
**Edit** Edit datastore definition (datastore group cannot be changed)

**View** View datastore definition

**Delete from datastore group**
  Remove datastore from group

## Defining affinity lists

The Affinity List Definition allows you to group datastores into logical groups. Affinity lists are used by Transaction Routing.



```
 File  Menu  Settings  Help
------------------------------------------------------------------------------
                         Affinity List                               Row 1 of 2
Command ===>  _______________________________________________ Scroll ===> PAGE

Name  . . . . : AFL1
Description . . _____________________________

Enter '/' to select action

    Datastore +  Description
_   IMD3         IMS V12
_   IMD4         IMS V13
******************************* Bottom of data ********************************
```

*Figure 78. Affinity List detail panel*

**Description**
  A description of this Affinity List.

**Option Field**
  You can issue an option against one datastore definition at a time or insert a new datastore definition. The acceptable values are:

  **/** display a pop-up window of options

  **I** insert a new datastore

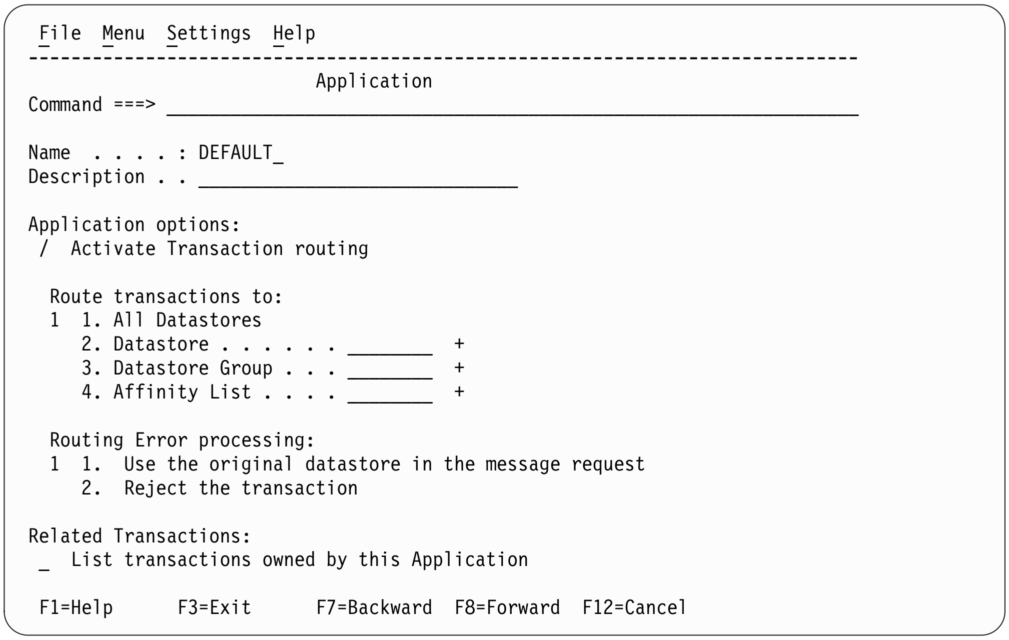
  **D** delete this datastore

**Datastore Name**
  Must be a valid datastore name. Use the Prompt function key (F4) to get a list of defined datastores.

## Defining applications

The Application Definition allows you to group the transactions controlled by IMS Connect.

You define an application to associate IMS Connect Extensions application and transaction definitions to a system. This enables those definitions to be used in conjunction with transaction routing and the other advanced features in the transaction definition, such as transaction expiration and client ID cancellation.



```
 File  Menu  Settings  Help
------------------------------------------------------------------------------
                           Application
Command ===> _________________________________________________________________

Name  . . . . : DEFAULT_
Description . . ______________________________

Application options:
 /  Activate Transaction routing

  Route transactions to:
  1  1. All Datastores
     2. Datastore . . . . . . ________  +
     3. Datastore Group . . . ________  +
     4. Affinity List . . . . ________  +

  Routing Error processing:
  1  1.  Use the original datastore in the message request
     2.  Reject the transaction

Related Transactions:
 _  List transactions owned by this Application

 F1=Help      F3=Exit      F7=Backward  F8=Forward  F12=Cancel
```

*Figure 79. Application detail panel*

**Description**
A description of this Application.

**Activate Transaction routing**
This field allows you to activate and deactivate the Transaction Routing feature for this Application.

Transaction Routing allows IMS Connect Extensions to alter the target IMS datastores that will process the incoming transaction request. See "Candidate datastore list" on page 158 for detailed examples of how Transaction Routing options work.

If transaction routing is inactive then all transaction affinity and datastore affinity rules are bypassed and the original target datastore defined on the incoming transaction request is used.

The acceptable field values are:

**/** Transaction Routing is active for this Application. IMS Connect Extensions will alter the target datastore if the combinations of transaction affinity and datastore affinity allow.

**blank** Transaction Routing is not active for this Application

**Route transactions to**
This field allows you to define which datastores (if any) can run the transactions defined in the Application if Transaction Routing is active for the transaction.

The acceptable field values are:

**1** All datastores in this IMS Connect system can support this Application.

**2** One datastore only. You must supply the name of a defined Datastore. Use the Prompt function key (F4) to get a list of defined datastores.

**3** Any datastore defined in this Datastore Group. You must supply the name of a defined Datastore Group. Use the Prompt function key (F4) to get a list of defined Datastore Groups.

**4** Any datastore defined in this Affinity List. You must supply the name of a defined Affinity List. Use the Prompt function key (F4) to get a list of defined Affinity Lists.

**Routing Error processing**
This field defines the processing options when a datastore is not available to process the incoming transactions.

The acceptable field values are:

**1** Use the original destination datastore in the message

**2** Reject the transaction

**Related Transactions**
Enter / (slash) to display the list of transactions that are defined in this application.

## Defining transactions

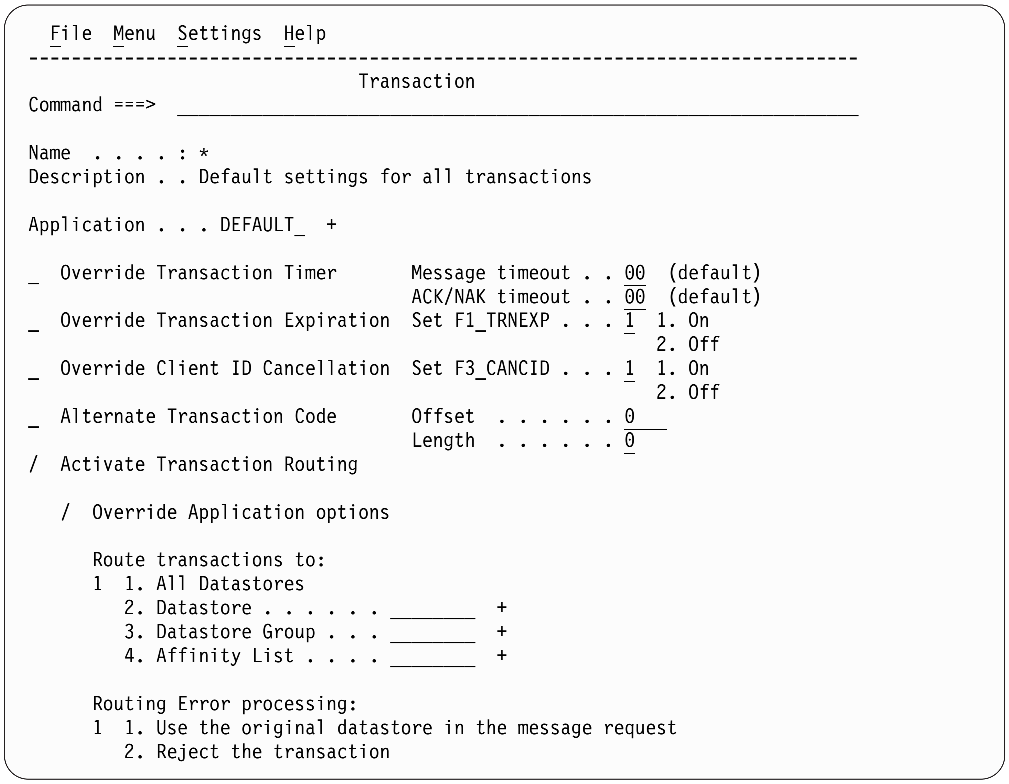
The Transaction Definition allows you to define transactions to be routed by IMS Connect Extensions through the transaction routing feature.

With IMS Connect Extensions, transactions can be defined with a prefix. This allows you to define a set of rules for a group of similarly named transactions; for example PAY* for all payroll transactions. You can even define a transaction "*" as the default definition for all transactions.

You can define a special transaction named $NOTRAN$. For messages without a valid transaction code the $NOTRAN$ record is checked. If the $NOTRAN$ record exists, it is used to route the message, otherwise the original data set is used. Note that no wild card searching can be performed, so the record must match exactly.

If Transaction Routing is set to Active, IMS Connect Extensions will try to match the full transaction name on the incoming message with a Transaction definition. If it is unsuccessful, the granularity of the match is reduced one character at a time until a match is found. If no match is found then the incoming message cannot be routed.

For example, if you have defined 3 transaction definitions PAR*, PART and PART* an incoming message for transaction PART will match the PART definition but it will not match PART* or PAR*



```
  File  Menu  Settings  Help
------------------------------------------------------------------------------
                               Transaction
Command ===> _________________________________________________________________

Name  . . . . : *
Description . . Default settings for all transactions

Application . . . DEFAULT_  +

_  Override Transaction Timer      Message timeout . . 00  (default)
                                   ACK/NAK timeout . . 00  (default)
_  Override Transaction Expiration  Set F1_TRNEXP . . . 1  1. On
                                                          2. Off
_  Override Client ID Cancellation  Set F3_CANCID . . . 1  1. On
                                                          2. Off
_  Alternate Transaction Code      Offset  . . . . . . 0___
                                   Length  . . . . . . 0
/  Activate Transaction Routing

   /  Override Application options

      Route transactions to:
      1  1. All Datastores
         2. Datastore . . . . . . ________  +
         3. Datastore Group . . . ________  +
         4. Affinity List . . . . ________  +

      Routing Error processing:
      1  1. Use the original datastore in the message request
         2. Reject the transaction
```

*Figure 80. Transaction detail panel*

**Description**

A description of this transaction.

**Application**

The name of an Application that this transaction belongs to. A transaction must be part of an Application. This is a mandatory field.

Type any valid Application name or use the Prompt function key (F4) to get a list of defined Applications.

**Override Transaction Timer**

The transaction timer feature allows you to set a timeout value in the IRM header of messages for this transaction or group of transactions.

Activating **Override Transaction Timer** means that the timeout values specified in the transaction definition will override the IRM setting.

When you override the transaction timer, you must also specify a message timeout value and optionally an ACK/NAK timeout value in IRM timer format. The acceptable field values are:

**00** Default

**E9** No timer

**FF** Wait indefinitely

***xx*** A hexadecimal representation of the timeout value in minutes, seconds, or hundredths of a second

For a full description of how to specify values in IRM timer format, refer to the Transaction Timer panel in the ISPF Help or see the "Timer interval specifications" topic in the *IMS Communications and Connections Guide*.

**MSGTO**=*xx*
Message timeout value.

**ACKTO**=*xx*
ACK/NAK timeout value (Optional).

**Override Transaction Expiration**
IRM_F1_TRNEXP is a flag on the IRM that determines whether the transaction expiration feature is active or inactive. If active, then IMS Connect sets the expiration time for the input transaction. Activating **Override Transaction Expiration** means that the value specified for F1_TRNEXP in the transaction definition (either ON or OFF) will override the IRM setting.

**F1_TRNEXP=ON | OFF**
ON indicates that transaction expiration is enabled for this transaction.

**Override Client ID Cancellation**
IRM_F3_CANCID is a flag on the IRM that determines whether the Client ID Cancellation feature is active or inactive. If it is active, then a message that specifies the same client ID as that of an active session will cancel the original session and then run. Activating **Override Client ID Cancellation** means that the value specified in the transaction definition (either ON or OFF) will override the IRM setting.

**F3_CANCID=ON | OFF**
ON indicates that cancellation of duplicate sessions is enabled for this transaction.

**Alternate Transaction Code**
Some applications use a single or common IMS transaction code, and store the actual transaction code elsewhere in the transaction payload. For example, OMEGAMON refers to the common transaction code as the umbrella transaction and the alternate transaction code as the user code.

If the Alternate Transaction Code feature is activated for this definition, IMS Connect Extensions uses the associated Offset and Length fields to locate the alternate transaction code within the transaction payload.

The offset value can be anywhere in the range 6 - 9980, but it must be found in the first application segment of the transaction. The length must be in the range 1 - 8.

The options are:

**/** Use the alternate transaction code at the specified offset within the transaction payload.

**blank** Do not look for an alternate transaction code in this definition.

**Activate Transaction Routing**
This field allows you to activate and deactivate the Transaction Routing feature for this transaction.

Transaction Routing allows IMS Connect Extensions to alter the target IMS datastores that will process the incoming transaction request. See "Candidate datastore list" on page 158 for detailed examples of how Transaction Routing options work.

If transaction routing is inactive then all transaction affinity and datastore affinity rules are bypassed and the original target datastore defined on the incoming transaction request is used.

The acceptable field values are:

**/** Transaction Routing is active for this transaction. IMS Connect Extensions will alter the target datastore if the combinations of transaction affinity and datastore affinity allow.

**blank** Transaction Routing is not active for this transaction

**Override Application options**
By default Transaction Routing options are taken from the Application Definition. This field allows you to override the Application options for Transaction Routing for this transaction definition.

The acceptable field values are:

**/** Override the Application options using the options defined in the **Route transactions to** and **Routing Error processing** fields.

**blank** Do not override. Use the Application options for Transaction Routing

**Route transactions to**
This field allows you to define which datastores (if any) can run this Transaction if Transaction Routing is active for this transaction.

The acceptable field values are:

**1** All datastores in this IMS Connect system can support this transaction.

**2** One datastore only. You must supply the name of a defined Datastore. Use the Prompt function key (F4) to get a list of defined datastores.

**3** Any datastore defined in this Datastore Group. You must supply the name of a defined Datastore Group. Use the Prompt function key (F4) to get a list of defined Datastore Groups.

**4** Any datastore defined in this Affinity List. You must supply the name of a defined Affinity List. Use the Prompt function key (F4) to get a list of defined Affinity Lists.

**Routing Error processing**
This field defines the processing options when a datastore is not available to process incoming transactions. This option only applies if Transaction Routing is active, when an incoming message has been routed to a new destination datastore and the destination is not available.

The acceptable field values are:

**1** Use the original destination datastore in the message

**2** Reject the transaction

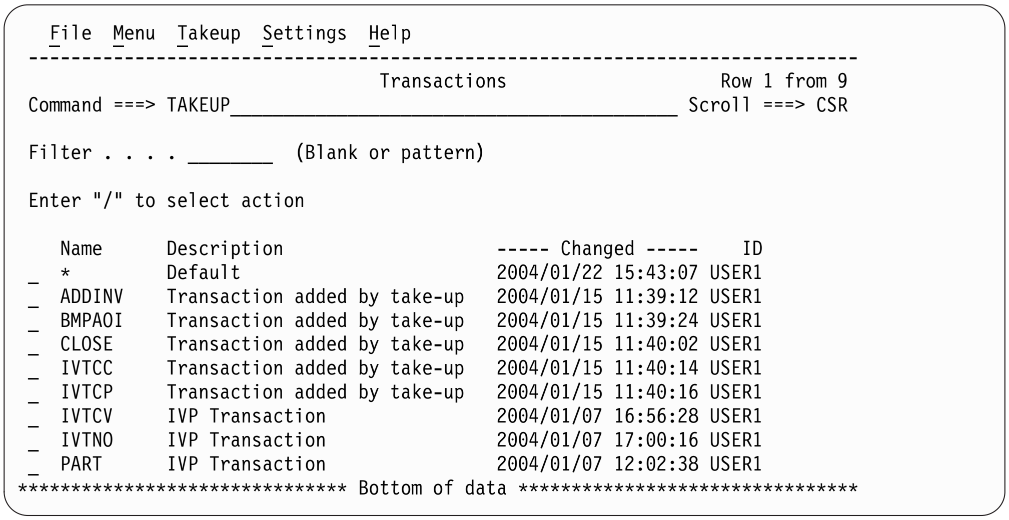
**Related concepts**:

"Transaction routing" on page 154
Transaction routing is the process by which IMS Connect Extensions selects a destination datastore for an incoming IRM message based on the transaction code and destination ID specified by the message.

## Generating transactions from MODBLKS using takeup

You can define transactions to IMS Connect Extensions using a TAKEUP facility which reads transactions from your IMS MODBLKS libraries.

You can invoke this facility from the Action Bar or the **TAKEUP** command on the Transactions list panel.



```
  File  Menu  Takeup  Settings  Help
------------------------------------------------------------------------------
                                 Transactions                    Row 1 from 9
Command ===> TAKEUP__________________________________________ Scroll ===> CSR

Filter . . . . ________  (Blank or pattern)

Enter "/" to select action

   Name      Description                    ----- Changed -----    ID
_  *         Default                        2004/01/22 15:43:07 USER1
_  ADDINV    Transaction added by take-up   2004/01/15 11:39:12 USER1
_  BMPAOI    Transaction added by take-up   2004/01/15 11:39:24 USER1
_  CLOSE     Transaction added by take-up   2004/01/15 11:40:02 USER1
_  IVTCC     Transaction added by take-up   2004/01/15 11:40:14 USER1
_  IVTCP     Transaction added by take-up   2004/01/15 11:40:16 USER1
_  IVTCV     IVP Transaction                2004/01/07 16:56:28 USER1
_  IVTNO     IVP Transaction                2004/01/07 17:00:16 USER1
_  PART      IVP Transaction                2004/01/07 12:02:38 USER1
******************************* Bottom of data ********************************
```

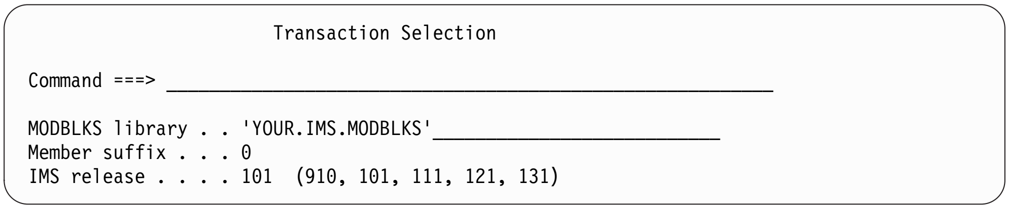
*Figure 81. Transactions list panel*

**Command Line**

In addition to the standard list panel commands (see "Defining IMS Connect systems" on page 189, the Transactions list panel recognizes the following command:

**TAKEUP**

The **TAKEUP** command invokes the transaction takeup facility from your IMS MODBLKS.



```
                       Transaction Selection

Command ===> __________________________________________________________

MODBLKS library . . 'YOUR.IMS.MODBLKS'___________________________
Member suffix . . . 0
IMS release . . . . 101  (910, 101, 111, 121, 131)
```

*Figure 82. Transaction selection, MODBLKS data set panel*

**MODBLKS Library**

The data set containing your IMS MODBLKS.

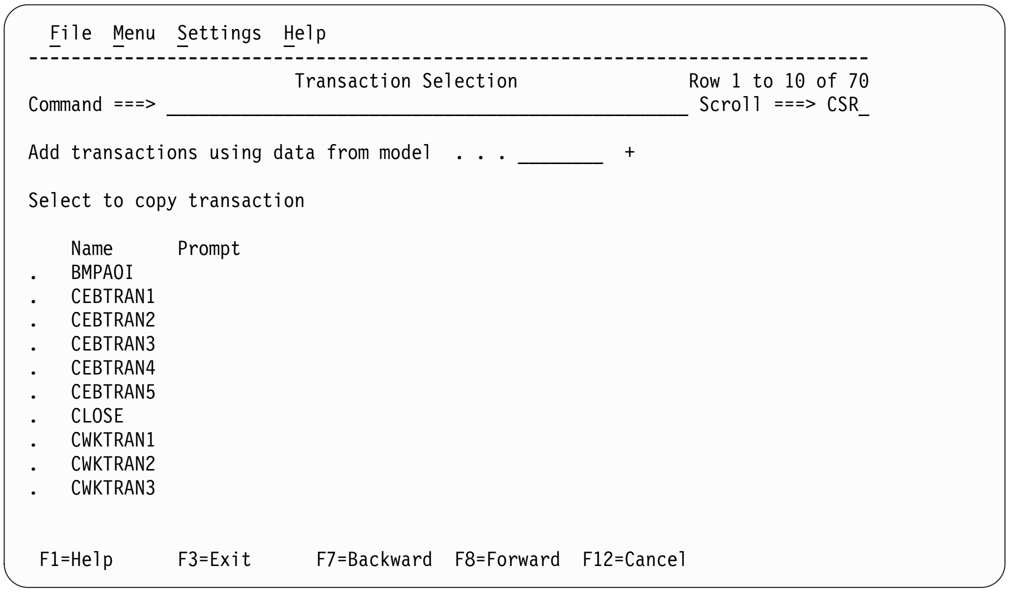
**Member Suffix**

The MODBLKS library member suffix to be used.

**IMS release**

The IMS release.

Press Enter to proceed to the Transaction Selection panel.



```
  File  Menu  Settings  Help
--------------------------------------------------------------------------------
                         Transaction Selection               Row 1 to 10 of 70
Command ===> _________________________________________________ Scroll ===> CSR_

Add transactions using data from model  . . . ________  +

Select to copy transaction

    Name      Prompt
.   BMPAOI
.   CEBTRAN1
.   CEBTRAN2
.   CEBTRAN3
.   CEBTRAN4
.   CEBTRAN5
.   CLOSE
.   CWKTRAN1
.   CWKTRAN2
.   CWKTRAN3


 F1=Help      F3=Exit      F7=Backward  F8=Forward  F12=Cancel
```

*Figure 83. Transaction Selection panel*

This panel lists all transactions defined in the MODBLKS library that are not already defined to IMS Connect Extensions. It allows you to select transactions to add to IMS Connect Extensions. Transactions are added using options modelled on the nominated existing transaction definition.

**Command Line**
The Transaction Selection panel recognizes the following commands:

**SELECT**
The **SELECT** command (which can be entered as SEL or S) selects entries from the list. You can select all entries by entering `s *` on the command line.

**LOCATE**
The **LOCATE** command (which can be entered as LOC or L) scrolls to the entry you specify or the one preceding it on the display if the entry is not found. Syntax is as follows: **LOCATE** *string*, where *string* can be any character string of ten or less characters. If *string* is found, that entry is scrolled to the top of the display. If it is not found, the entry preceding it in the current sort order is scrolled to the top of the display.

**RESET**
Resets line actions.

**Model** This field allows you to add new transaction definitions based on an existing definition.

Acceptable values are an existing transaction definition. Use the Prompt function key (F4) to get a list of defined transactions.

**Action Field**
The action field allows you to copy the entry and create a Transaction definition. The acceptable values are:

**S** Select this entry and create a Transaction definition.

## Defining system groups

A system group is literally a group of IMS Connect systems. After defining a system group, you can use it to quickly select a set of systems (typically a SYSPLEX) and view activity on that group. You can put the same system in more than one group.

To define a system group:

1. From the primary menu, select option 1.8 **System Groups**.

   The System Groups panel appears with a list of all system groups (if there are any).
2. Enter the command:

   ```
   NEW sysgroup
   ```

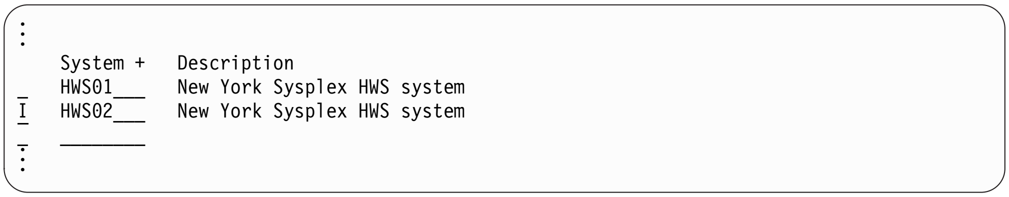
   where *sysgroup* is the 1-8 character name of your system group.

   The System Group edit panel appears.
3. Optionally, type a description for the System Group.
4. Specify which systems are part of this system group:
   a. Position the cursor in the System prompt and press the Prompt function key (F4).

      The System Definitions pop-up window displays a list of all IMS Connect systems that you have defined.
   b. Type an S next to the system that you want to select and press Enter.

      The system and its description appear in the System Group list.
   c. Type an I next to the system and press Enter.

      A new line appears.
   d. Repeat for each IMS Connect system you want to add.



```
⋮
   System +   Description
_  HWS01___   New York Sysplex HWS system
I  HWS02___   New York Sysplex HWS system
_  ________
⋮
```

*Figure 84. Putting systems in a system group*

5. Press the Exit function key (F3).

   The new system group appears in the System Groups list.

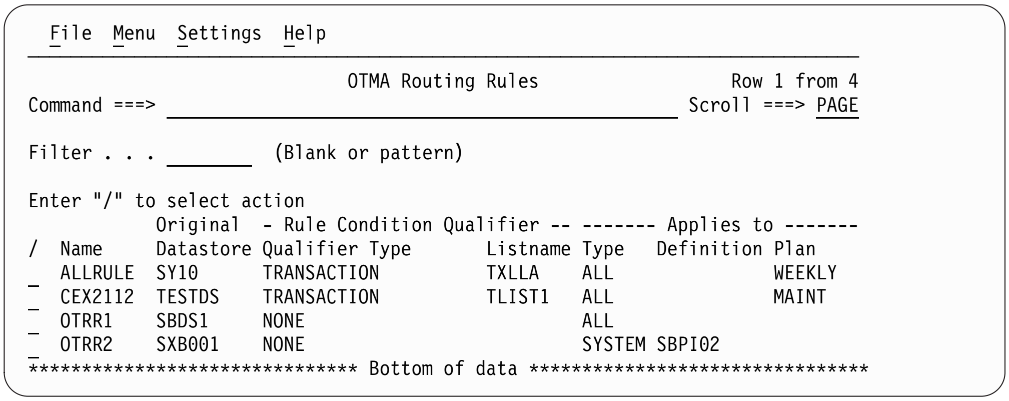
## Defining OTMA routing rules

OTMA routing rules enable you to specify routing behavior for IRM messages. They describe which routing lists to use to route messages that have a given original datastore identifier (DestID).

You can specify different target and fallback routing lists for each message type. A routing rule can be assigned along with other related rules to a routing plan.

You can have one master rule for a given DestID, as well as optional rules that specify different target datastores for a specified list of transaction codes.

From the primary menu, select option 1.9 **OTMA Routing Rules**. The OTMA Routing Rules panel appears with a list of all existing rules (if there are any). You can use the **Filter** field to list only items that match a filter pattern.



```
  File  Menu  Settings  Help
───────────────────────────────────────────────────────────────────────────────
                             OTMA Routing Rules                  Row 1 from 4
Command ===> ________________________________________________ Scroll ===> PAGE

Filter . . . ________  (Blank or pattern)

Enter "/" to select action
            Original  - Rule Condition Qualifier -- ------- Applies to -------
/  Name     Datastore Qualifier Type       Listname Type   Definition Plan
_  ALLRULE  SY10      TRANSACTION          TXLLA    ALL               WEEKLY
_  CEX2112  TESTDS    TRANSACTION          TLIST1   ALL               MAINT
_  OTRR1    SBDS1     NONE                          ALL
_  OTRR2    SXB001    NONE                          SYSTEM SBPI02
******************************* Bottom of data ********************************
```

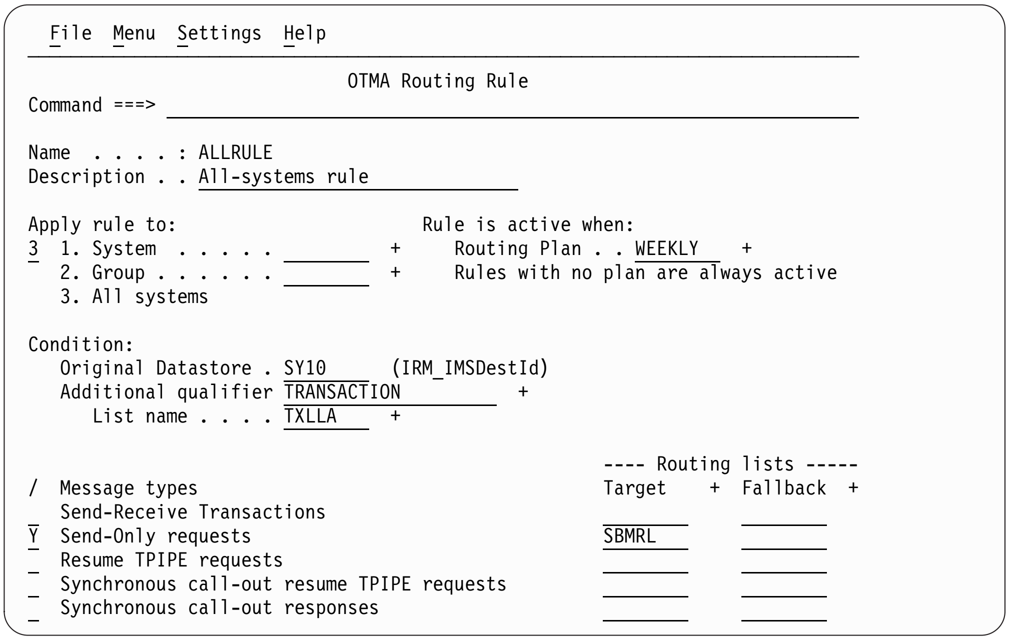
Figure 85. OTMA Routing Rules panel

To add a new routing rule, enter the command:

```
NEW rrule
```

where *rrule* is the 1-8 character name of your routing rule.

The OTMA Routing Rule edit panel appears. In this dialog you will define the routing behavior for messages with a specified DestID.



```
  File  Menu  Settings  Help
───────────────────────────────────────────────────────────────────────────────
                             OTMA Routing Rule
Command ===> _________________________________________________________________

Name  . . . . : ALLRULE
Description . . All-systems rule______________

Apply rule to:                      Rule is active when:
3  1. System  . . . . . ________  +     Routing Plan . . WEEKLY__  +
   2. Group . . . . . . ________  +     Rules with no plan are always active
   3. All systems

Condition:
   Original Datastore . SY10____  (IRM_IMSDestId)
   Additional qualifier TRANSACTION_________  +
      List name . . . . TXLLA___  +

                                                    ---- Routing lists -----
/  Message types                                    Target    +  Fallback  +
_  Send-Receive Transactions                        ________     ________
Y  Send-Only requests                               SBMRL___     ________
_  Resume TPIPE requests                            ________     ________
_  Synchronous call-out resume TPIPE requests       ________     ________
_  Synchronous call-out responses                   ________     ________
```

Figure 86. OTMA Routing Rule edit panel

**Description**
  A description of this routing rule.

**Apply rule to**
  The scope of the rule: either “1” for an individual system, “2” for a group, or “3” for all systems.

If you select **1 System** or **2 Group**, you can use the Prompt function key (F4) to choose from a list of all systems or groups.

**Routing Plan**
The routing plan specifies a group of routing rules that are intended to operate at the same time. To activate a routing plan, use the Set Commands dialog or SET PLAN host command and then refresh the routing rules.

A routing rule that is not assigned to any routing plan is always in effect. Where there is a conflict between an unassigned routing rule and a routing rule that is explicitly assigned to the routing plan that is currently active, the rule that is assigned to the current routing plan takes precedence. In other words, you can define unconditional rules to specify generic routing behavior and then override that behavior using specific rules that are assigned to the current routing plan.

Use Prompt (F4) to select from a list of defined routing plans.

**Original Datastore**
A string that appears in the IRM_IMSDestID field on incoming messages.

This does not have to be the name of a datastore defined in the IMS Connect configuration member. It is simply used as a string to identify matching messages to which this rule applies.

**Additional qualifier**
Specifies an optional condition that qualifies this rule. Use the Prompt function key (F4) to select a qualifier type. You can specify a qualifying rule without a matching master rule in the repository. If a master rule does not already exist for that DestID, an implied routing rule will be generated internally at run time. If NONE is specified, this routing rule will apply to all transactions that have this original datastore ID (DestID). In other words, this will be the master routing rule for this DestID.

**List name**
Use the Prompt function key (F4) to select a qualifying list for this qualifier type. For example, if **Additional qualifier** is Transaction, select from the list of defined Transaction List names. The values in this list are used to determine whether this rule applies to a given OTMA request.

**Message types**
For each message type you can specify one of these values:

**Y** Enable rules-based routing. If you enter Y, you must specify a Target routing list. The Fallback routing list is optional. In each field you can type the name of a routing list or press the Prompt function key (F4) to select from a list of routing list members.

Requests that match this routing rule are routed to a datastore in the target routing list, or, if none is available, to a datastore in the fallback routing list.

**blank** Inherit settings from rules applying to the same condition (IRM_IMSDestID) at either the All systems or Group level. A Group level rule will inherit settings from the All systems level. A system level rule will inherit settings from the Group level, if one is defined, otherwise from the All systems level. The Target and Fallback fields are ignored.

**N** Disable rules-based routing. In other words, this rule will not apply to this message type at the level specified in the **Apply rule to** field. The Target and Fallback fields are ignored.

When you have finished, press the Exit function key (F3). The new routing rule appears in the OTMA Routing Rules panel.

Before OTMA routing rules can take effect, you must select the **Activate OTMA rules** option in the system definition.

## Defining OTMA routing lists

Routing lists enable you to define collections of datastores to use as candidates in an OTMA routing rule.

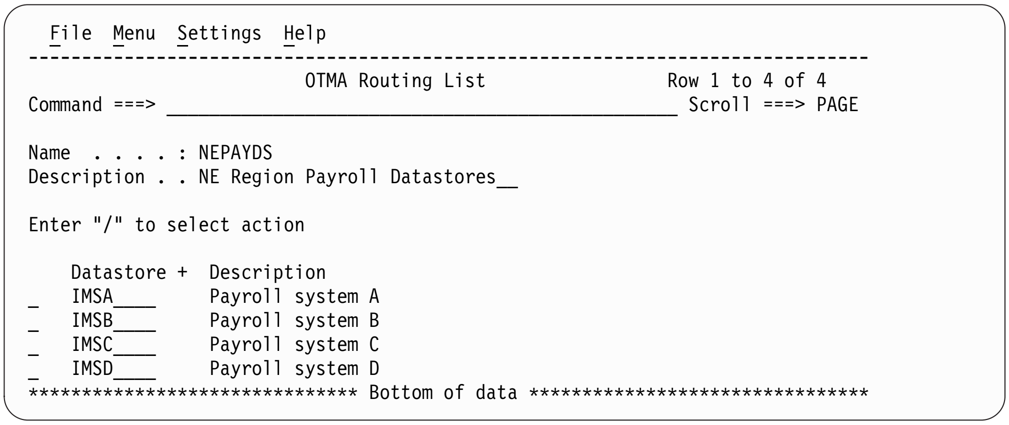
From the primary menu, select option 1.10 **OTMA Routing Lists**. The OTMA Routing Lists panel appears with a list of all existing items (if there are any). You can use the **Filter** field to list only items that match a filter pattern.

Enter the command:

```
NEW rlist
```

where *rlist* is the 1-8 character name of your routing list.

The OTMA Routing List edit panel appears. In this dialog you will assign one or more datastores to the routing list.



```
  File  Menu  Settings  Help
-------------------------------------------------------------------------------
                          OTMA Routing List                 Row 1 to 4 of 4
Command ===> ________________________________________________ Scroll ===> PAGE

Name  . . . . : NEPAYDS
Description . . NE Region Payroll Datastores__

Enter "/" to select action

    Datastore +  Description
_   IMSA____     Payroll system A
_   IMSB____     Payroll system B
_   IMSC____     Payroll system C
_   IMSD____     Payroll system D
******************************* Bottom of data ********************************
```

*Figure 87. OTMA Routing List edit panel*

**Description**
A description of this routing list.

**Option Field**
You can issue an option against one datastore, for example R=Repeat, D=Delete, or I=Insert.

Enter / (slash) to display a pop-up window of all the options.

**Datastore**
Must be a valid datastore name. Use the Prompt function key (F4) to get a list of defined datastores.

When you have finished, press the Exit function key (F3). The new routing list appears in the OTMA Routing Lists panel.

## Defining ODBM targets

ODBM targets define ODBMs and their associated aliases in an IMSplex that IMS Connect is connected to. Once defined to IMS Connect Extensions they become the building blocks for ODBM Routing Lists.

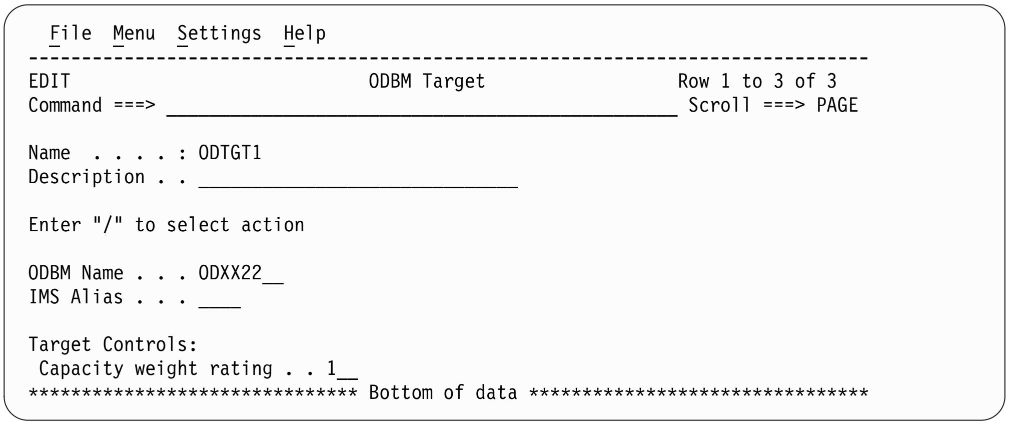
From the primary menu, select option 1.11 **ODBM Targets**. The ODBM Targets panel appears with a list of all existing items (if there are any). You can use the **Filter** field to list only items that match a filter pattern.

Enter the command:

```
NEW target
```

where *target* is the 1-8 character name of your ODBM target.

The ODBM Target edit panel appears.



```
  File  Menu  Settings  Help
--------------------------------------------------------------------------------
EDIT                            ODBM Target                  Row 1 to 3 of 3
Command ===> ________________________________________________ Scroll ===> PAGE

Name  . . . . : ODTGT1
Description . . ______________________________

Enter "/" to select action

ODBM Name . . . ODXX22__
IMS Alias . . . ____

Target Controls:
 Capacity weight rating . . 1__
******************************* Bottom of data ********************************
```

*Figure 88. ODBM Target edit panel*

**Description**
A description of this ODBM target.

**ODBM Name**
The name of the Open Database Manager.

**IMS Alias**
The name of the IMS alias associated with the ODBM.

When the routing routine returns a target candidate and the alias name is blank, CEXROUT0 will attempt to select a target using the alias names associated with the target ODBM name. In this case, workload balancing is not used and all alias members are considered equal in weight for the request. This results in a form of round-robin processing.

**Capacity weight rating**
This field allows you to define the capacity weighting for this ODBM target. The rating is used by workload balancing to reroute incoming messages to alternate ODBM target.

Acceptable field values are numbers in the range 1 - 100 and 0. A value of zero has a special meaning. It indicates that the target is not a candidate for routing.

The higher the rating number, the more workload this ODBM target can handle. Weightings are relative to other ODBM target definitions.

Examples:

1. If you have 3 ODBM targets of equal capacity you could define them all with the same rating number (for example, 1 or 50 or 99). IMS Connect Extensions would apportion incoming requests equally amongst all 3 ODBM targets.
2. If you have 2 ODBM targets and one has twice the capacity as the other, you could define them with weightings of 1 and 2, or 10 and 20, or 40 and 80 and IMS Connect Extensions will apportion twice as much work to the ODBM target with the higher weighting.

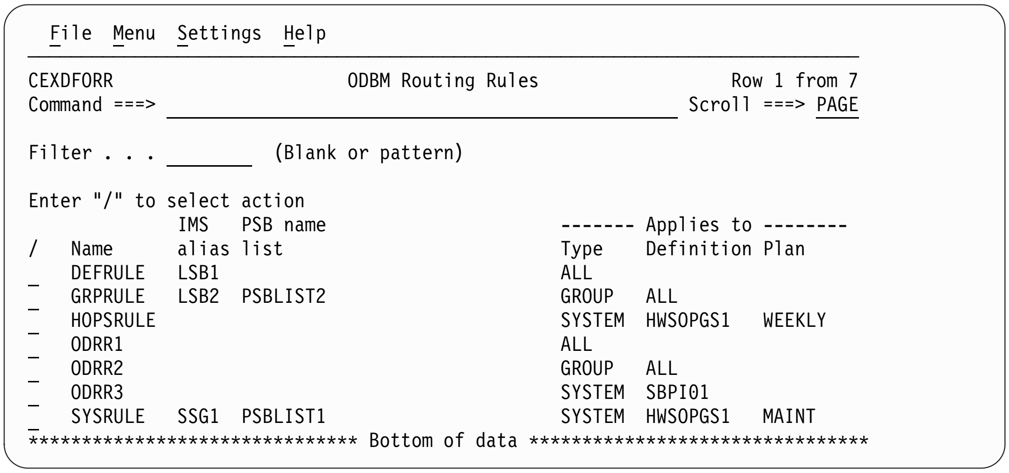
When you have finished, press the Exit function key (F3). The new ODBM target appears in the ODBM Targets panel.

## Defining ODBM routing rules

ODBM routing rules enable you to specify routing behavior for DRDA requests. They convert an input alias into a target selected from a collection of ODBM targets that you specify.

You can specify different target and fallback routing lists. A routing rule can be assigned along with other related rules to a routing plan.

From the primary menu, select option 1.12 **ODBM Routing Rules**. The ODBM Routing Rules panel appears with a list of all existing rules (if there are any). You can use the **Filter** field to list only items that match a filter pattern.



```
  File  Menu  Settings  Help
──────────────────────────────────────────────────────────────────────────────
CEXDFORR                      ODBM Routing Rules                  Row 1 from 7
Command ===> ________________________________________________ Scroll ===> PAGE

Filter . . . ________  (Blank or pattern)

Enter "/" to select action
              IMS   PSB name                        ------- Applies to --------
/   Name      alias list                            Type    Definition Plan
_   DEFRULE   LSB1                                  ALL
_   GRPRULE   LSB2  PSBLIST2                        GROUP   ALL
_   HOPSRULE                                        SYSTEM  HWSOPGS1   WEEKLY
_   ODRR1                                           ALL
_   ODRR2                                           GROUP   ALL
_   ODRR3                                           SYSTEM  SBPI01
_   SYSRULE   SSG1  PSBLIST1                        SYSTEM  HWSOPGS1   MAINT
******************************* Bottom of data ********************************
```

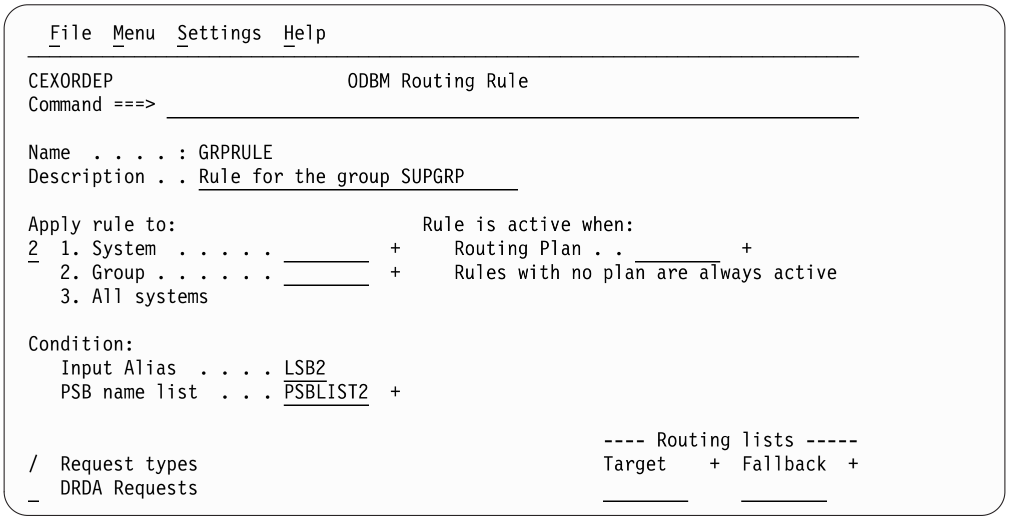
*Figure 89. ODBM Routing Rules panel*

To add a new routing rule, enter the command:

```
NEW rrule
```

where *rrule* is the 1-8 character name of your routing rule.

The ODBM Routing Rule edit panel appears. In this dialog you will define the routing behavior for requests with a specified input alias.



```
  File  Menu  Settings  Help
-----------------------------------------------------------------------------
CEXORDEP                     ODBM Routing Rule
Command ===> ________________________________________________________________

Name  . . . . : GRPRULE
Description . . Rule for the group SUPGRP

Apply rule to:                       Rule is active when:
2  1. System  . . . . . ________  +     Routing Plan . . ________  +
   2. Group . . . . . . ________  +     Rules with no plan are always active
   3. All systems

Condition:
   Input Alias  . . . . LSB2
   PSB name list  . . . PSBLIST2  +

                                                   ---- Routing lists -----
/  Request types                                   Target    +  Fallback  +
_  DRDA Requests                                   ________     ________
```

*Figure 90. ODBM Routing Rule edit panel*

**Description**
A description of this routing rule.

**Apply rule to**
The scope of the rule: either “1” for an individual system, “2” for a group, or “3” for all systems.

If you select **1 System** or **2 Group**, you can use the Prompt key (F4) to choose from a list of all systems or groups.

**Routing Plan**
The routing plan specifies a group of routing rules that are intended to operate at the same time. To activate a routing plan, use the Set Commands dialog or SET PLAN host command and then refresh the routing rules.

A routing rule that is not assigned to any routing plan is always in effect. Where there is a conflict between an unassigned routing rule and a routing rule that is explicitly assigned to the routing plan that is currently active, the rule that is assigned to the current routing plan takes precedence. In other words, you can define unconditional rules to specify generic routing behavior and then override that behavior using specific rules that are assigned to the current routing plan.

Use Prompt (F4) to select from a list of defined routing plans.

**Condition**
The Input Alias specifies the alias name provided by the client in the DRDA request.

When a PSB name list is specified, a routing rule is built for the alias name and each member of the PSB name list. If the PSB name list is blank, the routing rule applies to all PSB names.

The values for both the alias and PSB name relate to data in objects included in the DRDA request “ACCRDB-Access RDB” (code point 2001).

**Request types**
DRDA Requests is the only supported request type for ODBM routing. You can specify one of these values:

**Y** Enable rules-based routing. If you enter Y, you must specify a Target routing list. The Fallback routing list is optional. In each field you can type the name of a routing list or press the Prompt function key (F4) to select from a list of routing list members.

Requests that match this routing rule are routed to an ODBM target in the target routing list, or, if none is available, to a target in the fallback routing list.

**blank** Inherit settings from rules applying to the same condition at either the All systems or Group level. A Group level rule will inherit settings from the All systems level. A system level rule will inherit settings from the Group level, if one is defined, otherwise from the All systems level. The Target and Fallback fields are ignored.

**N** Disable rules-based routing. In other words, this rule will not apply to this request type at the level specified in the **Apply rule to** field. The Target and Fallback fields are ignored.

When you have finished, press the Exit function key (F3). The new routing rule appears in the ODBM Routing Rules panel.

Before ODBM routing rules can take effect, you must select the **Activate ODBM rules** option in the system definition.

## Defining ODBM routing lists

Routing lists enable you to define collections of ODBM targets to use as candidates in an ODBM routing rule.

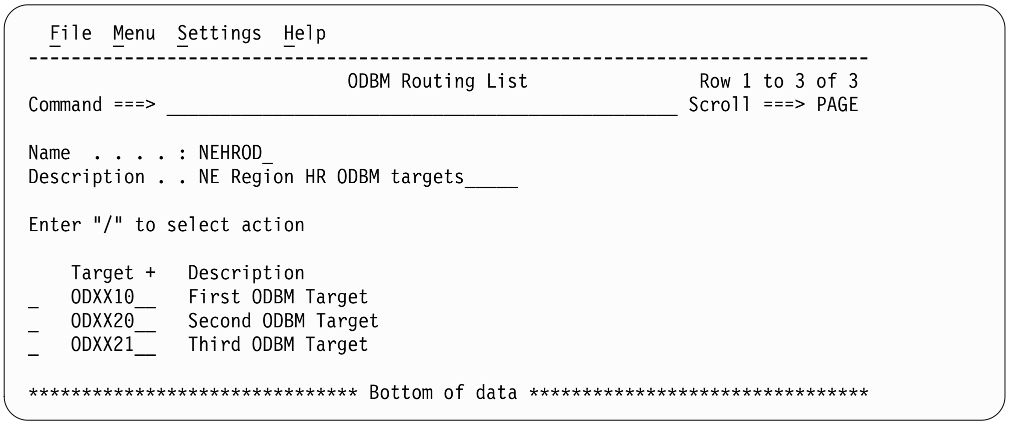
From the primary menu, select option 1.13 **ODBM Routing Lists**. The ODBM Routing Lists panel appears with a list of all existing items (if there are any). You can use the **Filter** field to list only items that match a filter pattern.

Enter the command:

```
NEW rlist
```

where *rlist* is the 1-8 character name of your routing list.

The ODBM Routing List edit panel appears. In this dialog you will assign one or more datastores to the routing list.



```
  File  Menu  Settings  Help
-------------------------------------------------------------------------------
                             ODBM Routing List                Row 1 to 3 of 3
Command ===> ________________________________________________ Scroll ===> PAGE

Name  . . . . : NEHROD_
Description . . NE Region HR ODBM targets_____

Enter "/" to select action

    Target +   Description
_   ODXX10__   First ODBM Target
_   ODXX20__   Second ODBM Target
_   ODXX21__   Third ODBM Target

******************************* Bottom of data ********************************
```

*Figure 91. ODBM Routing List edit panel*

**Description**
A description of this routing list.

**Option Field**
You can issue an option against one ODBM target, for example R=Repeat, D=Delete, or I=Insert.

Enter / (slash) to display a pop-up window of all the options.

**Target**
Must be a valid ODBM target. Use the Prompt function key (F4) to get a list of defined targets.

When you have finished, press the Exit function key (F3). The new routing list appears in the ODBM Routing Lists panel.

## Defining PSB name lists

PSB name lists enable you to define collections of PSB names to which an ODBM routing rule can be applied.

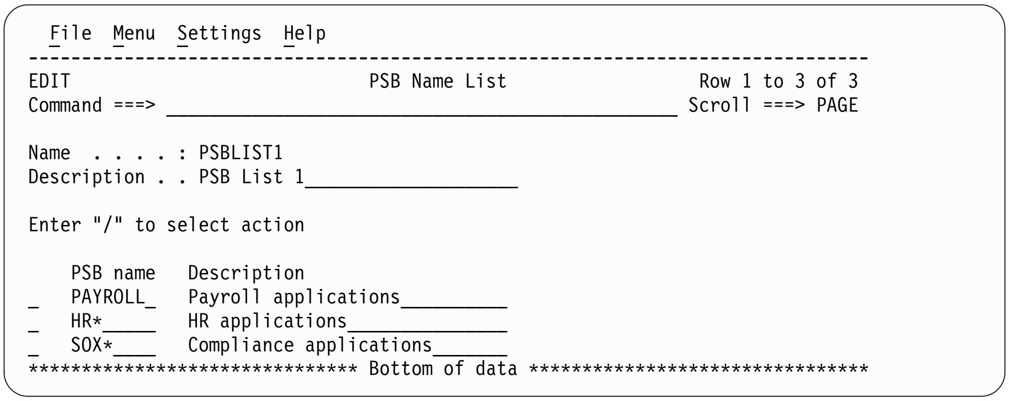
From the primary menu, select option 1.14 **PSB Name Lists**. The PSB Name List panel appears with a list of all existing items (if there are any). You can use the **Filter** field to list only items that match a filter pattern.

Enter the command:

```
NEW psblist
```

where *psblist* is the 1-8 character name of your PSB name list.

The PSB Name List edit panel appears. In this dialog you will assign one or more PSB names to the PSB name list.



```
   File  Menu  Settings  Help
 --------------------------------------------------------------------------------
 EDIT                            PSB Name List                     Row 1 to 3 of 3
 Command ===> ________________________________________________ Scroll ===> PAGE

 Name  . . . . : PSBLIST1
 Description . . PSB List 1___________________

 Enter "/" to select action

     PSB name   Description
 _   PAYROLL_   Payroll applications_________
 _   HR*_____   HR applications______________
 _   SOX*____   Compliance applications______
 ******************************* Bottom of data *********************************
```

*Figure 92. PSB Name List edit panel*

**Description**
A description of this PSB name list.

**Option Field**
You can issue an option against one PSB name, for example R=Repeat, D=Delete, or I=Insert.

Enter / (slash) to display a pop-up window of all the options.

**PSB name**
Can be an actual PSB name or a generic name that includes a single trailing wildcard ("*"). Embedded wildcards are not allowed in generic PSB names.

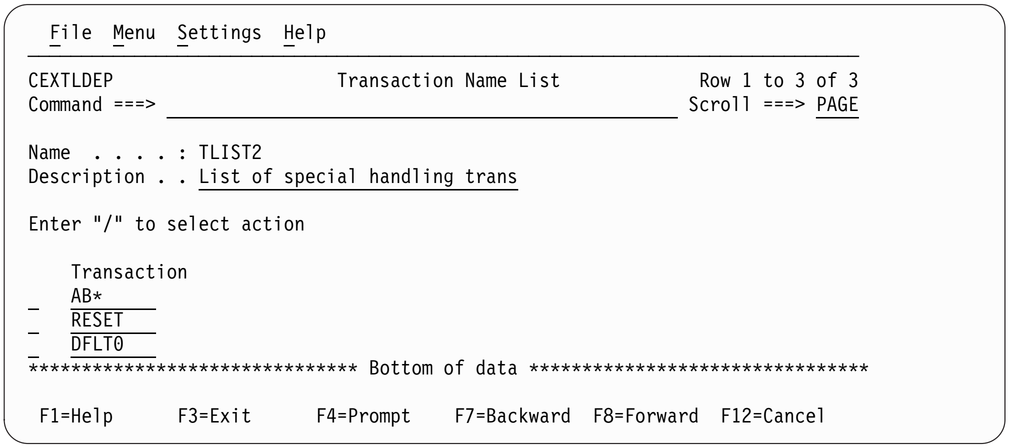
When an ODBM routing rule uses a PSB List, a rule is generated for each PSB name in the list. IMS Connect Extensions will try to match the full PSB name on the incoming DRDA request with the rule. If it is unsuccessful, the granularity of the match is reduced one character at a time until a match is found. If no match is found then the incoming request cannot be routed.

**Description**
You can provide an optional description of the PSB name being defined.

When you have finished, press the Exit function key (F3). The new PSB name list appears in the PSB Name Lists panel.

## Defining transaction lists

A Transaction List definition is a collection of transaction codes. You can use a transaction list in an OTMA routing rule to route selected transactions to a different routing list from the one specified in the master routing rule.



```
  File  Menu  Settings  Help
 ──────────────────────────────────────────────────────────────────────────────
CEXTLDEP                    Transaction Name List             Row 1 to 3 of 3
Command ===> ________________________________________________ Scroll ===> PAGE

Name  . . . . : TLIST2
Description . . List of special handling trans

Enter "/" to select action

    Transaction
_   AB*
_   RESET
_   DFLT0
****************************** Bottom of data ********************************

 F1=Help      F3=Exit      F4=Prompt    F7=Backward  F8=Forward  F12=Cancel
```

*Figure 93. Transaction List detail panel*

Use the Transaction Name List panel to enter the transaction codes for this transaction list.

**Description**
A description of this transaction list.

**Option Field**
You can issue an option against one transaction code, for example R=Repeat, D=Delete, or I=Insert.

Enter / (slash) to display a pop-up window of all the options.

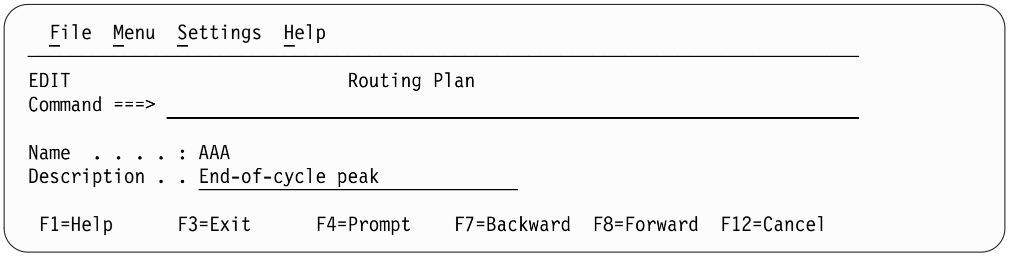
**Transaction**
Enter one transaction code per line.

Transaction codes can be fully qualified transaction names or generic transaction names. A generic transaction name consists of one or more characters followed by a trailing asterisk (*). For example, AB* would match any transaction code starting with AB, such as AB, ABC, and ABC12. Embedded asterisks are not allowed in generic transaction names.

When you have finished, press the Exit function key (F3). The new transaction list appears in the Transaction Lists panel.

## Defining routing plans

Routing plans are used in IMS Connect Extensions to group routing rules that are intended to operate at the same time.



```
  File  Menu  Settings  Help
────────────────────────────────────────────────────────────────────
EDIT                           Routing Plan
Command ===> _______________________________________________________

Name  . . . . : AAA
Description . . End-of-cycle peak____________

 F1=Help      F3=Exit      F4=Prompt    F7=Backward  F8=Forward  F12=Cancel
```

*Figure 94. Routing Plan detail panel*

Use the Routing Plan panel to enter a plan name and description. Routing rules can be assigned to a routing plan so that they can subsequently be activated as a set. The association between a routing rule and a routing plan is made in the OTMA and ODBM routing rule definition.

**Description**
A description of this routing plan.

When you have finished, press the Exit function key (F3). The new routing plan appears in the Routing Plans panel.

**Related reference**:

"Defining OTMA routing rules" on page 219
OTMA routing rules enable you to specify routing behavior for IRM messages. They describe which routing lists to use to route messages that have a given original datastore identifier (DestID).

"Defining ODBM routing rules" on page 224
ODBM routing rules enable you to specify routing behavior for DRDA requests. They convert an input alias into a target selected from a collection of ODBM targets that you specify.

# Chapter 20. Changing settings dynamically

The Commands menu lets you navigate to all IMS Connect Extensions commands within the ISPF dialog. These options allow you to dynamically change values for definitions, and manage user exits, journals, datastores, and traces.

**Tip:** Many of these commands can also be issued from a REXX exec or the Operations Console.

To open the Commands menu, enter the **CX** line action next to a system in the Status Monitor or in the Operations - Systems View dialog.

**Exits** Reload, add or delete, disable and enable user exits for the next message requesting processing by this user exit without interruption of IMS Connect execution.

**Refresh**
Rebuild IMS Connect Extensions definitions. IMS Connect Extensions reads the definitions from the repository and rebuilds the in-memory values.

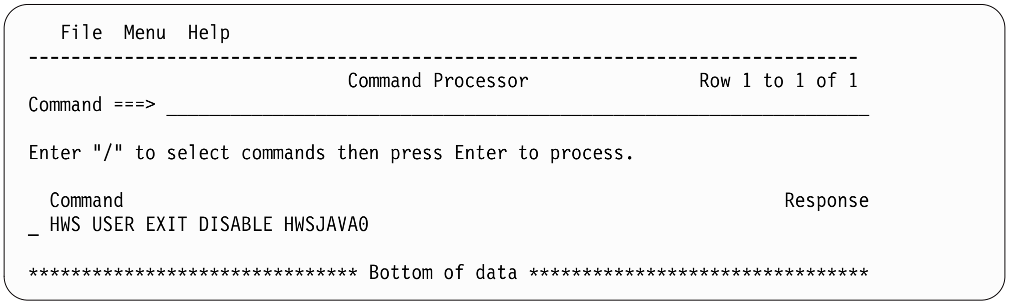
**Security**
Refresh SAF Class rules or delete any cached User ID security profiles.

**Set** Temporarily override IMS Connect Extensions definition settings. The setting values are based on the input parameters (and not the repository).

**Journal**
Switch the Active Journal data set or refresh some Active and Archive Journal options.

**Datastores**
Open the Datastore Monitor. The Datastore Monitor provides more detailed information about datastores and enables selected datastores to be started, stopped, drained, or resumed. See "Datastore Monitor" on page 104.

After one of these options has been selected and the command values entered, the command processor panel is displayed. This allows you to confirm the command request before it is processed by IMS Connect Extensions.



```
   File  Menu  Help
------------------------------------------------------------------------------
                              Command Processor                Row 1 to 1 of 1
Command ===> __________________________________________________________________

Enter "/" to select commands then press Enter to process.

  Command                                                               Response
_ HWS USER EXIT DISABLE HWSJAVA0

****************************** Bottom of data ********************************
```

*Figure 95. Example: User Exits command processor pre-processing*

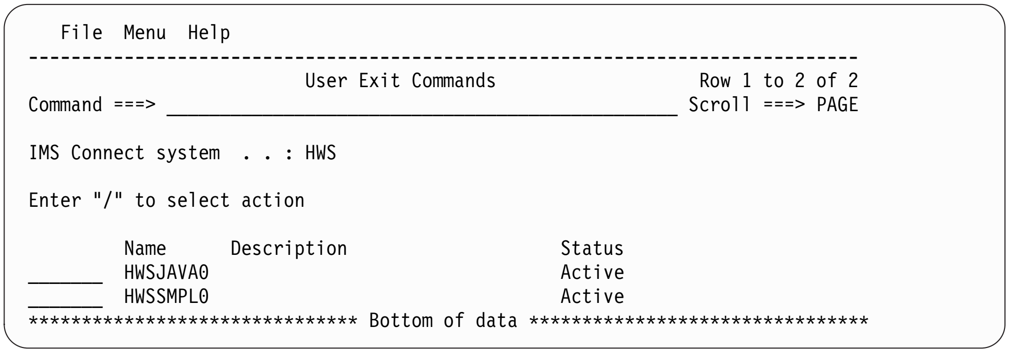
Type / to select a command and press Enter to proceed.

IMS Connect Extensions generates the commands and displays the return code and any messages on the command processor panel.

# Changing user exits

The USER EXIT command enables a selected user exit to be reloaded, added, deleted, or made active or inactive for the next message requesting processing by this user exit. This is performed without interrupting the execution of IMS Connect.

The User Exit Commands panel lists exits that are already loaded. The Status field displays the current status of the user exit. `Active` indicates that the user exit is processing incoming messages.



```
   File  Menu  Help
------------------------------------------------------------------------------
                         User Exit Commands                  Row 1 to 2 of 2
Command ===> ________________________________________________ Scroll ===> PAGE

IMS Connect system  . . : HWS

Enter "/" to select action

         Name      Description                   Status
_______  HWSJAVA0                                Active
_______  HWSSMPL0                                Active
******************************* Bottom of data ********************************
```

*Figure 96. User Exit Commands panel*

## Primary command

**ADD** The **ADD** command loads a new user exit executable. The user exit must exist in one of the STEPLIB data sets associated with the IMS Connect system and be defined to IMS Connect Extensions. If the user exit is already known to IMS Connect or the exit is not found in the STEPLIB data set concatenation, the ADD request is rejected.

The new user exit executable is initialized by IMS Connect Extensions and after successful initialization is ready to process incoming messages for the MSG IDs specified.

**Notes:**

1. If the new user exit uses Message IDs that are used by existing user exits, the ADD request is rejected.
2. IMS Connect Extensions does not automatically update the IMS Connect configuration member with the name of the user exit. You must do this manually.
3. Once a user exit has been added, it counts against the total number of message exits allowed by IMS Connect. This is true even if the user exit is later deleted or disabled.

## Line actions

You can enter the following line actions next to a user exit.

**/** Display a pop-up window of available line actions.

**RELOAD**
The RELOAD option reloads a new copy of the user exit executable. If a RELOAD operation is already in progress for the user exit, the request is rejected.

All requests currently using the existing copy of the user exit are allowed to complete that user exit phase (READ, XMIT, EXER) before the current copy is removed.

Messages that are processed for READ with the current user exit may be processed by the new user exit for XMIT or EXER. Any user exit which is Reloaded must be able to support XMIT where the READ was processed by a different version of the user exit.

**Tips:**

1. If possible, RELOAD user exits during a period of low activity.
2. The new version of the user exit must replace the current user exit in the IMS Connect STEPLIB concatenation or be defined higher in the STEPLIB concatenation.
3. The new copy of the user exit executable is initialized by IMS Connect Extensions. If the user exit fails to initialize, the RELOAD request is rejected.
4. The RELOAD process may take some time to complete, as all messages being processed by the current version of the message exit must complete before the RELOAD process is considered complete.
5. The **RELOAD** command cannot be used to change or alter the Message ID strings supported by the user exit. If you need to change the Message ID string for a user exit, use the **ADD** command to add a new user exit supporting the Message IDs.
6. During a RELOAD, any XIB storage being used by the user exit must be capable of being shared simultaneously by both the current and the new user exit.
7. After the **RELOAD** command is issued for a given user exit, the exit must process at least one input message before the **RELOAD** command can be used again for the same user exit.

**DELETE**

The DELETE option deletes the association between the MSG IDs and the user exit. When a user exit has been deleted, all messages for those MSG IDs will be rejected by IMS Connect Extensions.

Note that the user exit is not physically deleted from the IMS Connect system. The user exit executable is not physically deleted from the load library. If required, the **ADD** command can be used to reinstate a deleted user exit.

**DISABLE**

The DISABLE option suspends processing for the user exit. The exit is not physically removed from the IMS Connect system.

IMS Connect Extensions will reject incoming messages for those MSG IDs supported by the user exit. Messages for XMIT or EXER are supported.

**ENABLE**

The ENABLE option re-enables a previously disabled user exit and re-associates MSG IDs with the user exit.

If the user exit is not in a disabled state, the command is rejected.

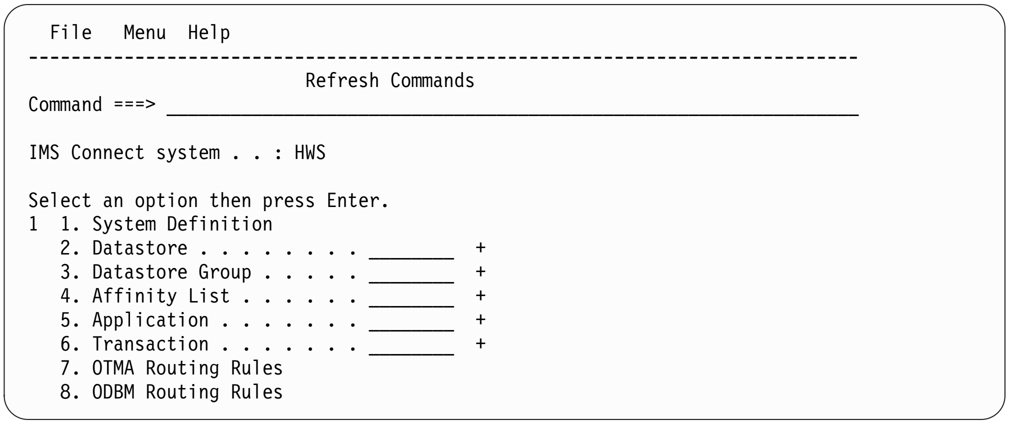
## User exit considerations

The following considerations and restrictions apply to all user exit commands.

1. If the remote client is using IMS conversational processing, the user exits and the remote client have to be able to tolerate an environment in which some iterations of the conversation are processed by one version of the user exit and subsequent iterations by a different version.
2. If the remote client is using persistent sessions, the user exits and the remote client have to be able to tolerate an environment where some messages for the session are processed by one version of the user exit and subsequent messages for the session are processed by a different version.
3. If a user exit has been Disabled or Deleted and IMS Connect Extensions Advanced features is turned off, the user exits may begin to receive messages.

## Changing definitions dynamically with REFRESH commands

The **REFRESH** command causes IMS Connect Extensions to read the definitions from the repository and rebuild the in-memory values.



```
  File   Menu  Help
------------------------------------------------------------------------------
                          Refresh Commands
Command ===> _________________________________________________________________

IMS Connect system . . : HWS

Select an option then press Enter.
1  1. System Definition
   2. Datastore . . . . . . . . ________  +
   3. Datastore Group . . . . . ________  +
   4. Affinity List . . . . . . ________  +
   5. Application . . . . . . . ________  +
   6. Transaction . . . . . . . ________  +
   7. OTMA Routing Rules
   8. ODBM Routing Rules
```

*Figure 97. Refresh Commands panel*

To refresh a datastore, datastore group, affinity list, application, or transaction, use the Prompt function key (F4) to select the definition from a list. These definitions can only be refreshed for the system on which the command will run.

The Refresh Commands panel recognizes the following options.

### System Definition

The System Definition option reloads values from the repository for the following System Definition fields:

- Message recall count
- Mixed case password option
- Collection level
- Pacing option
- Interval count, warning threshold and reject threshold
- Security option
- ACEE Cache option and ageing interval
- IMS Connect validation option and validation type
- Security class
- OTMA rules-based routing option

- ODBM routing option
- Transaction routing option
- Defined applications
- Workload balancing option
- Session message limit option and threshold
- Statistics collection option

All these values will be reloaded from the System Definition regardless of whether they have been changed. Any changes previously made by individual **SET** commands will be overridden.

**Note:**

1. Options for Pacing, Security, Transaction Routing, Workload Balancing and Statistics Collection will **not** be activated if the IMS Connect system was started with Advanced Features inactive.
2. If the list of defined Applications is changed, IMS Connect Extensions will reload all transaction definitions associated with the new list of Applications.
3. The Security class is reloaded whenever the class name is changed (and the Security and IMS Connect validation options are active) or whenever the Security option or IMS Connect validation options are activated.

The following System Definition values are **not** reloaded when the System Definition is refreshed:

- Port number
- Host name
- IMS Connect Extensions option
- Event Collection option
- All Active Journal definitions and options
- All Archive Journal definitions and options
- Log Record Number
- Advanced Features option
- Publisher API option

These values can only be changed by **restarting** the IMS Connect system.

## Datastore

The Datastore option reloads values from the repository for all datastore fields:

- Datastore Group name
- Pacing option
- Warning threshold and reject threshold
- Routing options (including Send Only and Resume TPIPE routing rules)
- Capacity weight rating
- Primary datastore

All these values will be reloaded from the datastore definition regardless of whether they have been changed. Any changes previously made by individual SET commands will be overridden.

Datastores are only available to be refreshed if they are defined both to IMS Connect Extensions and in the IMS Connect configuration member.

If the Datastore Group name or Transaction Routing options or Capacity weight rating are changed, IMS Connect Extensions will rebuild the Candidate List, updating the Transaction Routing definitions.

You cannot add a new datastore using the Datastore option. New datastores can be added by changing the IMS Connect configuration member, adding datastore definitions to IMS Connect Extensions and then restarting the IMS Connect system.

## Datastore Group

The Datastore Group option reloads values from the repository for all Datastore Group fields:

- Pacing option
- Warning threshold and reject threshold

All of these values will be reloaded from the Datastore Group definition regardless of whether they have been changed. Any changes previously made by individual **SET** commands will be overridden.

Groups are only available to be refreshed if they contain a datastore that is defined in the IMS Connect configuration member.

You cannot add a new Datastore Group using the Datastore Group option. New Datastore Groups will be added if you REFRESH a datastore that references a Datastore Group not previously defined when IMS Connect Extensions was started.

## Affinity List

The Affinity List option reloads the specified Affinity List. The list of datastores defined in the Affinity List will be rebuilt in memory, replacing the previous definition.

If any Datastore Affinity or Transaction Affinity refer to the Affinity List, IMS Connect Extensions will rebuild the Candidate List, updating the Transaction Routing definitions.

You cannot add a new Affinity List using the Affinity List option. New Affinity Lists will be added if you REFRESH a datastore, transaction or application definition that references an Affinity List not previously defined when IMS Connect Extensions was started.

Affinity lists are only available to be refreshed if they contain a datastore that is defined in the IMS Connect configuration member.

## Application

The Application option reloads values from the repository for all Application fields:

- Transaction Routing option
- Route transaction selection
- Routing error option

All of these values will be reloaded from the Application definition regardless of whether they have been changed. Any changes previously made by individual **SET** commands will be overridden.

Applications are only available to be refreshed if they are included in the IMS Connect Extensions system definition.

Refreshing an Application will automatically update all the transactions that are part of the Application and IMS Connect Extensions will rebuild the Candidate List, updating the Transaction Routing definitions.

You cannot add a new Application using the Application option. Instead, you add new applications by first adding them to the system definition, and then refreshing that definition.

## Transaction

The Transaction option reloads values from the repository for all Transaction fields:

- Application Name
- Transaction Timer Option (including timeout values)
- Transaction Expiration Option (including **Set F1_TRNEXP** setting)
- Client ID Cancellation Option (including **Set F3_CANCID** setting)
- Transaction Routing option
- Route transaction selection
- Routing error option

All of these values will be reloaded from the Transaction definition regardless of whether they have been changed. Any changes previously made by individual **SET** commands will be overridden.

If the Application name or Transaction Routing options are changed, IMS Connect Extensions will rebuild the Candidate List, updating the Transaction Routing definitions.

You can add a new transaction using the Transaction option.

## OTMA Routing Rules

The OTMA Routing Rules option rebuilds all of the system internal routing rules for OTMA messages based on the current routing rule and routing list definitions in the repository. The routing rules are written to the IMS Connect Extensions log (CEXPRINT) at system startup (see Figure 59 on page 143) and after OTMA routing rules are refreshed.

If you change the active OTMA routing plan you must then refresh the OTMA routing rules to load the new set of rules. If you have changed another definition that would affect rules-based routing behavior for OTMA messages, you must refresh that definition first. This is so that when the routing rules are refreshed, they operate on the updated definition.

For example, if you have defined a set of routing lists and routing rules and then activated OTMA rules-based routing, you must first refresh the system definition and then refresh the routing rules.

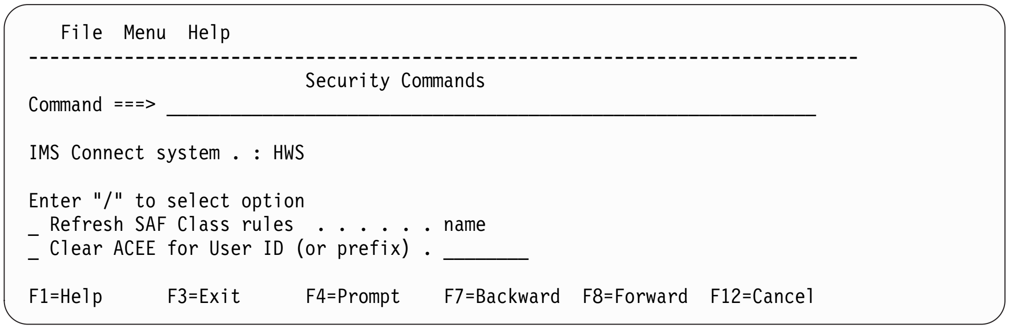
### ODBM Routing Rules

The ODBM Routing Rules option rebuilds all of the system internal routing rules for DRDA requests based on the current routing rule, routing list, and ODBM target and PSB name list definitions in the repository. The routing rules are written to the IMS Connect Extensions log (CEXPRINT) at system startup and after ODBM routing rules are refreshed.

If you change the active ODBM routing plan you must then refresh the ODBM routing rules to load the new set of rules.

## Refreshing SAF Class rules and clearing the ACEE cache

The **SECURITY** command is used to Refresh SAF Class rules and to delete any cached User ID security profiles.



```
   File  Menu  Help
------------------------------------------------------------------------------
                           Security Commands
Command ===> ______________________________________________________________

IMS Connect system . : HWS

Enter "/" to select option
_ Refresh SAF Class rules  . . . . . . name
_ Clear ACEE for User ID (or prefix) . ________

F1=Help      F3=Exit      F4=Prompt    F7=Backward  F8=Forward  F12=Cancel
```

*Figure 98. Security Commands panel*

The Security Commands panel recognizes the following options:

**Refresh SAF Class rules**
If IMS Connect Extensions is performing IMS Connect validation, it preloads the security Resource Class in storage for performance reasons. When the security Resource Rules change, they must be refreshed and reloaded into storage.

This option allows the SAF Class rules to be reloaded immediately.

**Clear ACEE for User ID**
In order to minimize security processing and overhead, IMS Connect Extensions caches ACEE structures.

This option allows the User ID ACEEs to be deleted immediately rather than wait for Ageing Interval processing.

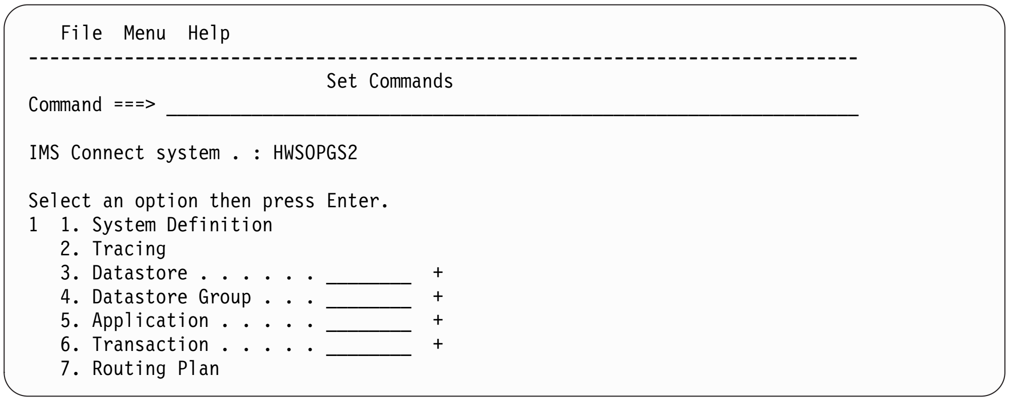
You can clear ACEE caches for a specific user or all users with a matching prefix.

**Prefix** Use a prefix and only ACEE structures matching the prefix will be deleted. These examples show how the prefix works:

| | |
|---|---|
| ***** | Delete all ACEE structures |
| **AB*** | Delete all ACEE structures starting with AB |

## Temporarily changing definitions with SET commands

The **SET** command is used to temporarily override IMS Connect Extensions in-memory settings. The new values are based on input parameters from the screen (and not the repository).



```
   File  Menu  Help
------------------------------------------------------------------------------
                            Set Commands
Command ===> _________________________________________________________________

IMS Connect system . : HWSOPGS2

Select an option then press Enter.
1  1. System Definition
   2. Tracing
   3. Datastore . . . . . . ________  +
   4. Datastore Group . . . ________  +
   5. Application . . . . . ________  +
   6. Transaction . . . . . ________  +
   7. Routing Plan
```

*Figure 99. Set Commands panel*

To set a datastore, datastore group, application, or transaction, use the Prompt function key (F4) to select the definition. This list only includes definitions that can be SET for the system on which the command will run.

### SET SYSTEM DEFINITION

The SET SYSTEM DEFINITION command causes IMS Connect Extensions definitions for a single System Definition to be modified. IMS Connect Extensions reads the input parameters from the panel and rebuilds the in-memory values.

The panel displays the current values (as opposed to the values in the repository) for the fields you may be able to modify. Change as many values as you wish and press Enter to process the commands.

**Note:**

1. If the IMS Connect system started with event collection inactive then you cannot change the collection level.
2. If the system was started with advanced features inactive, then none of the advanced features can be overridden through the SET SYSTEM DEFINITION command.

### SET TRACING

The SET TRACING command causes IMS Connect Extensions to trace based on filtering criteria.

Tracing records are written to the Active Journal data set and then, if archiving is enabled, archived to the Archive Journals. Tracing continues until it is deactivated or until the system shuts down. You can also set a flag to automatically reactivate tracing when the system restarts.

You can trace activity by port. For OTMA workloads you can also filter activity using one of the following conditional resource types: Client Name, Transaction, Message Exit, User ID, LTERM, or IP Address.

If you want tracing records, ensure the Collection level is set to 1 or more. If Collection level is set to 0 no tracing records will be written.

The panel displays the current values, if any, for the fields.

**Activate Tracing**
Use this field to activate or deactivate the tracing feature for IMS Connect Extensions.

Enter / (slash) to activate tracing. To make tracing persistent, also select **Reactivate tracing after system restart**.

**Note:** You can only make a trace persistent at the same time as you activate it.

**Tracing Level**
This field allows you to activate and set the level of tracing for OTMA workloads.

The possible values are:

**0** No tracing records are written.

**1** Tracing records are written.

**2** Tracing records are written with an additional section containing client application data.

For a list of trace event records, see “IMS Connect trace records” on page 439.

**Port** Enter a decimal value in the range 1 - 65535 to specify which IMS Connect TCP/IP port you want to initiate a trace against. Only activities directed at that port are traced. Use an asterisk (*) for all ports or LOCAL for local port.

This option applies to both OTMA and ODBM workloads.

**Conditional trace resource**
In addition to the port, you can filter traces by one of the following:

- 1. Client Name. Enter up to 8 alphanumeric characters. The first character cannot be numeric.
- 2. Transaction. Enter up to 8 alphanumeric characters. The first character cannot be numeric.
- 3. Message Exit. Enter up to 8 alphanumeric characters. The first character cannot be numeric.
- 4. User ID. The value must specify a string that represents a valid user ID, an asterisk, or a partial user ID ending in an asterisk (*). No other wildcards are permitted.
- 5. LTERM (logical terminal). Enter up to 8 alphanumeric characters. The first character cannot be numeric.
- 6. The IPv4 address, IPv6 address, or host name of the client. If you enter a host name, IMS Connect Extensions will attempt to resolve the IP address.

**Note:** Conditional tracing is not applicable when tracing ODBM activity. To trace ODBM activity, simply specify the port.

Press Enter to generate the commands.

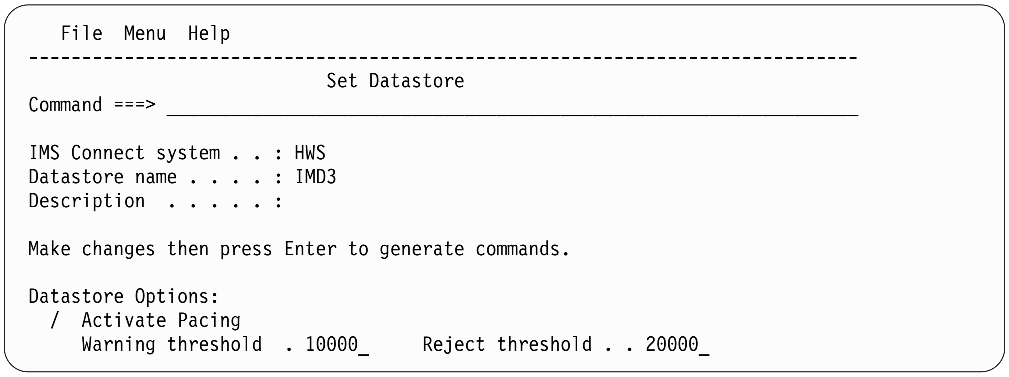
**Related concepts**:

Chapter 4, “Event collection,” on page 55
IMS Connect Extensions continuously collects events as incoming messages or DRDA requests are processed or when other changes occur, such as when IMS Connect starts up and shuts down or when a datastore goes offline.

## SET DATASTORE

The SET DATASTORE command causes IMS Connect Extensions definitions for a single datastore to be modified. IMS Connect Extensions reads the input parameters from the panel and rebuilds the in-memory values.

A datastore is only available to be SET if it is defined both to IMS Connect Extensions and in the IMS Connect configuration member.



```
   File  Menu  Help
------------------------------------------------------------------------------
                            Set Datastore
Command ===> _________________________________________________________________

IMS Connect system . . : HWS
Datastore name . . . . : IMD3
Description  . . . . . :

Make changes then press Enter to generate commands.

Datastore Options:
  /  Activate Pacing
     Warning threshold  . 10000_     Reject threshold . . 20000_
```

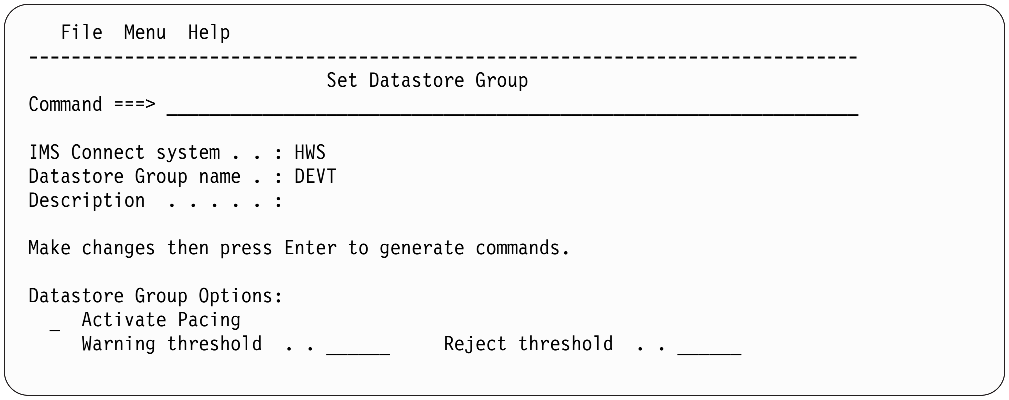
Figure 100. Set Datastore command panel

The panel displays the current values (not the values from the Datastore definition in the repository). Change as many values as you wish and press Enter to process.

## SET DATASTORE GROUP

The SET DATASTORE GROUP command causes IMS Connect Extensions definitions for a single Datastore Group to be modified. IMS Connect Extensions reads the input parameters from the panel and rebuilds the in-memory values.

Datastore groups are only available to be SET if they contain at least one datastore that is defined both to IMS Connect Extensions and in the IMS Connect configuration member.



```
   File  Menu  Help
------------------------------------------------------------------------------
                            Set Datastore Group
Command ===> _________________________________________________________________

IMS Connect system . . : HWS
Datastore Group name . : DEVT
Description  . . . . . :

Make changes then press Enter to generate commands.

Datastore Group Options:
  _  Activate Pacing
     Warning threshold  . . ______     Reject threshold  . . ______
```

Figure 101. Set Datastore Group command panel

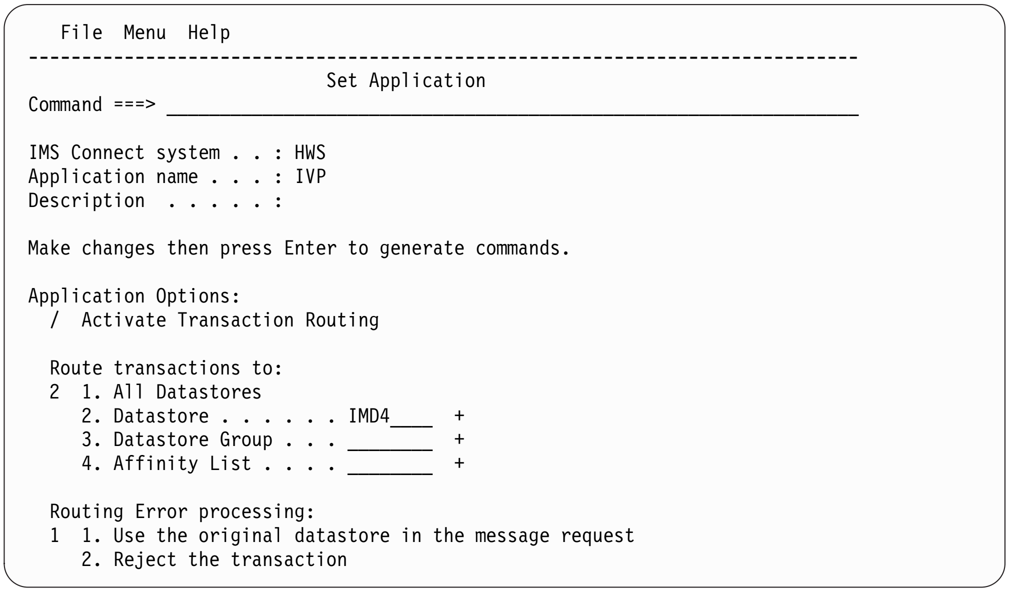
The panel displays the current values (not the values from the Datastore Group definition in the repository). Change as many values as you wish and press Enter to process.

**Related reference**:

"Defining datastore groups" on page 209
Datastore Group Definition allows you to group datastores controlled by IMS Connect Extensions.

## SET APPLICATION

The SET APPLICATION command causes IMS Connect Extensions definitions for a single application to be modified. IMS Connect Extensions reads the input parameters from the panel and rebuilds the in-memory values.



```
   File  Menu  Help
 ------------------------------------------------------------------------------
                            Set Application
 Command ===> _________________________________________________________________

 IMS Connect system . . : HWS
 Application name . . . : IVP
 Description  . . . . . :

 Make changes then press Enter to generate commands.

 Application Options:
   /  Activate Transaction Routing

   Route transactions to:
   2  1. All Datastores
      2. Datastore . . . . . . IMD4____  +
      3. Datastore Group . . . ________  +
      4. Affinity List . . . . ________  +

   Routing Error processing:
   1  1. Use the original datastore in the message request
      2. Reject the transaction
```

*Figure 102. Set Application command panel*

The panel displays the current values, not the values from the Application definition in the repository. Change as many values as you wish and press Enter to process.

**Note:**

1. Applications are only available to be SET if they are included in the IMS Connect Extensions system definition.
2. The SET APPLICATION command will propagate changes to the transactions, which do not have the override option active, which belong to the Application.
3. You can only change the transaction routing option to a datastore, Datastore Group or Affinity List that is already defined in the Candidate List for this IMS Connect system.

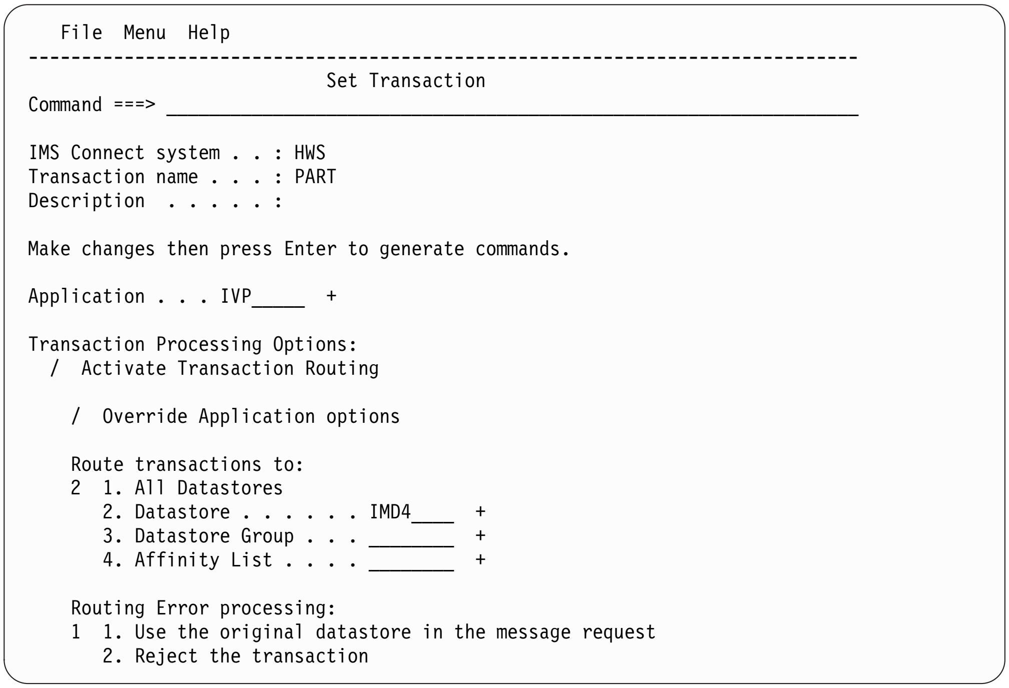
**Related reference**:

"Defining applications" on page 211
The Application Definition allows you to group the transactions controlled by IMS Connect.

"System Definition - Applications" on page 203
When you select **Define Applications for *sysname*** from the System Definitions panel, you are prompted to list the applications controlled by the system.

## SET TRANSACTION

The SET TRANSACTION command causes IMS Connect Extensions definitions for a single transaction to be modified. IMS Connect Extensions reads the input parameters from the panel and rebuilds the in-memory values.



```
   File  Menu  Help
------------------------------------------------------------------------------
                            Set Transaction
Command ===> _________________________________________________________________

IMS Connect system . . : HWS
Transaction name . . . : PART
Description  . . . . . :

Make changes then press Enter to generate commands.

Application . . . IVP_____  +

Transaction Processing Options:
  /  Activate Transaction Routing

    /  Override Application options

    Route transactions to:
    2  1. All Datastores
       2. Datastore . . . . . . IMD4____  +
       3. Datastore Group . . . ________  +
       4. Affinity List . . . . ________  +

    Routing Error processing:
    1  1. Use the original datastore in the message request
       2. Reject the transaction
```

*Figure 103. Set Transaction command panel*

The panel displays the current values (not the values from the Transaction definition in the repository). Change as many values as you wish and press Enter to process.

**Note:**

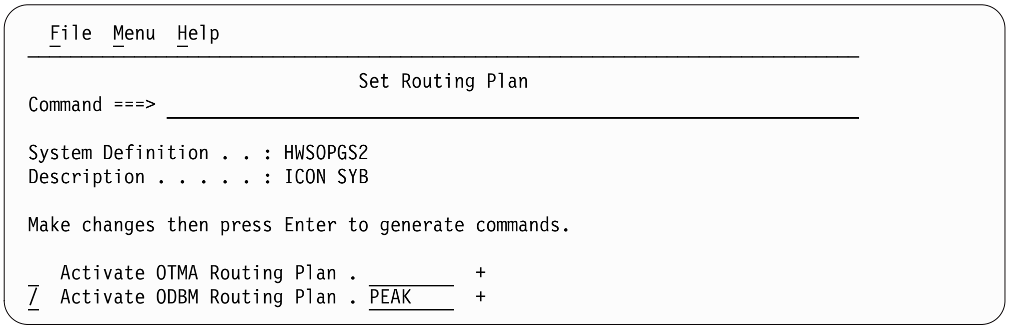
1. If Transaction Routing is active and you do not set the override option, the transaction will take the transaction routing options defined for the parent Application.
2. The SET TRANSACTION command can set Transaction Routing to Active (even when not requested) depending on the parent Application options.
3. You can only change the transaction routing option to a datastore, Datastore Group or Affinity List that is already defined in the Candidate List for this IMS Connect system.

**Related reference**:

"Defining transactions" on page 213
The Transaction Definition allows you to define transactions to be routed by IMS Connect Extensions through the transaction routing feature.

## SET PLAN

The SET PLAN command causes IMS Connect Extensions to switch to a new OTMA or ODBM routing plan.



```
  File  Menu  Help
──────────────────────────────────────────────────────────────────────────────
                               Set Routing Plan
Command ===> _________________________________________________________________

System Definition . . : HWSOPGS2
Description . . . . . : ICON SYB

Make changes then press Enter to generate commands.

_  Activate OTMA Routing Plan . ________  +
/  Activate ODBM Routing Plan . PEAK____  +
```

*Figure 104. Set Routing Plan command panel*

The routing rules will be regenerated when IMS Connect restarts. If you change or deactivate the active OTMA routing plan or ODBM routing plan and want to rebuild the in-memory routing rules and routing lists immediately, refresh the corresponding OTMA Routing Rules or ODBM Routing Rules.

## Refreshing journal settings and switching the active journal

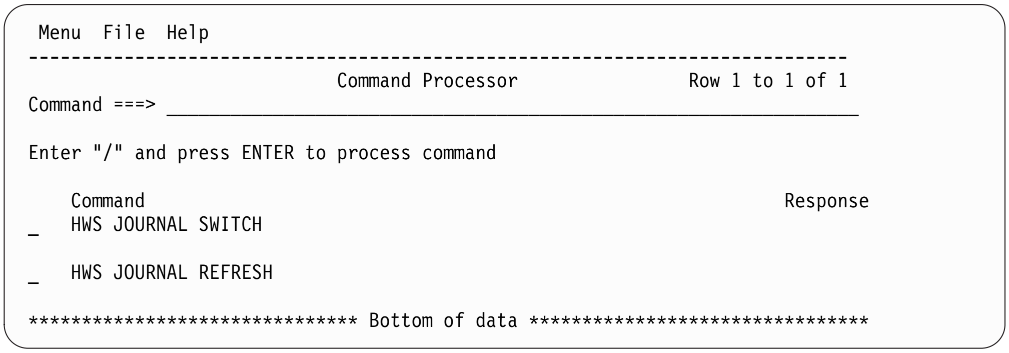
This topic describes the commands for dynamically switching the active journal and refreshing journal settings.

**JOURNAL SWITCH**
Switches the current active journal to the next journal. If the archive journal option is active, starts a new archive job and archives the current active journal.

**JOURNAL REFRESH**
Refreshes the following options:

**Active Journal**
Refreshes the journal full option.

**Archive Journal**
Refreshes the archive journal data set name, device type, job statement information and archive JCL skeleton. You can change other archive journal options by modifying the archive JCL skeleton.



```
 Menu  File  Help
----------------------------------------------------------------------------
                             Command Processor            Row 1 to 1 of 1
Command ===> _________________________________________________________________

Enter "/" and press ENTER to process command

    Command                                                          Response
_   HWS JOURNAL SWITCH

_   HWS JOURNAL REFRESH

******************************* Bottom of data ********************************
```

*Figure 105. JOURNAL command processor panel*

# Part 6. Utilities and APIs reference

These topics provide reference information for the CEX host command environment for REXX and for IMS Connect Extensions definition utilities, IVPs, and APIs.

**Topics:**

# Chapter 21. CEX host command environment: reference

This topic describes the commands and functions that are available in the CEX host command environment for REXX.

This section documents intended Programming Interfaces that allow the customer to write programs to obtain the services of IMS Connect Extensions.

You can use the CEX host command environment to perform tasks such as suspending or resuming routing to a datastore, switching the active journal (equivalent to the JOURNAL SWITCH command), initiating an IMS Connect Extensions resource trace, refreshing definitions (equivalent to the **REFRESH** command), and running commands on a target system or datastore.

**Note:** The host command environment for REXX performs all of the functions of the batch command utility CEXDFCMD and supports some additional options.

**Related concepts**:

Chapter 11, "Automating operations tasks using REXX," on page 123
The CEX host command environment for REXX enables IMS Connect Extensions commands to be embedded in REXX programs, which allows more flexible automation of IMS Connect operations.

**Related reference**:

"ADD" on page 248
The **ADD** command adds an entry to the in-memory datastore table for a datastore that is defined to IMS Connect but was not defined to IMS Connect Extensions at startup.

"CLEAR" on page 249
The **CLEAR** command clears one or more users from the ACEE cache.

"CONNECT" on page 250
The **CONNECT** command connects to an IMS Connect system so that subsequent host commands can be issued to that system.

"DELAY" on page 252
The **DELAY** command pauses command execution for a specified number of seconds.

"DRAIN" on page 252
The **DRAIN** command causes eligible IMS Connect sessions to be marked for closure at the next inbound message. Sessions are closed normally and not canceled.

"OPTION" on page 253
The **OPTION** command is used to specify run-time options for the REXX command environment.

"QUERY" on page 254
The **QUERY** command reports on statistics and settings for several IMS Connect Extensions resources. These include: the current tracing options; the routing status and the number of pending messages for one or more datastores; the ACEE cache status, the active routing plan; and active session details.

"REFRESH" on page 259
The **REFRESH** command rebuilds the in-memory copy of the selected definitions for the target IMS Connect system. IMS Connect Extensions reads the definitions from the repository and rebuilds the in-memory values.

"ROUTE" on page 260
The **ROUTE** command changes the routing status of a datastore to suspend or restore it as a candidate for routing.

"SET" on page 262
The **SET** command activates or deactivates the OTMA routing plan or ODBM routing plan.

"SHELL" on page 263
The **SHELL** command runs a command on a specified target system or datastore. You can use this to automate some procedures such as stopping all the datastores associated with an IMS system across multiple IMS Connect instances.

"SWITCH" on page 264
The **SWITCH** command switches the active journal for an IMS Connect system.

"TRACE" on page 264
Use the **TRACE** host command to start or stop tracing. You can specify filter conditions to control which additional event records are written to the Active Journal data set.

"UPDATE" on page 266
The **UPDATE** command allows certain settings to be changed both in-memory and in the repository definition. These settings include the capacity weight rating for a datastore or Open Database target, and the event collection level or session message limit options for an IMS Connect system.

"WTO" on page 268
The **WTO** command issues a Write To Operator message to the console. You can use this to automate responses to certain conditions.

# ADD

The **ADD** command adds an entry to the in-memory datastore table for a datastore that is defined to IMS Connect but was not defined to IMS Connect Extensions at startup.

When a datastore is to be dynamically added to IMS Connect, it is preferable to add the datastore definition in IMS Connect Extensions *first*. However, if the CREATE IMSCON command has already been issued, you can use the ADD DS host command to add the datastore definition. This makes the datastore available to IMS Connect Extensions without having to restart your IMS Connect instance.

If the new datastore has been defined in one or more routing rules, refresh the OTMA routing rules using the REFRESH host command or Refresh Commands dialog. This makes the dynamically added datastore available for OTMA rules-based routing and visible in the Status Monitor.

**Note:** If the datastore credentials are still cached from when the datastore was originally added to IMS Connect, they are used. If the cached datastore credentials are not available, the details supplied in the ADD host command, including the tmember name and optionally the super member name, are used.

## Syntax

```
►►──ADD──,DS=datastore──,MEMBER=member_name──,TMEMBER=tmember_name──────────────►
```

```
►─┬──────────────────────┬──┬───────────────────────┬─,GROUP=XCF_group──────────►
  └─,SMEMBER=super_member─┘  └─,APPLNAME=appl_group─┘

  ┌─,STEM=CEX──┐
►─┴─,STEM=stem─┴──┬──────────────┬────────────────────────────────────────────►◄
                  └─,CONID=conid─┘
```

## Parameters

**DS** (Required) The name of the datastore that is to be added.

**TMEMBER**
(Required) The XCF member name of the IMS that this IMS Connect communicates with in the XCF group.

**MEMBER**
(Required) The XCF member name that identifies IMS Connect in the XCF group that is specified by the GROUP parameter.

**SMEMBER**
The name of the OTMA super member to which this datastore belongs.

**APPLNAME**
The TCP/IP APPL name defined to RACF in the PTKTDATA statement.

**GROUP**
(Required) The z/OS XCF group for the IMS OTMA.

**STEM** The common stem for the REXX variables that are set by this command. The default stem is CEX.

**CONID**
A unique connection ID that was established by a previous CONNECT command. If no CONID is specified then the current default connection is used. The default connection is the last connection that was specified either by a CONNECT command or by an OPTION command using the CONID keyword.

## Example

The member CEXRXC71 in the SCEXSAMP library contains a detailed example based on the ADD DS host command.

# CLEAR

The **CLEAR** command clears one or more users from the ACEE cache.

## Syntax

```
           ┌─TYPE=ACEE─┐                   ┌─,STEM=CEX──┐
►►──CLEAR──┴───────────┴──,USERID=userid───┴─,STEM=stem─┴──┬──────────────┬──────►◄
                                                          └─,CONID=conid─┘
```

## Parameters

**TYPE** TYPE=ACEE is the default and only option.

**USERID**

The User ID for which you want to clear the ACEE. You can use an asterisk (*) to denote a wildcard. For example, to delete all ACEE structures:

```
USERID=*
```

To delete all user IDs starting with AB:

```
USERID=AB*
```

**STEM** The common stem for the REXX variables that are set by this command. The default stem is CEX.

**CONID**

A unique connection ID that was established by a previous CONNECT command. If no CONID is specified then the current default connection is used. The default connection is the last connection that was specified either by a CONNECT command or by an OPTION command using the CONID keyword.

## Example

The member CEXRXC51 in the SCEXSAMP library contains a detailed example showing how to clear the ACEE cache for a specific user ID or mask.

# CONNECT

The **CONNECT** command connects to an IMS Connect system so that subsequent host commands can be issued to that system.

You can use multiple CONNECT statements in the same job and assign a unique identifier (CONID) to each connection. For example, you could connect to multiple systems and then switch the journal on each system in turn.

## Syntax

```
►►──CONNECT──HWSID=IMS_Connect_system──,HOST=host_name──,PORT=console_port──────►


   ┌─,PASSTIK=NO──,USERID=user_ID──┬──────────────────┬──┐
   │                               └─,PASSWD=password─┘  │ ┌─,STEM=CEX──┐
►──┼─────────────────────────────────────────────────────┼─┼────────────┼──────►
   │                ┌─,APPLID=IMS_Connect_system─┐       │ └─,STEM=stem─┘
   ├─,PASSTIK=YES───┼────────────────────────────┼───────┤
   │                └─,APPLID=application_ID─────┘       │
   │                   ┌─,APPLID=IMS_Connect_system─┐    │
   └─,PASSWD=PASSTIK───┼────────────────────────────┼────┘
                       └─,APPLID=application_ID─────┘

   ┌─,CONID=IMS_Connect_system─┐
►──┼───────────────────────────┼───────────────────────────────────────────────►◄
   └─,CONID=conid──────────────┘
```

The parameters are:

**HWSID**

The name of the IMS Connect system.

**HOST**

A DNS name, IPV4 address, or IPV6 address that identifies the host for the IMS Connect system. For example, `localhost`.

**Note:** The following alternative keywords are supported in place of HOST=*host_name*:

- **DNSNAME**=*DNS_name*
- **IPV4ADR**=*IPv4_address*
- **IPV6ADR**=*IPv6_address*

**PORT**

The TCP/IP port of the IMS Connect Extensions console listener.

**PASSTIK=NO**

If PASSTIK is not specified it defaults to NO.

**USERID**

Enter the user ID associated with the connection. If omitted, the user ID associated with the job step is used.

**PASSWD**

The user password that is associated with *user_ID*. The PASSWD parameter is required if Activate Access Control is enabled in the system definition, and:

- The exec will issue host commands that update the IMS Connect Extensions repository or in-memory definition, such as ADD, REFRESH, or UPDATE, or:
- The exec will issue host commands that affect the operation of IMS or IMS Connect, such as CLEAR, DRAIN, ROUTE, SHELL, or SWITCH.

**PASSTIK=YES**

A PassTicket will be generated for the user ID associated with the job step.

PASSWD=PASSTIK is accepted as a synonym for PASSTIK=YES.

**APPLID**

The application ID is used to generate the PassTicket. It must match the application ID that was specified on the system definition. If an application ID is not explicitly specified in the CONNECT command, it defaults to the name of the IMS Connect system (HWSID).

**STEM**

The common stem for the REXX variables that are set by this command. The default stem is CEX.

**CONID**

A unique connection ID that is established by this CONNECT command and which can be referenced by subsequent commands. If a CONID is not explicitly specified in the CONNECT command, it defaults to the name of the IMS Connect system (HWSID).

Here is an example of a CONNECT control card:

```
⋮
address CEX "CONNECT HOST=SRV2,PORT=4199,HWSID=HWSOPGS1,"||,
                    "PASSTIK=YES,APPLID=MYAPPL,CONID=SYS1"
address CEX "ROUTE ACTION=DRAIN,DSLIST=RLIST5,AUTORESUME,CONID="SYS1"
```

**Note:** If you receive an error message relating to the TCP/IP request, check that you specified the IMS Connect Extensions console port, not another port such as an HWS port. See "Connection failure between a client and a console listener" on page 367.

# DELAY

The **DELAY** command pauses command execution for a specified number of seconds.

## Syntax

```
                        ┌─,STEM=CEX──┐
►►──DELAY──TIME=ssss────┼────────────┼──────────────────────────────────────►◄
                        └─,STEM=stem─┘
```

## Parameters

**TIME** Specifies the time delay in seconds. Enter a value in the range 1 - 3600.

**STEM** The common stem for the REXX variables that are set by this command. The default stem is CEX.

## Example

The CEXRXC23 member in the SCEXSAMP library uses DELAY to poll at set intervals while checking the progress of a ROUTE ACTION=DRAIN command:

```
⋮
call TestSystems
do while pendingResp > 0
  address CEX "DELAY TIME="testInterval
  elapsedTime = elapsedTime + testInterval
  if elapsedTime > timeOut then do
    if SesnRemain = 'SHUTDOWN' then do
⋮
```

# DRAIN

The **DRAIN** command causes eligible IMS Connect sessions to be marked for closure at the next inbound message. Sessions are closed normally and not canceled.

Sessions can be identified by the original client ID or the SVT token associated with the session.

For a session to be eligible for drain, the following conditions apply:

- The session must be a persistent session.
- The session cannot have an active IMS conversation.
- The session cannot have active Resume TPIPE requests.

## Syntax

```
            ┌─TYPE=SESSION──┬──────────────────────┬─┐
            │               ├─,CLIENTID=clientID───┤ │
            │               ├─,TOKEN=svt_token─────┤ │
            │               └─,TOKEN=*─────────────┘ │   ┌─,STEM=CEX──┐
►►──DRAIN───┴────────────────────────────────────────┴───┼────────────┼──────►
                                                          └─,STEM=stem─┘
```

```
►─┬──────────────┬──────────────────────────────────────────────────────►◄
  └─,CONID=conid─┘
```

### Parameters

**TYPE=SESSION**
TYPE=SESSION is the default and only option.

**CLIENTID**
If the session is identified by client ID, you must use the initial client ID value, not the current client ID.

**TOKEN**
The *svt_token* value must match the session token returned by the QUERY TYPE=SESSIONS command. This corresponds to the Event Key that is displayed on the Active Sessions detailed information panel.

TOKEN=* causes all eligible active sessions to be marked for closure.

**STEM** The common stem for the REXX variables that are set by this command. The default stem is CEX.

**CONID**
A unique connection ID that was established by a previous CONNECT command. If no CONID is specified then the current default connection is used. The default connection is the last connection that was specified either by a CONNECT command or by an OPTION command using the CONID keyword.

### Response variables

| Field name | Description |
|---|---|
| DRNCNT | Count of sessions marked for draining |

### Example: DRAIN

```
⋮
address CEX "DRAIN TYPE=SESSION TOKEN=*"
```

The member CEXRXC26 in the SCEXSAMP library contains a detailed example of how to drain persistent sessions using various selection criteria.

## OPTION

The **OPTION** command is used to specify run-time options for the REXX command environment.

### Syntax

```
►►─OPTION─┬────────────────┬──┬───────────────────┬──┬──────────────┬──────►
          ├─MSGLVL=QUIET───┤  ├─,TIMEZONE=CLIENT──┤  └─,CONID=conid─┘
          └─MSGLVL=VERBOSE─┘  ├─,TIMEZONE=GMT─────┤
                              └─,TIMEZONE=SERVER──┘
```

```
   ┌─,STEM=CEX──┐
►──┤            ├─────────────────────────────────────────────────►◄
   └─,STEM=stem─┘
```

## Parameters

**MSGLVL**
Select one of the following:

**VERBOSE**
Displays additional messages before the command output showing the start time and echoing the host commands that will be processed.

**QUIET**
Does not echo messages to the command output. This is the default behavior if an OPTION command is not issued to change it.

**TIMEZONE**
This parameter sets the timezone for other host commands such as SHELL and QUERY TYPE=SESSIONS that display time values. Select one of the following:

**CLIENT**
Displays times using the time offset of the machine executing the utility. This is the default behavior if an OPTION command is not issued to change it.

**GMT** Displays times using the Greenwich Mean Time (GMT) offset.

**SERVER**
Displays times using the time offset of the server. (That is, the machine where the IMS Connect resides that is the target of a subsequent host command.)

**CONID**
A unique connection ID that was established by a previous CONNECT command. Make this CONID the default connection for subsequent commands.

**STEM** The common stem for the REXX variables that are set by this command. The default stem is CEX.

## Example

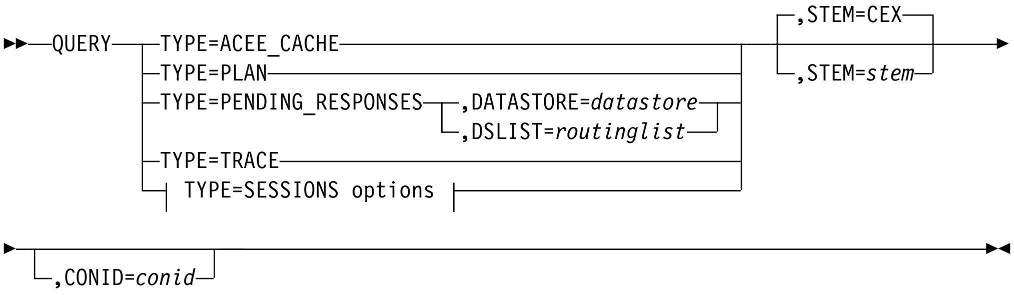
This example shows an OPTION statement being used to change the default connection to the system identified by CONID=SRV3 before issuing a query:

```
⋮
OPTION MSGLVL=VERBOSE,TIMEZONE=GMT,CONID=SRV3
QUERY TYPE=SESSIONS,PERSISTENT=Y,SESTYPE=OTMA
```

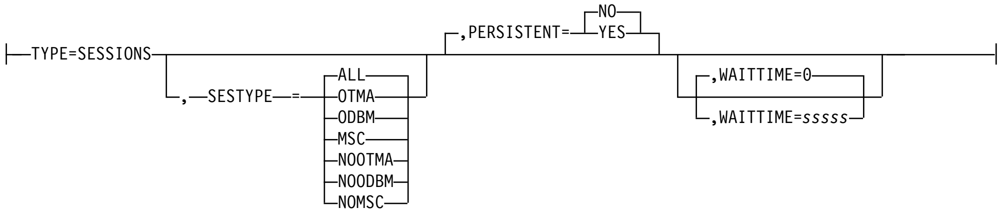
# QUERY

The **QUERY** command reports on statistics and settings for several IMS Connect Extensions resources. These include: the current tracing options; the routing status and the number of pending messages for one or more datastores; the ACEE cache status, the active routing plan; and active session details.

## Syntax



### TYPE=SESSIONS options:



## Parameters

**TYPE=ACEE_CACHE**

Display information about the caching of security credentials in IMS Connect to improve performance.

The information is presented in two sections: `Cache space` and `Accounting stats`. The first relates to the space usage in the ACEE cache. The second presents statistics on different types of cache services requests.

**Response variables**

| Field name | Description |
|---|---|
| ACEEPCT | Percentage of the ACEE cache used |
| DELETES | Number of elements deleted from the ACEE cache |
| ELEMLEN | Length of each element in the ACEE cache |
| ELEMNUM | Number of elements in the ACEE cache |
| GETNEXTS | Number of get next requests to the ACEE cache |
| GETS | Number of get requests to the ACEE cache |
| INITSIZE | Initial size of the ACEE cache |
| INSERTS | Number of elements inserted in the ACEE cache |
| STATS | Number of stat requests to the ACEE cache |
| XPNDCNT | The number of times the ACEE cache has been expanded |
| XPNDMAX | The maximum number of times the ACEE cache can be expanded |
| XPNDSIZE | The increment by which the ACEE cache can be expanded |

**Example**

The CEXRXC44 member in the SCEXSAMP library contains a detailed example of how to report ACEE cache statistics for a system.

**TYPE=PENDING_RESPONSES**

Returns the number of messages that are pending a response from IMS (that is, messages that are in a Waiting for Datastore state). The command output is echoed in a message in the JES message log. You can use this in an automated script to drain and then safely shut down an IMS system when the number of pending inbound and outbound messages falls to zero.

**DATASTORE**

A single datastore that is to be referenced by QUERY command processing.

**DSLIST**

An OTMA routing list comprising the datastores that are to be referenced by QUERY command processing.

**Example**

The CEXRXC21, CEXRXC22, and CEXRXC23 members in the SCEXSAMP library contain examples of QUERY TYPE=PENDING_RESPONSES being used to report progress of a datastore drain.

| Field name | Description |
|---|---|
| DST.0 | The number of datastores. Subsequent fields are qualified with a numeric in the range 1 to DST.0 |
| DSTNAME.*n* | The datastore names |
| PENDING.*n* | Pending response counts |
| STATUS.*n* | Status of the datastores |

**Examples**

The CEXRXC21, CEXRXC22, and CEXRXC23 members in the SCEXSAMP library contain detailed examples of how to drain one or more datastores and then shut down and restart the IMS system.

**TYPE=PLAN**

Displays the names of the currently active OTMA routing plan and ODBM routing plan, or blank if they are inactive.

**Tip:** You can test the REXX variable containing the plan name and then issue a SET PLAN command to change it.

**Response variables**

| Variable name | Description |
|---|---|
| OTMAPLAN | Routing plan used for OTMA messages (blank if no OTMA routing plan is inactive) |
| ODBMPLAN | Routing plan used for DRDA requests (blank if no ODBM routing plan is inactive) |

**Examples**

The CEXRXC31 member in the SCEXSAMP library contains an example of how to set an OTMA routing plan or ODBM routing plan.

**TYPE=SESSIONS**

Returns active session attributes in REXX stem variables. You can use the OPTION host command to specify a timezone offset for displaying time fields.

**PERSISTENT**

Specify PERSISTENT=Y to include sessions with idle persistent sockets. Idle persistent sockets are those persistent socket sessions that are predicted to be in the READ PREPARE state.

**SESTYPE**

Use this field to filter the types of sessions displayed:

**ALL** Return all sessions

**OTMA**
Return OTMA sessions

**ODBM**
Return ODBM sessions

**MSC** Return MSC sessions

**NOOTMA**
Exclude OTMA sessions

**NOODBM**
Exclude ODBM sessions

**NOMSC**
Exclude MSC sessions

**WAITTIME**

Specify WAITTIME=*sssss* to return all matching sessions whose wait time is greater than *sssss*. WAITTIME=0 returns all matching sessions, whatever their value for wait time.

**Response variables**

| **Field name** | **Description** |
|---|---|
| SESSION.0 | Count of session data returned |
| SESSION.CLIENTFAMILY.*n* | IPv4 or IPv6 |
| SESSION.CLIENTID.*n* | Client ID |
| SESSION.CLIENTIPADDRESS.*n* | Client IP address |
| SESSION.CLIENTPORT.*n* | Client port number |
| SESSION.COMMITMODE.*n* | Commit mode |
| SESSION.EXITDEFINED.*n* | Exit defined flag, value is YES or NO |
| SESSION.EXITNAME.*n* | Name of the exit that processed the session |
| SESSION.ICLIENTID.*n* | Identifier of the initial connecting client |
| SESSION.INCONVERSATION.*n* | Conversation in progress |
| SESSION.INPUTALIAS.*n* | Routing exit input alias |
| SESSION.IRMTIMER.*n* | Response reason code |
| SESSION.LASTTRACETIME.*n* | Time last event occurred on the session |
| SESSION.LTERM.*n* | LTERM override name |
| SESSION.MSCCONNUID.*n* | MSC connection user ID |
| SESSION.MSCGENIMSID.*n* | MSC generic IMS identifier |

| Field name | Description |
|---|---|
| SESSION.MSCLCLIMSID.*n* | MSC local IMS identifier |
| SESSION.MSCLCLPLKID.*n* | MSC local physical link |
| SESSION.MSCPARTNERID.*n* | MSC partner |
| SESSION.MSCRC.*n* | MSC return code |
| SESSION.MSCREMOTEIMSID.*n* | MSC remote IMS identifier |
| SESSION.MSCRMTPLKID.*n* | MSC remote physical link |
| SESSION.MSCRSN.*n* | MSC reason code |
| SESSION.ODBMNAME.*n* | Routing exit ODBM name |
| SESSION.ORIGINALDS.*n* | Datastore specified in the message as the target |
| SESSION.OUTPUTALIAS.*n* | Routing exit output alias |
| SESSION.OUTPUTGROUP.*n* | Security exit output group |
| SESSION.OUTPUTUSERID.*n* | Security exit output user ID |
| SESSION.PORT.*n* | Port |
| SESSION.PREDICTEDSTATUS.*n* | Description of session status |
| SESSION.PSBNAME.*n* | Routing exit PSB name |
| SESSION.READEXITRC.*n* | Exit return code |
| SESSION.READEXITRSN.*n* | Exit reason code |
| SESSION.REROUTENAME.*n* | Reroute name |
| SESSION.ROUTINGEXITRC.*n* | Routing exit return code |
| SESSION.ROUTINGEXITRSN.*n* | Routing exit reason code |
| SESSION.ROUTINGSERVRC.*n* | Routing server return code |
| SESSION.ROUTINGSERVRSN.*n* | Routing server reason code |
| SESSION.RTPIPEACTIVE.*n* | RTPIPE is active |
| SESSION.SECURITYEXITRC.*n* | Security exit return code |
| SESSION.SECURITYEXITRSN.*n* | Security exit reason code |
| SESSION.SECURITYSERVRC.*n* | Security server return code |
| SESSION.SECURITYSERVRSN.*n* | Security server reason code |
| SESSION.SESMSGCNT.*n* | Message count for the session |
| SESSION.SESSIONTYPE.*n* | Session type |
| SESSION.SOCKET.*n* | Socket number |
| SESSION.SOCKETTYPE.*n* | Socket type |
| SESSION.STARTTIME.*n* | Session start time |
| SESSION.SVTTOKEN.*n* | Session token |
| SESSION.SYNCHLEVEL.*n* | Synch level |
| SESSION.TARGETDS.*n* | Datastore assigned as the target by IMS Connect Extensions |
| SESSION.TRANSACTION.*n* | Transaction code |
| SESSION.TRIGGERTYPE.*n* | Trigger type |
| SESSION.USERID.*n* | User ID |
| SESSION.WAITTIME.*n* | Time since last event |

| Field name | Description |
|---|---|
| SESSION.WRITEEXITRC.*n* | Write exit return code |
| SESSION.WRITEEXITRSN.*n* | Write exit reason code |
| SESSION.XMLADAPTERNAME.*n* | XML adapter name |
| SESSION.XMLMAPNAME.*n* | XML map name |

**Examples**

The CEXRXC42 member in the SCEXSAMP library uses QUERY TYPE=SESSIONS to return details of persistent sessions on a system. The CEXRXC26 member uses QUERY TYPE=SESSIONS to report the progress of a DRAIN TYPE=SESSION command.

**TYPE=TRACE**

Reports whether tracing is active or inactive on the system and, if it is active, the current tracing options.

**Response variables**

| Field name | Description |
|---|---|
| ACTIVE | Indicates whether tracing is active |
| CONDNAME | The specific condition for which tracing is requested |
| CONDTYPE | The type of condition for which tracing is requested |
| LEVEL | Tracing level |
| PERSISTS | Indicates whether "KEEP_ACTIVE" was specified |
| PORT | Tracing port |

**Examples**

The CEXRXC45 member in the SCEXSAMP library contains a detailed example of how to report the tracing settings for a system.

**STEM** The common stem for the REXX variables that are set by this command. The default stem is CEX.

**CONID**

A unique connection ID that was established by a previous CONNECT command. If no CONID is specified then the current default connection is used. The default connection is the last connection that was specified either by a CONNECT command or by an OPTION command using the CONID keyword.

**Related tasks**:

"Querying the status of tracing" on page 86
You can query whether tracing is active for an IMS Connect system through the Operations dialog or Status Monitor.

# REFRESH

The **REFRESH** command rebuilds the in-memory copy of the selected definitions for the target IMS Connect system. IMS Connect Extensions reads the definitions from the repository and rebuilds the in-memory values.

### Syntax

Use REFRESH in the following situations:

- When repository changes are to be activated.
- To revert to the repository definitions when a SET command has made dynamic changes.

```
                                                      ┌─,STEM=CEX──┐
►►──REFRESH──DEFN=(─┬─AFL──,──affinity_list───┬─)──┼────────────┼──────────────────►
                    ├─APP──,──application─────┤    └─,STEM=stem─┘
                    ├─DS──,──datastore────────┤
                    ├─DSG──,──datastore_group─┤
                    ├─EX──,──exit─────────────┤
                    ├─HWS─────────────────────┤
                    ├─JRN─────────────────────┤
                    ├─ORR─────────────────────┤
                    ├─RTG─────────────────────┤
                    ├─SAF─────────────────────┤
                    └─TXN──,──transaction─────┘

►──┬────────────────┬─────────────────────────────────────────────────────────────►◄
   └─,CONID=conid───┘
```

### Parameters

**DEFN=(*type*,*name*)**

*type* is the definition type to be refreshed. For example, to refresh all OTMA routing rules:

```
address CEX "REFRESH DEFN=(RTG)"
```

For some definition types *name* can be used to refresh a specified definition. For example, to refresh a datastore named IMD4:

```
address CEX "REFRESH DEFN=(DS,IMD4)"
```

**STEM** The common stem for the REXX variables that are set by this command. The default stem is CEX.

**CONID**

A unique connection ID that was established by a previous CONNECT command. If no CONID is specified then the current default connection is used. The default connection is the last connection that was specified either by a CONNECT command or by an OPTION command using the CONID keyword.

### Example

The member CEXRXC31 in the SCEXSAMP library shows an example of REFRESH being used after activating a routing plan.

## ROUTE

The **ROUTE** command changes the routing status of a datastore to suspend or restore it as a candidate for routing.

The ROUTE command has two modes:

- ROUTE ACTION=DRAIN changes the routing status of each specified datastore from Normal to Suspended and withdraws it as a candidate for routing.
- ROUTE ACTION=RESUME changes the routing status of each specified datastore from Suspended to Normal and restores it as a candidate for routing.

To assist with automated operations, the result of these commands is echoed in messages in the JES message log. For example:

```
CEX5131I ROUTE DRAIN for Datastore DS01, AUTORESUME is not set.
CEX5131I ROUTE DRAIN for Datastore DS02, AUTORESUME is set.
CEX5131I ROUTE DRAIN for Datastore DS03, AUTORESUME is not set. Datastore is already draining.

CEX5131I ROUTE RESUME for Datastore DS01.
CEX5132W ROUTE RESUME for Datastore DS03 Datastore is not currently drained
```

**Note:** The Drain/Resume feature only affects routing done by IMS Connect Extensions. It has no effect on routing that might be done by the IMS Connect message exit.

## Syntax

```
               ┌─ACTION=DRAIN─┬─────────────────────┬─┬─,AUTORESUME─┐
               │              ├─,DATASTORE=datastore─┤ │             │
               │              └─,DSLIST=routinglist──┘ │             │
►►──ROUTE──────┴─ACTION=RESUME─┬──────────────────────┴─────────────┴──────────►
                               ├─,DATASTORE=datastore─┤
                               └─,DSLIST=routinglist──┘

  ┌─,STEM=CEX──┐
►─┴────────────┴─┬──────────────┬──────────────────────────────────────────────►◄
  └─,STEM=stem─┘ └─,CONID=conid─┘
```

## Parameters

**ACTION**
Select one of the following:

**DRAIN**
All datastores referenced by the command are marked as Drained in terms of candidacy for routing. A datastore in Drained status will not be a candidate for routing.

**RESUME**
All datastores referenced by the command have their Drained status removed and become candidates for routing.

**DATASTORE**
A single datastore that is to be referenced by ROUTE command processing.

**DSLIST**
An OTMA routing list comprising the datastores that are to be referenced by ROUTE command processing.

**AUTORESUME**
This optional keyword indicates that the datastore is to be automatically resumed when an IMS Datastore Available event record (X'10') is received. If AUTORESUME is specified it applies only to datastores that are the target of the ROUTE command.

AUTORESUME can only be used with ACTION=DRAIN.

**Note:** AUTORESUME requires a fix to be applied for an IMS Connect issue ("The DSIB control block passed from IMS Connect should contain the datastore name. Instead, it contains the IMS XCF member name.") The required APAR for IMS Connect V12 is PM85243. The required APAR for IMS Connect V11 is PM70277.

**STEM** The common stem for the REXX variables that are set by this command. The default stem is CEX.

**CONID**
A unique connection ID that was established by a previous CONNECT command. If no CONID is specified then the current default connection is used. The default connection is the last connection that was specified either by a CONNECT command or by an OPTION command using the CONID keyword.

### Examples

The CEXRXC21, CEXRXC22, and CEXRXC23 members in the SCEXSAMP library contain detailed examples of how to drain one or more datastores and then shut down and restart the IMS system.

## SET

The **SET** command activates or deactivates the OTMA routing plan or ODBM routing plan.

If you change or deactivate the active OTMA or ODBM routing plan you must then refresh the corresponding OTMA Routing Rules or ODBM Routing Rules to rebuild the in-memory routing rules and routing lists.

### Syntax

```
                                                                ┌─,STEM=CEX──┐
►►──SET──PLAN───┬─,TYPE=OTMA─┬───┬─,ACTIVATE=routing_plan─┬───┼────────────┼──────────►
                └─,TYPE=ODBM─┘   └─,DEACTIVATE────────────┘   └─,STEM=stem─┘

►──┬───────────────┬───────────────────────────────────────────────────────────────►◄
   └─,CONID=conid──┘
```

### Parameters

**TYPE** The routing plan type: OTMA or ODBM.

**ACTIVATE | DEACTIVATE**
ACTIVATE=*routing_plan* activates the named routing plan.

DEACTIVATE means that no routing plan of the specified type will be active. When no routing plan is activated, only unconditional rules are active on the current system. Unconditional rules are rules that are not assigned to a routing plan.

**STEM** The common stem for the REXX variables that are set by this command. The default stem is CEX.

**CONID**

A unique connection ID that was established by a previous CONNECT command. If no CONID is specified then the current default connection is used. The default connection is the last connection that was specified either by a CONNECT command or by an OPTION command using the CONID keyword.

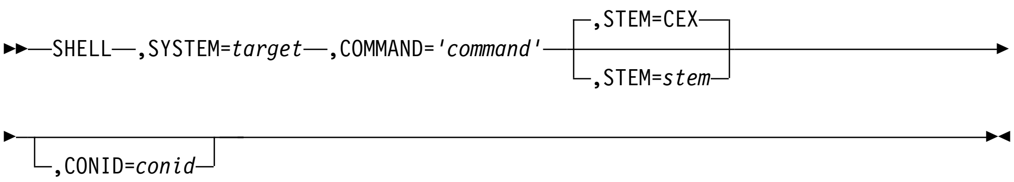
Here is an example of a pair of commands to activate an OTMA routing plan and then rebuild the in-memory OTMA routing rules and routing lists:

```
⋮
address CEX "SET PLAN,TYPE=OTMA,ACTIVATE=PEAK"
address CEX "REFRESH DEFN=RTG"
```

# SHELL

The **SHELL** command runs a command on a specified target system or datastore. You can use this to automate some procedures such as stopping all the datastores associated with an IMS system across multiple IMS Connect instances.

## Syntax



## Parameters

**SYSTEM**

The target system or datastore on which the command is to run.

**COMMAND**

The command to be executed. If *target* is set to the name of an IMS Connect instance then the command is treated as an IMS Connect command, as in this example:

```
SHELL SYSTEM=HWSOPPS1,COMMAND='STOPIP PLXDH'
```

If *target* is set to a datastore name then the command is treated as an IMS Type-1 command:

```
SHELL SYSTEM=IMSD,COMMAND='/SWITCH OLDS'
```

**STEM** The common stem for the REXX variables that are set by this command. The default stem is CEX.

**CONID**

A unique connection ID that was established by a previous CONNECT command. If no CONID is specified then the current default connection is used. The default connection is the last connection that was specified either by a CONNECT command or by an OPTION command using the CONID keyword.

## Response variables

| Field name | Description |
|---|---|
| TARGET | Shell command target |
| TIME | Shell command time of day |
| LINE.0 | Shell command response line count. Subsequent lines are qualified with a numeric in the range 1 to LINE.0 |
| LINE.*n* | Shell command response line |

## Example

The CEXRXC02 and CEXRXC40 members in the SCEXSAMP library contain detailed examples of how to issue IMS Connect commands and IMS Type-1 commands using the SHELL host command.

# SWITCH

The **SWITCH** command switches the active journal for an IMS Connect system.

## Syntax

```
                          ┌─,STEM=CEX──┐
►►──SWITCH──TYPE=JOURNAL──┼────────────┼──┬───────────────┬──────────────►◄
                          └─,STEM=stem─┘  └─,CONID=conid──┘
```

## Parameters

**TYPE=JOURNAL**
This parameter is required.

**STEM** The common stem for the REXX variables that are set by this command. The default stem is CEX.

**CONID**
A unique connection ID that was established by a previous CONNECT command. If no CONID is specified then the current default connection is used. The default connection is the last connection that was specified either by a CONNECT command or by an OPTION command using the CONID keyword.

## Example

The CEXRXC01 member in the SCEXSAMP library issues the SWITCH command to switch the active journal for an IMS Connect system.

# TRACE

Use the **TRACE** host command to start or stop tracing. You can specify filter conditions to control which additional event records are written to the Active Journal data set.

## Syntax

```
                                ┌─,KEEP_ACTIVE=NO──┐   ┌─,PORT=*─────┐
►►──TRACE──┬─LEVEL=0─┬──────────┼──────────────────┼───┼─────────────┼──────────────────────►
           ├─LEVEL=1─┤          └─,KEEP_ACTIVE=YES─┘   ├─,PORT=port──┤
           └─LEVEL=2─┘                                 └─,PORT=LOCAL─┘

                                                     ┌─,STEM=CEX──┐
►──┬─────────────────────────────────────────────┬───┼────────────┼───┬──────────────┬──►◄
   ├─,COND=(CLIENT_NAME,client_name)─────────────┤   └─,STEM=stem─┘   └─,CONID=conid─┘
   ├─,COND=(TRANSACTION,transaction_name)────────┤
   ├─,COND=(MESSAGE_EXIT,exit_name)──────────────┤
   ├─,COND=(USERID,user_ID)──────────────────────┤
   ├─,COND=(LTERM,logical_terminal)──────────────┤
   └─,COND=(IP_ADDRESS,host_name_or_IP)──────────┘
```

## Parameters

**LEVEL**

Determines whether the application data section is included in the tracing event records. This option is only applicable to an OTMA workload.

**0** Set tracing to off.

**1** Set tracing on.

**2** Set tracing on and include client application data in the trace records.

**KEEP_ACTIVE**

Tracing continues until it is deactivated or until a system restart. You can enable tracing to continue after a shutdown and restart of IMS Connect by setting KEEP_ACTIVE=YES. You can only make a trace persistent at the same time as you activate it.

**PORT** Enter a decimal value in the range 1 - 65535 to specify which IMS Connect TCP/IP port you want to initiate a trace against. Use LOCAL to trace against the local port.

**COND**

Set the condition which writes a trace record. For example, if you want to trace activity for a particular transaction use the TRANSACTION condition. You can use one of the following conditions:

**CLIENT_NAME**

The client name. The value can be up to 8 characters. It must not start with a number.

**TRANSACTION**

Transaction code. The value can be up to 8 characters. It must not start with a number.

**MESSAGE_EXIT**

User message exit processing the messages. The value can be up to 8 characters. It must not start with a number.

**USERID**

User ID specified in the message. The value can be up to 8 characters. The value must specify a string that represents a valid user ID, an asterisk, or a partial user ID ending in an asterisk (*). No other wildcards are permitted.

**LTERM**

Logical Terminal name. The value can be up to 8 characters. It must not start with a number.

**IP_ADDRESS**

The IPv4 address, IPv6 address, or host name of the client. If you enter a host name, IMS Connect Extensions will attempt to resolve the IP address.

**Note:** Conditional tracing is not applicable when tracing ODBM activity. To trace ODBM activity, simply specify the port.

**Note:** You do not need to specify conditions if you are only deactivating tracing (LEVEL=0).

**STEM** The common stem for the REXX variables that are set by this command. The default stem is CEX.

**CONID**

A unique connection ID that was established by a previous CONNECT command. If no CONID is specified then the current default connection is used. The default connection is the last connection that was specified either by a CONNECT command or by an OPTION command using the CONID keyword.

### Examples

Here are example TRACE statements:

```
⋮
address CEX "TRACE LEVEL=1,PORT=8889,COND=(USERID,HWSUSR)"
address CEX "TRACE LEVEL=2,PORT=8889,COND=(TRANSACTION,PARTX)"
address CEX "TRACE LEVEL=0"
```

The CEXRXC11 member in the SCEXSAMP library contains a detailed example of how to change OTMA tracing settings.

**Related tasks**:

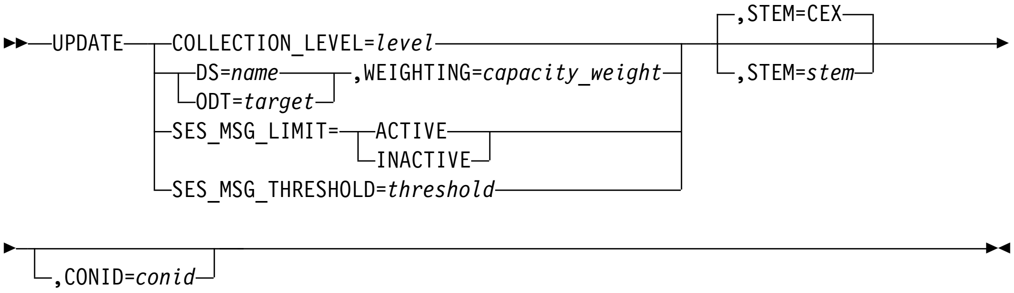
"Starting the IMS Connect Extensions trace" on page 85
You can start tracing for an active IMS Connect system through the Operations dialog or Status Monitor.

"Stopping the IMS Connect Extensions trace" on page 87
You can stop tracing for an active IMS Connect system through the Operations dialog or Status Monitor.

## UPDATE

The **`UPDATE`** command allows certain settings to be changed both in-memory and in the repository definition. These settings include the capacity weight rating for a datastore or Open Database target, and the event collection level or session message limit options for an IMS Connect system.

### Syntax



## Parameters

**COLLECTION_LEVEL**

The Event Collection level determines which events are actually recorded in the journal. The acceptable field values are:

**0** Minimum level. Collects startup and shutdown events along with some infrequent error events.

**1** Accounting level. Collects Return from Exit events, OTMA timeout and session error events, and ODBM registration and routing events. This level provides accounting information in terms of the number of messages by Transaction, User Exit, and so on.

**2** Transit time reporting. Collects the minimum number of records to run simple transit time reports.

**3** Comprehensive performance analysis. Collects all TCP/IP read and write events which provides for analysis of TCP/IP activity and remote ICON connectivity.

**4** Maximum level. Collects **all** event records.

You could use this, for example, to reduce IMS Connect Extensions resource usage at peak times or when event collection is not required.

**SES_MSG_LIMIT**

The Session Message Limit feature specifies the maximum number of input messages a persistent session can receive before it is closed. If SES_MSG_LIMIT=ACTIVE, the SYSPLEX distributor can rebalance the session workload between available IMS Connect systems.

**SES_MSG_THRESHOLD**

Specifies the number of input messages a persistent session can receive before it will be closed. This value must be between 1 and 999999 if the Session Message Limit feature is ACTIVE.

**Restriction:**

1. Only one of the main keywords (COLLECTION_LEVEL, SES_MSG_LIMIT, SES_MSG_THRESHOLD, WEIGHTING) can be specified per **UPDATE**. Therefore, setting both SES_MSG_LIMIT and SES_MSG_THRESHOLD requires separate **UPDATE** calls.
2. The order of the calls is significant. You can't first activate SES_MSG_LIMIT with an invalid threshold and then update SES_MSG_THRESHOLD to a valid value. Instead, update SES_MSG_THRESHOLD to a valid value and then set SES_MSG_LIMIT=ACTIVE.

**WEIGHTING**
The relative capacity weighting of each target. You can keep a target in reserve by assigning it a capacity of zero, making it temporarily ineligible as a routing candidate. Specify a value in the range 0 - 100.

**DS** The target datastore.

**ODT** The ODBM target.

**STEM** The common stem for the REXX variables that are set by this command. The default stem is CEX.

**CONID**
A unique connection ID that was established by a previous CONNECT command. If no CONID is specified then the current default connection is used. The default connection is the last connection that was specified either by a CONNECT command or by an OPTION command using the CONID keyword.

### Examples

The CEXRXC61, CEXRXC62, and CEXRXC63 members in the SCEXSAMP library contain examples of how to change various settings using the UPDATE command.

## WTO

The **WTO** command issues a Write To Operator message to the console. You can use this to automate responses to certain conditions.

The WTO is issued with ROUTCDE=11. This is commonly referred to as “write to programmer” (WTP). The message is intended for the problem programmer.

This routing code is used when the program issuing the message cannot route the message to the programmer through a system output (SYSOUT) data set. The message appears in the JESYSMSG data set. If the message is issued by a TSO user, the message is also sent to the TSO user's screen.

### Syntax

```
                              ┌─,STEM=CEX──┐
►►──WTO──TEXT=message_text────┼────────────┼──────────────────────────────►◄
                              └─,STEM=stem─┘
```

### Parameters

**TEXT** A quoted or unquoted string of up to 126 characters. If the string is not quoted then it is changed to uppercase. Strings containing blanks must be quoted.

**STEM** The common stem for the REXX variables that are set by this command. The default stem is CEX.

### Examples

The CEXRXC22 and CEXRXC23 members in the SCEXSAMP library contain examples of using WTO to report a timeout condition.

# Chapter 22. Definition utilities

IMS Connect Extensions provides utilities to administer its definitions.

These utilities allow you to perform tasks such as:

- Migrate definitions from test systems to production systems.
- Migrate definitions from one site to another.
- Create new definitions.
- Transform multiple definitions.
- Verify the integrity of the repository.

**Definition maintenance utility**
Allows you to add and delete definitions from a repository.

**Definition takeup utility**
The utility reads the configuration member of an IMS Connect system and produces definitions for that system, its datastores, and its exits. You enter definitions produced by this utility into an IMS Connect Extensions repository with the definition maintenance utility.

**Definition extract utility**
Reads IMS Connect Extensions repositories and extracts statements for the definition maintenance utility. This utility can apply transformations to the definitions it extracts. For example, change the prefix on all datastore names.

**Import/export utility**
Imports and exports definitions in batch and validates the integrity of the repository.

## General syntax

The rules for the SYSIN control cards are:

- To continue a command across multiple lines enter a non-blank character in column 72.
- Comments begin with an asterisk (*) in column 1 and can be 1 - 79 characters in length. For example:

```
*here is a comment.
```

**Related concepts**:

"Repository architecture" on page 17
Each instance of IMS Connect with IMS Connect Extensions refers to a repository, which contains configuration data for the IMS Connect Extensions components that start with IMS Connect.

# Definition maintenance utility

The definition maintenance utility allows you to add and delete definitions. You can produce input for this utility manually or generate input with either the takeup utility or the extract utility.

The takeup utility produces input for this utility from the IMS Connect configuration member (HWSCFG); the extract utility produces input for this utility from an IMS Connect Extensions repository.

The maintenance utility program name is CEXDFDSI. You can invoke the program by passing its name to the FUNEXEC program; you can run FUNEXEC from JCL as an executable job step program.

## Definition maintenance utility - example JCL

This example JCL adds definitions to a repository:

```
⋮
//STEP     EXEC PGM=FUNEXEC,PARM='CEXDFDSI'
//STEPLIB  DD DISP=SHR,DSN=funpre.SFUNLINK
//         DD DISP=SHR,DSN=cexpre.SCEXLINK
//REPOSTRY DD DISP=SHR,DSN=definitions.data.set
//MSGOUT   DD SYSOUT=*
//SYSIN    DD *
*Add an HWS system
ADD DEFN=(HWS,HWS1),                                                    +
    CEX_PORT=1234,                                                      +
    HOST=LOCALHOST,                                                     +
    ACTIVE_DS=('MY.ACTIVE.&ID'),                                        +
    ARCHIVE_DS=('MY.ARCH(+1)',                                          +
                ARCHIVE_JCL='SCEXSAMP(CEXARCH)')

*Add an exit
ADD DEFN=(EXIT,HWSJAVA1)

*Add a datastore
ADD DEFN=(DS,IMS1)
⋮
more batch commands
/*
```

## Job control statements

The job control statements in the preceding example are:

**STEP EXEC**
Specifies the program name FUNEXEC with the name of the maintenance utility, CEXDFDSI, as a parameter.

**STEPLIB DD**
IMS Connect Extensions batch utilities require modules in the IMS Connect Extensions (CEX) link library and its functional support (FUN) link library.

**MSGOUT DD**
Defines a sequential data set for message output. The data set can be written to a system output device, a tape, or a DASD volume. This data set has the following attributes:
LRECL=133
RECFM=FBA

**REPOSTRY DD**
Specifies the IMS Connect Extensions repository into which you want to insert definitions. You must use REPOSTRY as the ddname.

**SYSIN DD**
Defines an input sequential data set (SYSIN) containing the batch commands that you want to process. The data set can be defined as job

control in-stream data (as shown in the preceding example), a tape file, or a DASD file. This data set has the following attributes:

LRECL=80
RECFM=FB

## ADD command: general syntax

The definition maintenance utility **ADD** command adds an IMS Connect Extensions definition to a repository, optionally replacing an existing definition.

### Format

```
                                                          ┌─,REPLACE=NO──┐
►►──ADD──DEFN=(──┬─*───┬─,name)──┬─────────────────────┬──┼──────────────┼──────────►
                 ├─AFL─┤         └─,DESC='description'─┘  └─,REPLACE=YES─┘
                 ├─APP─┤
                 ├─DS──┤
                 ├─DSG─┤
                 ├─EX──┤
                 ├─GRP─┤
                 ├─HWS─┤
                 ├─ORL─┤
                 ├─ORR─┤
                 ├─ORT─┤
                 ├─PLN─┤
                 ├─PSB─┤
                 ├─RTG─┤
                 ├─RTL─┤
                 ├─TXL─┤
                 └─TXN─┘

►──┤ parameters for definition type ├────────────────────────────────────────────►◄
```

### Parameters

**DEFN**=(*type*,*name*)
The definition type and its name. For example, to specify an IMS Connect system definition named MYSYS type:

```
DEFN=(HWS,MYSYS)
```

For information about the different parameters for each definition type, see:

- "Affinity list (AFL) ADD command" on page 272
- "Application (APP) ADD command" on page 273
- "Datastore (DS) ADD command" on page 274
- "Datastore group (DSG) ADD command" on page 277
- "User exit (EX) ADD command" on page 278
- "System group (GRP) ADD command" on page 280
- "IMS Connect system (HWS) ADD command" on page 281
- "ODBM routing list (ORL) ADD command" on page 291
- "ODBM routing rule (ORR) ADD command" on page 292
- "ODBM target (ORT) ADD command" on page 294

- "PSB name list (PSB) ADD command" on page 295
- "Routing plan (PLN) ADD command" on page 296
- "OTMA routing rule (RTG) ADD command" on page 297
- "OTMA routing list (RTL) ADD command" on page 300
- "Transaction list (TXL) ADD command" on page 301
- "Transaction code (TXN) ADD command" on page 302

**DESC**='*definition description*'
A description of the definition. A description that contains spaces must be enclosed in quotation marks. The description can be up to 30 characters long.

**REPLACE**=**NO** | **YES**
Determines what happens if you try to input a definition with the same type and name as one that already exists in the repository:

**YES** The definition replaces the existing definition.

**NO** All ADD commands directed at the repository fail.

## Affinity list (AFL) ADD command

The DEFN AFL ADD command adds an affinity list to a repository. An affinity list is a list of datastores that you can use to set potential candidates for message routing. You only need this definition if you use routing.

### Format

```
                                                ┌─,REPLACE=NO──┐
►►──DEFN=(AFL,name)──┬──────────────────────┬───┴─,REPLACE=YES─┴──────────────────►
                     └─,DESC='description'──┘

          ┌─,──────┐
          ▼        │
►─,DSLIST=(──dsname──┴─)──────────────────────────────────────────────────────────►◄
```

### Parameters

**DEFN**=(**AFL**,*name*)
Define an affinity list called *name*.

**DESC**='*definition description*'
A description of the definition. A description that contains spaces must be enclosed in quotation marks. The description can be up to 30 characters long.

**REPLACE**=**NO** | **YES**
Determines what happens if you try to input a definition with the same type and name as one that already exists in the repository:

**YES** The definition replaces the existing definition.

**NO** All ADD commands directed at the repository fail.

**DSLIST**=(*datastores,...*)
A comma-delimited list of datastores.

## Application (APP) ADD command

The DEFN APP ADD command adds an application to a repository. An application is a list of transaction codes with routing options for those codes. Application definitions associate transactions with a system. They are required for transaction routing and other transaction-based features.

### Format

```
                                          ┌─,REPLACE=NO──┐
►►──DEFN=(APP,name)──┬─────────────────────┬─┼──────────────┼──────────────────────────────────►
                     └─,DESC='description'─┘ └─,REPLACE=YES─┘

  (1) ┌─,TRANSACTION_ROUTING=INACTIVE─┐                                                   (2)
►─────┼───────────────────────────────┼─┬─────────────────────────────────────────────────┬──────►◄
      └─,TRANSACTION_ROUTING=ACTIVE───┘ │                           ┌─,FAILURE=ORIGINAL─┐  │
                                        └─,ROUTING_RULE=(─┬─ALL──────┬─┼───────────────────┼─)┘
                                                          ├─DS,name──┤ └─,FAILURE=REJECT───┘
                                                          ├─DSG,name─┤
                                                          └─AFL,name─┘
```

**Notes:**

1 ROUTING is a valid synonym for the TRANSACTION_ROUTING keyword.

2 You must specify a routing rule if you activate routing.

### Parameters

**DEFN**=(**APP**,*name*)
Define an application called *name*. You can use any 8-character name for your applications.

**DESC**='*definition description*'
A description of the definition. A description that contains spaces must be enclosed in quotation marks. The description can be up to 30 characters long.

**REPLACE**=**NO** | **YES**
Determines what happens if you try to input a definition with the same type and name as one that already exists in the repository:

**YES** The definition replaces the existing definition.

**NO** All ADD commands directed at the repository fail.

**TRANSACTION_ROUTING**=**ACTIVE** | **INACTIVE**
Specifies whether transaction routing is active for this application. If active, then a message with a transaction code matching one in this application group **can** be routed. If inactive, then messages with a transaction code matching one in this application group will never be routed.

ROUTING is a valid synonym for the TRANSACTION_ROUTING keyword, but note that specifying both the ROUTING and TRANSACTION_ROUTING keywords will result in an error.

**ROUTING_RULE**=(*candidate_datastores*,*failure_option*)
This parameter consists of a routing rule and a failure option:

A transaction routing rule consists of a list of *candidate datastores*. IMS Connect Extensions will only route messages to those datastores that are candidate datastores for both the transaction code and original destination datastore of an incoming message. You can specify *candidate_datastores* as one of the following:

**ALL** All datastores

**DS**,*name*
A datastore called "*name*".

**DSG**,*name*
A datastore group called "*name*".

**AFL**,*name*
An affinity list called "*name*".

The transaction routing rule also consists of a *failure option*. This option specifies what to do when a message has no eligible candidate datastores (because the transaction and original destination datastore do not share any available candidates). Choose from either:

**FAILURE=ORIGINAL**
Use the destination ID set by the message (the original datastore) as the target datastore.

**FAILURE=REJECT**
Reject the message.

Here is an example routing rule:

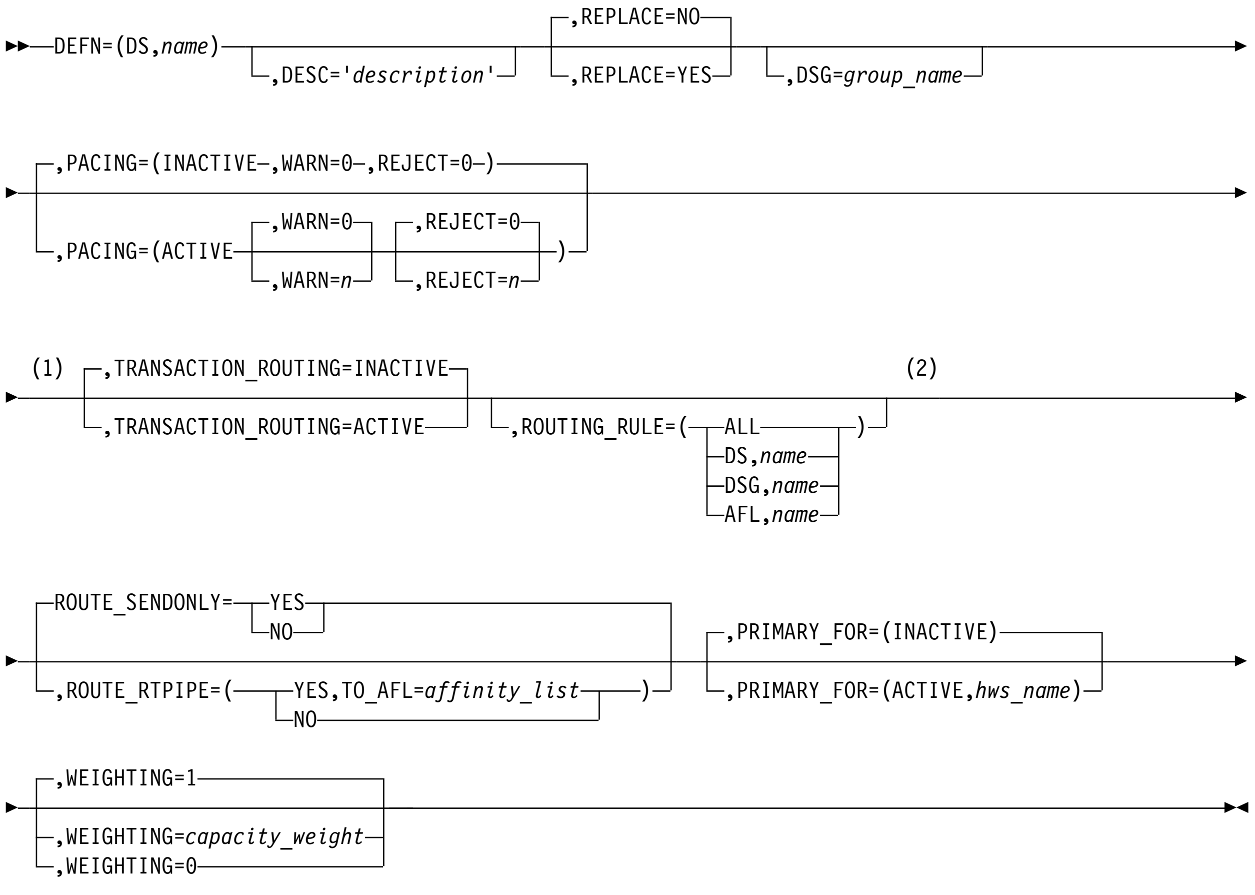
```
ROUTING_RULE=(AFL,MYATMDS,FAILURE=REJECT)
```

With this routing rule, an incoming message specifying this datastore as its DESTID can potentially be routed to one of the datastores in the affinity list. If the routing rule for the transaction code on the incoming message does not specify at least one candidate datastore from the MYATMDS affinity list that is available, then, based on the failure option, the message will be rejected by IMS Connect Extensions.

## Datastore (DS) ADD command

The DEFN DS ADD command adds a datastore to a repository.

## Format



**Notes:**

1 ROUTING is a valid synonym for the TRANSACTION_ROUTING keyword.

2 You must specify ROUTING_RULE if you activate transaction routing.

## Parameters

**DEFN**=(**DS**,*name*)
Define a datastore called *name*.

**DESC**='*definition description*'
A description of the definition. A description that contains spaces must be enclosed in quotation marks. The description can be up to 30 characters long.

**REPLACE**=<u>**NO**</u> | **YES**
Determines what happens if you try to input a definition with the same type and name as one that already exists in the repository:

**YES** The definition replaces the existing definition.

**NO** All ADD commands directed at the repository fail.

**DSG**=*group_name*
(Optional.) The name of the datastore group for this datastore. IMS Connect

Extensions aggregates statistics for datastore groups, allows you to set pacing thresholds for the entire group, and use the datastore group name as the destination on incoming messages (see the PRIMARY_FOR parameter).

**PACING**=(**ACTIVE** | **INACTIVE**,**WARN**=*n*,**REJECT**=*n*)
Pacing allows you to generate warnings or reject messages when an IMS Connect system tries to send messages to this datastore too quickly. The default for warning and reject thresholds is 0. A value of 0 means that the threshold for the datastore is not used by IMS Connect Extensions. Otherwise, the unit for the warning and reject thresholds is messages per 20-second interval .

**TRANSACTION_ROUTING**=**ACTIVE** | **INACTIVE**
Specifies whether transaction routing is active for this datastore. If active, then messages with a destination ID that matches the name of this datastore **can** be routed. If inactive, then messages with this datastore as the destination will never be routed. That is, IMS Connect Extensions will not alter their destination.

ROUTING is a valid synonym for the TRANSACTION_ROUTING keyword, but note that specifying both ROUTING and TRANSACTION_ROUTING will result in an error

**ROUTING_RULE**=(*candidate_datastores*)
This parameter specifies a transaction routing rule:

A transaction routing rule consists of a list of *candidate datastores*. IMS Connect Extensions will only route messages to those datastores that are candidate datastores for both the transaction code and original destination datastore of an incoming message. You can specify *candidate_datastores* as one of the following:

**ALL** All datastores

**DS**,*name*
A datastore called "*name*".

**DSG**,*name*
A datastore group called "*name*".

**AFL**,*name*
An affinity list called "*name*".

**ROUTE_SENDONLY and ROUTE_RTPIPE**
Determines how messages of type Send Only and Resume TPIPE are routed. See "Routing Send Only messages with transaction routing" on page 162 for details.

**PRIMARY_FOR**=(**INACTIVE** | **ACTIVE**,*system_name*)
Select whether to activate primary datastore routing. If you activate primary routing, IMS Connect Extensions will route messages to this datastore if both of the following statements are true:

- The incoming message specifies the name of the datastore group for this datastore as its destination ID.
- The message is processed by the IMS Connect system with the *system name* you specify.

**WEIGHTING**=*capacity_weight*
Specifies the capacity weight for this datastore. When workload balancing and transaction routing are active and IMS Connect Extensions finds more than one candidate datastore to route to, the weighting determines the probability, relative to the weighting of other candidate datastores, that IMS Connect

Extensions will route messages to this datastore. The value can be a number in the range 1 - 100 or 0. A value of zero has a special meaning. It indicates that the datastore is not a candidate for routing.

For example, suppose the following datastores are candidates for routing:

```
DS1: WEIGHTING=4
DS2: WEIGHTING=8
DS3: WEIGHTING=0
```

In this case IMS Connect Extensions will be twice as likely to route to DS2, with a capacity weight of 8, than to DS1, with a capacity weight of 4. No transactions will be routed to DS3.

## Datastore group (DSG) ADD command

The DEFN DSG ADD command adds a datastore group to a repository. You can associate a datastore with one (and only one) datastore group. You must define datastore groups if you route messages to a primary datastore.

### Format

```
                                                   ┌─,REPLACE=NO──┐
►►──DEFN=(DSG,name)──┬───────────────────────┬─────┼──────────────┼──────────►
                     └─,DESC='description'─┘       └─,REPLACE=YES─┘

  ┌─,PACING=(INACTIVE─,WARN=0─,REJECT=0─)──────────────┐
►─┼────────────────────────────────────────────────────┼─────────────────────►◄
  │                    ┌─,WARN=0─┐  ┌─,REJECT=0─┐      │
  └─,PACING=(ACTIVE────┼─────────┼──┼───────────┼──)───┘
                       └─,WARN=n─┘  └─,REJECT=n─┘
```

### Parameters

**DEFN**=(**DSG**,*name*)
Define a datastore group called *name*.

**DESC**='*definition description*'
A description of the definition. A description that contains spaces must be enclosed in quotation marks. The description can be up to 30 characters long.

**REPLACE**=**NO** | **YES**
Determines what happens if you try to input a definition with the same type and name as one that already exists in the repository:

**YES** The definition replaces the existing definition.

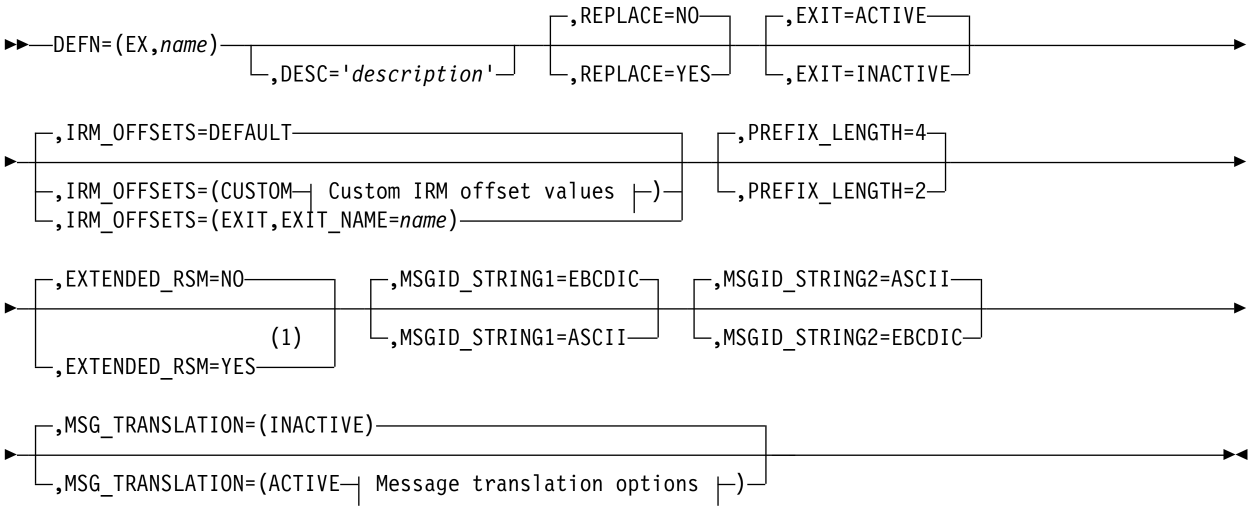
**NO** All ADD commands directed at the repository fail.

**PACING**=(**ACTIVE** | **INACTIVE**,**WARN**=*n*,**REJECT**=*n*)
Pacing allows you to generate warnings or reject messages when an IMS Connect system tries to send messages to this datastore group too quickly. . The default for warning and reject thresholds is "**0**". A "**0**" means that the threshold for this datastore group is not used by IMS Connect Extensions. IMS Connect Extensions will use the threshold values for the datastores in this group and for the IMS Connect system processing the message. The unit for the warning and reject thresholds is messages sent by an IMS Connect system per 20-second interval.

## User exit (EX) ADD command

The DEFN EX ADD command adds a user exit to a repository. You must define all exits for which you want to use IMS Connect Extensions functions.

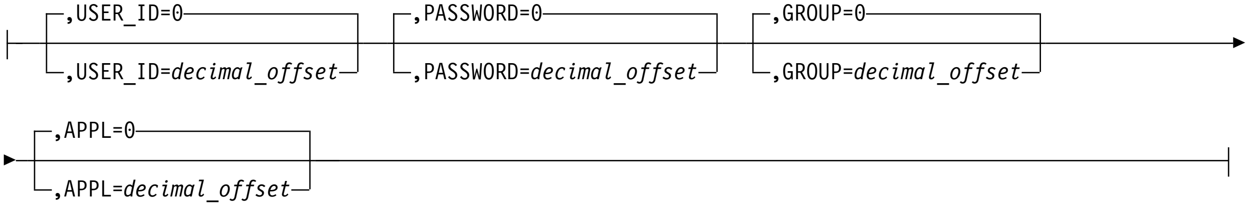
### Format



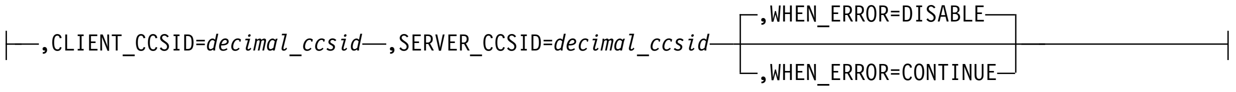
**Notes:**

1 You must specify a 4-byte length prefix if you use an extended RSM.

**Custom IRM offset values:**



**Message translation options:**



## Parameters

**DEFN**=(**EX**,*name*)

Define a user exit called *name*. The name must be the same as the name of the exit in the IMS Connect configuration member (HWSCFG) of an IMS Connect system defined in this repository.

**DESC**='*definition description*'

A description of the definition. A description that contains spaces must be enclosed in quotation marks. The description can be up to 30 characters long.

**REPLACE**=**NO** | **YES**

Determines what happens if you try to input a definition with the same type and name as one that already exists in the repository:

**YES** The definition replaces the existing definition.

**NO** All ADD commands directed at the repository fail.

**EXIT**=**ACTIVE** | **INACTIVE**

Specifies whether IMS Connect Extensions processing is active for the exit. If the exit is active, then IMS Connect Extensions can apply additional processing to messages handled by this exit. For example, route messages to alternate datastores or provide security processing.

**IRM_OFFSETS**=**DEFAULT** | (**CUSTOM**,*IRM_offset_values*) | (**EXIT**,**EXIT_NAME**=*name*)

(Optional) Specifies how to obtain the offsets within the user portion of the IRM header of the following fields: user ID, password or PassTicket, SAF group name, and optional application name. IMS Connect Extensions passes these fields to the external security manager to perform user ID verification.

**DEFAULT**

If you use the supplied sample exits unmodified or if you do not use IMS Connect Extensions security features, you do not need to specify this parameter.

**CUSTOM**

Use this parameter to specify custom offsets in the following cases:

- If you use your own message processing exits.
- If you use the supplied exits but have changed offsets for any of the fields that are required for security processing.

For example, to specify a user ID, password, group, and APPL immediately following the common IRM prefix:

```
IRM_OFFSETS=(CUSTOM,
             USER_ID=28,
             PASSWORD=36,
             GROUP=44,
             APPL=52)
```

Offsets are expressed in decimal, start at zero, and do not include the 4-byte IRM prefix. The offsets have to be separated by at least 8 bytes. Overlapping offsets are not allowed.

**EXIT**

Specify the name of a user exit that returns the security-related IRM fields. The security exit is called as each message is received and processed by IMS Connect Extensions. Use this parameter in the following cases:

- The security fields are not located at fixed offsets within the IRM. For example, if the field values are generated dynamically.

- The field values are not in a form your security manager can recognize. For example, if you use your own password encryption scheme.

To specify MYEXIT as the exit that provides security fields, for example, enter:

```
IRM_OFFSETS=(EXIT,EXIT_NAME=MYEXIT)
```

**PREFIX_LENGTH=4 | 2**
Specify whether the exit processes messages with a 4-byte or 2-byte length prefix.

**EXTENDED_RSM=NO | YES**
Specify whether IMS Connect Extensions can append enhanced information on some errors to the Response Status Message (RSM) that is returned to the remote client.

This option is only available to user exits that support the RSM protocol and build messages with a 4-byte length prefix.

**Note:** Activating this feature changes the length of the RSM. Some remote clients may have processing that depends on the RSM length.

**MSGID_STRING1=EBCDIC | ASCII,MSGID_STRING2=ASCII | EBCDIC**
Specify the code page, ASCII or EBCDIC, for each message identifier (MSGID) string set by the INIT routine for this exit. At the time of writing, the IBM-supplied exits require string1 to be set to EBCDIC and string2 to be set to ASCII.

**MSG_TRANSLATION=ACTIVE,... | INACTIVE**
Specifies whether the exit uses IMS Connect Extensions translation instead of the exit. If you activate translation you must specify a client and server CCSID.

**CLIENT_CCSID**=*decimal_ccsid*
The CCSID for clients attempting to connect to IMS.

**SERVER_CCSID**=*decimal_ccsid*
The CCSID for the host running IMS.

**WHEN_ERROR=CONTINUE | DISABLE**
This option determines what happens if IMS Connect Extensions encounters an error and cannot perform translation.

**CONTINUE**
The exit continues to process messages using its own translation routine.

**DISABLE**
The exit is disabled and will no longer accept messages for processing.

## System group (GRP) ADD command

The DEFN GRP ADD command adds a system group to a repository. A system group allows you to define a grouping of IMS Connect systems, typically reflecting your sysplex configuration. You can view summary statistics for all systems in a group and issue commands to the group..

### Format

```
                                                  ┌─,REPLACE=NO──┐
►►──DEFN=(GRP,name)──┬─────────────────────────┬──┼──────────────┼──────────────────►
                     └─,DESC='description'─────┘  └─,REPLACE=YES─┘

             ┌─,─────────────┐
             ▼               │
►─,HWSLIST=(───system_name───┴─)─────────────────────────────────────────────────────►◄
```

### Parameters

**DEFN**=(**GRP**,*name*)
: Define a system group called *name*. You can use any alphanumeric string of up to 8 characters. The group name is not case sensitive.

**DESC**='*definition description*'
: A description of the definition. A description that contains spaces must be enclosed in quotation marks. The description can be up to 30 characters long.

**REPLACE**=**NO** | **YES**
: Determines what happens if you try to input a definition with the same type and name as one that already exists in the repository:

**YES** The definition replaces the existing definition.

**NO** All ADD commands directed at the repository fail.

**HWSLIST**=(*system1*,*system2*,...)
: Specify a comma-delimited list of IMS Connect systems that are part of this system group. The same system can be part of more than one group.

## IMS Connect system (HWS) ADD command

The DEFN HWS ADD command adds an IMS Connect system to a repository.

## Format

```
                                                   ┌─,REPLACE=NO──┐
►►──DEFN=(HWS,name)──┬──────────────────────┬──────┼──────────────┼──,CEX_PORT=port_number──────────►
                     └─,DESC='description'──┘      └─,REPLACE=YES─┘

                                  ┌─,MSG_RECALL=800────┐ ┌─,CEX=ACTIVE───┐
►─,HOST=host_name_or_IP_addr──────┼────────────────────┼─┼───────────────┼──────────────────────────►
                                  └─,MSG_RECALL=0-800──┘ └─,CEX=INACTIVE─┘

  ┌─,MIXED_CASE_PASSWORDS=ACTIVE────┐ ┌─,EVENT_COL=(ACTIVE─,LEVEL=4─)─────────────────┐
►─┼─────────────────────────────────┼─┼───────────────────────────────────────────────┼─────────────►
  └─,MIXED_CASE_PASSWORDS=INACTIVE──┘ │                               ┌─,LEVEL=4──────┐   │
                                      └─,EVENT_COL=(──┬─ACTIVE───┬────┼───────────────┼─)─┘
                                                      └─INACTIVE─┘    └─,LEVEL=──┬─0─┬┘
                                                                                 ├─1─┤
                                                                                 ├─2─┤
                                                                                 └─3─┘

  ┌─,LOG_PFX=A0──┐ ┌─,PUBLISHER=(INACTIVE,MAX_CLIENTS=0)──┐
►─┼──────────────┼─┼──────────────────────────────────────┼─────────────────────────────────────────►
  └─,LOG_PFX=nn──┘ └─,PUBLISHER=(ACTIVE,MAX_CLIENTS=nn)───┘

  (1)
►─────,ACTIVE_DS=(─┤ Active data set configuration ├─)──────────────────────────────────────────────►

  (2)                                                     ┌─,ADVANCED=ACTIVE────┐
►─────,ARCHIVE_DS=(─┤ Archive data set configuration ├─)──┼─────────────────────┼──────────────────►
                                                          └─,ADVANCED=INACTIVE──┘

  ┌─,PACING=(INACTIVE─,COUNT=3─,WARN=0─,REJECT=0─)───────────────┐
►─┼──────────────────────────────────────────────────────────────┼─────────────────────────────────►
  │                  ┌─,COUNT=3─┐ ┌─,WARN=0─┐ ┌─,REJECT=0─┐      │
  └─,PACING=(ACTIVE──┼──────────┼─┼─────────┼─┼───────────┼─)────┘
                     └─,COUNT=n─┘ └─,WARN=n─┘ └─,REJECT=n─┘

  ┌─,SESSION_MESSAGE_LIMIT=INACTIVE───────────────────────────────┐
►─┼───────────────────────────────────────────────────────────────┼────────────────────────────────►
  │                                                ┌─100───┐      │
  └─,SESSION_MESSAGE_LIMIT=ACTIVE,THRESHOLD=(──────┴─value─┴──)───┘

►─┬───────────────────────────────────────────────┬─┬─────────────────────────────────────────────┬►
  └─,COMMANDS=(─┤ Commands configuration ├─)──────┘ └─,SECURITY=(─┤ Security configuration ├─)────┘

                                ┌─,STATS=ACTIVE────┐
►─┤ Routing configuration ├─────┼──────────────────┼──,HWS_CONFIG=dsname(member)───────────────────►◄
                                └─,STATS=INACTIVE──┘
```

**Notes:**

1 Active data set definition is not required if event collection is set to INACTIVE.

2 Archive data set definition is not required if the active data set definition sets archiving to INACTIVE.

**Active data set configuration:**

```
                 ┌─,NUM_ACTIVES=3─┐  ┌─,ARCHIVE=YES─┐  ┌─,WHEN_FULL=REUSE───┐
├──DSN_template──┼────────────────┼──┼──────────────┼──┼────────────────────┼──────────►
                 └─,NUM_ACTIVES=n─┘  └─,ARCHIVE=NO──┘  ├─,WHEN_FULL=WAIT────┤
                                                       ├─,WHEN_FULL=DISCARD─┤
                                                       └─,WHEN_FULL=END─────┘

►──┬──────────────────┬──┬──────────────────────┬──┬────────────────┬──┬────────────────┬──►
   └─,MAN_CLASS=class─┘  └─,STORAGE_CLASS=class─┘  └─,VOLSER=volser─┘  └─,DEVICE=device─┘

                          ┌─,SPACE=CYLS─┐  ┌─,PRIMARY_Q=200───┐  ┌─,OVERFLOW=NO──┐
►──┬───────────────────┬──┼─────────────┼──┼──────────────────┼──┼───────────────┼──►
   └─,DATA_CLASS=class─┘  └─,SPACE=TRKS─┘  └─,PRIMARY_Q=nnnnn─┘  └─,OVERFLOW=YES─┘

   ┌─,OVERFLOW_SIZE=4──┐
►──┼───────────────────┼──────────────────────────────────────────────────────┤
   └─,OVERFLOW_SIZE=nn─┘
```

**Archive data set configuration:**

```
                 ┌─,MAX_ACTIVE=3─┐                         ┌─,MAX_VOLUMES=1──┐
├──DSN_template──┼───────────────┼──┬───────────────────┬──┼─────────────────┼──►
                 └─,MAX_ACTIVE=n─┘  └─,MAX_TIME=seconds─┘  └─,MAX_VOLUMES=nn─┘

                        ┌─,JOB_CARD1='//CEXARCH JOB (ACCOUNT),''NAME'''─┐
►──┬─────────────────┬──┼───────────────────────────────────────────────┼──►
   └─,MAX_SIZE=nnnnn─┘  └─,JOB_CARD1='line1_string'─────────────────────┘

►──┬───────────────────────────┬──┬───────────────────────────┬──,ARCHIVE_JCL=dsname(member)──►
   └─,JOB_CARD2='line2_string'─┘  └─,JOB_CARD3='line3_string'─┘

   ┌─,ARCHIVE_CLEANUP=7───────────┐
►──┼──────────────────────────────┼──────────────────────────────────────────┤
   └─,ARCHIVE_CLEANUP=num_of_days─┘
```

**Commands configuration:**

```
   ┌─INACTIVE─┐   ┌─,CONTROL_ACCESS=NO──────────────────────────────────────────────────────────────┐
├──┼──────────┼───┼─────────────────────────────────────────────────────────────────────────────────┼──┤
   └─ACTIVE───┘   └─,CONTROL_ACCESS=(──┬─NO──┬──,SECURITY_APPLID=applid,GEN_PASSTICKET=──┬─NO──┬──)─┘
                                       └─YES─┘                                          └─YES─┘
```

**Security configuration:**

```
   ┌─INACTIVE─┐   ┌─,ACEE_CACHE=INACTIVE─┐   ┌─,AGE_INTERVAL=60────┐
├──┼──────────┼───┼──────────────────────┼───┼─────────────────────┼──────────────────────────────►
   └─ACTIVE───┘   └─,ACEE_CACHE=ACTIVE───┘   └─,AGE_INTERVAL=nnnn──┘

         (1)
  ┌─────,ACT_VAL=INACTIVE────────────────────────────────────────────────────────┐
►─┼──────────────────────────────────────────────────────────────────────────────┼──────────────┤
  └─,ACT_VAL=ACTIVE,SECURITY_CLASS=SAF_class,VAL_TYPE=(─┬─IMSCONNECT───────────────────┬─)─┘
                                                         ├─IMSCONNECT_IPADDRESS_PORT─┤
                                                         │   (2)                       │
                                                         ├─1───────────────────────────┤
                                                         └─2───────────────────────────┘
```

**Notes:**

1 VALIDATION is a valid synonym for the ACT_VAL keyword.

2 1 is a valid synonym for the IMSCONNECT keyword. 2 is a valid synonym for the IMSCONNECT_IPADDRESS_PORT keyword.

**Routing configuration:**

```
      (1)
   ┌──────,OTMA-RULES_ROUTING=INACTIVE─┐   ┌─,ODBM-RULES_ROUTING=INACTIVE─┐
├──┼───────────────────────────────────┼───┼──────────────────────────────┼────────────────────►
   └─,OTMA-RULES_ROUTING=ACTIVE────────┘   └─,ODBM-RULES_ROUTING=ACTIVE───┘

     (2)
  ┌──────,TRANSACTION_ROUTING=INACTIVE──────────────────┐   ┌─,BALANCING=INACTIVE─┐
►─┼─────────────────────────────────────────────────────┼───┼─────────────────────┼─────────┤
  └─,TRANSACTION_ROUTING=ACTIVE,APPLIST=(─application─)─┘   └─,BALANCING=ACTIVE───┘
```

**Notes:**

1 RULES-BASED_ROUTING is a valid synonym for the OTMA-RULES_ROUTING keyword.

2 ROUTING is a valid synonym for the TRANSACTION_ROUTING keyword.

## Required parameters

These are the parameters that you must specify:

**DEFN**=(**HWS**,*name*)

Define a system called *name*. The *name* must have the same value as that of the ID parameter of the HWS configuration statement. This statement is in the IMS Connect configuration member (HWSCFG) for this IMS Connect system.

**CEX_PORT**=*port_number*

Specifies the IMS Connect Extensions console listener port. This TCP/IP port is used by IMS Connect Extensions clients to connect to IMS Connect Extensions enabled IMS Connect systems and provide operational control and monitoring for them. This port should not be confused with the ports IMS Connect uses to process messages. Assign an unallocated TCP/IP port using decimal port values (1 - 65535).

**HOST**=*hws_host_name*
Specifies the host name or IP address of the IMS Connect Extensions enabled IMS Connect system you are trying to connect to. Use LOCALHOST if you are running the IMS Connect Extensions ISPF dialog on the same machine as the IMS Connect system you are trying to connect to.

**ACTIVE_DS**=(*journal_name_template*)
Specifies the data set name prefix and other attributes for active journal data sets. See "Active data set configuration" on page 287.

**ARCHIVE_DS**=(*journal_name_template*)
Specifies the data set name prefix and other attributes for archive journal data sets. See "Archive data set configuration" on page 288.

## Optional parameters

**DESC**='*definition description*'
A description of the definition. A description that contains spaces must be enclosed in quotation marks. The description can be up to 30 characters long.

**REPLACE**=**NO** | **YES**
Determines what happens if you try to input a definition with the same type and name as one that already exists in the repository:

**YES** The definition replaces the existing definition.

**NO** All ADD commands directed at the repository fail.

**CEX**=**ACTIVE** | **INACTIVE**
Specify whether to activate or deactivate IMS Connect Extensions features for this IMS Connect system.

**MIXED_CASE_PASSWORDS**=**ACTIVE** | **INACTIVE**
Specify how IMS Connect Extensions is to handle passwords. When this option is active, passwords are treated as mixed case and no case translation is performed. When this option is inactive, passwords are converted to uppercase before any security processing.

**Note:** The SECURITY PWCASE control option determines how mixed-case passwords are handled. You can use this to specify that passwords are to be translated to uppercase, or left as mixed case, or else the setting from the external security manager is to be used.

If PWCASE=SYSTEMDEFINITION, the setting from the Activate Mixed Case Passwords option in the system definition is used. If PWCASE is set to anything other than SYSTEMDEFINITION, the Activate Mixed Case Passwords option in the system definition is ignored.

**MSG_RECALL**= *10* **-** *800*
Specifies the maximum number of IMS Connect and IMS Connect Extensions messages that can appear in the IMS Connect Extensions online message log.

**EVENT_COL**=**ACTIVE** | **INACTIVE,LEVEL**=*0* **-** *4*
Specifies the event collection level. A higher number means IMS Connect Extensions records more IMS Connect events. The default is 4, the highest collection level.

**LOG_PFX**= *A0* **-** *FF*

Specifies the log prefix IMS Connect Extensions uses for the events it collects in its journals. The default prefix is A0. Use the default value unless you are already generating custom IMS log records with the A0 prefix: some IBM IMS tools, such as IMS Problem Investigator, will automatically recognize A0-prefixed log records and analyze the IMS Connect data in them. Such tools would have to be configured to recognize IMS Connect event data recorded with any other prefix.

**PUBLISHER**=**INACTIVE** **|** **ACTIVE,MAX_CLIENTS**=*0* **-** *99*

Specifies whether to activate the publisher API. Publisher API clients are typically monitoring applications. If you are using OMEGAMON for IMS, you need to activate this feature and set the maximum clients (MAX_CLIENTS) to a number equal to or greater than 1.

**ADVANCED**=**ACTIVE** **|** **INACTIVE**

Specify whether to activate advanced features. It is recommended that you always leave this option active and deactivate features you do not wish to use individually.

**PACING**=(*pacing parameters*)

Specifying pacing parameters allows you to respond to surges in message activity. The parameters are:

**ACTIVE** **|** **INACTIVE**

Activate pacing for this IMS Connect system. If pacing is inactive, then messages processed by this IMS Connect system will not trigger warnings or get rejected.

**COUNT**=*2* **-** *9* **|** **3**

Specifies the interval count. The interval count is the number of consecutive 20-second intervals for which a warning or reject threshold must be reached. See the WARN and REJECT parameters for details.

**WARN**=*n* **|** **0**

Specify the warning threshold in messages per 20-second interval. If the warning threshold is exceeded for *n* consecutive intervals (where *n* is the interval count), then IMS Connect Extensions writes a warning to the system log. Specify `0` if you do not want to use a warning threshold for this system. You can still specify a warning threshold for datastores or datastore groups.

**REJECT**=*n* **|** **0**

Specify the reject threshold in messages per 20-second interval. If the reject threshold is exceeded for *n* consecutive intervals (where *n* is the interval count), then this IMS Connect system will reject all messages for a 20 second period. Specify `0` if you do not want to use a reject threshold for this system. You can still specify a reject threshold for datastores or datastore groups.

Here is an example pacing configuration:

```
PACING=(ACTIVE,WARN=500)
```

These settings will write a warning message to the system log if this IMS Connect system processed more than 500 messages in each of three consecutive 20-second intervals.

**SESSION_MESSAGE_LIMIT**=**INACTIVE** **|** **ACTIVE,THRESHOLD**=*1* **-** *999999*

Specify whether the Session Message Limit feature is active. If it is active, IMS

Connect Extensions will close a persistent session when it has received the number of input messages specified in THRESHOLD. The default value for THRESHOLD is 100.

**STATS=ACTIVE | INACTIVE**
Specify whether to activate statistics collection. Statistics collection is required for most IMS Connect Extensions features.

**HWS_CONFIG**=*dsname(member)*
The HWS_CONFIG parameter specifies the data set and member name that contains the IMS Connect configuration member.

## Active data set configuration

These are all the parameters that you can specify for the active journal data sets.

**ACTIVE_DS**=(*active data set parameters*)

***DSN_template***
Specifies the data set name prefix for active journal data sets. The active journal is where IMS Connect Extensions stores the IMS Connect related events it collects. The active journal consists of a series of permanent data sets that IMS Connect Extensions uses in rotation. The data sets are allocated by IMS Connect Extensions, with the suffix **P***nn*, where *nn* represents the active journal number, 01 - 32. For example, if you specify `ACTIVE_DS(MY.ACTIVE.JOURNAL)`, IMS Connect Extensions creates these data sets:

```
'MY.ACTIVE.JOURNAL.P01'
'MY.ACTIVE.JOURNAL.P02'
'MY.ACTIVE.JOURNAL.P03'
```

You can use the **&ID.** symbol in the data set name. IMS Connect Extensions replaces the symbol with the name of the IMS Connect system for which the journal is recorded.

**NUM_ACTIVES**=*1 - 32* | *3*
Specifies the number of active journal data sets IMS Connect Extensions creates before it attempts to reuse the oldest one.

**ARCHIVE=YES | NO**
Specify if you want the Archive Manager to copy the Active Journals to the Archive Journal data sets.

**WHEN_FULL**=*option_for_unarchived_datasets*
Specifies what to do when the active journal data sets and the overflow are full and there are unarchived event records. Normally, IMS Connect Extensions reuses active data sets as needed. However, if you switch archiving off or if archiving fails then it will use the journal full option to determine what to do. The options are:

**REUSE**
Reuse the data sets even though they have not been archived.

**WAIT**
Queue incoming messages and make clients wait for IMS Connect Extensions to complete archiving of an active journal data so that it can reuse it.

**DISCARD**
Discard event records until an active journal data set is archived and then reuse that data set.

**END**

End archiving and the event collection processing. IMS Connect Extensions will no longer collect events for the system until you next restart that system.

**MAN_CLASS**=*management_class*

The system-managed storage (SMS) management class for the data sets. Omit this parameter to use the default class for your site.

**STORAGE_CLASS**=*storage_class*

The SMS storage class for the data sets. Omit this parameter to use the default storage class for your site.

**VOLSER**=*volume_serial*

The volume serial number of the direct access storage device for the data sets. Omit this parameter to use the default volume serial.

**DEVICE**=*device_type*

The device type, such as SYSDA, for the DASD volume of the active data sets.

**DATA_CLASS**=*data_class*

The data class for the data sets. Omit this parameter to use the default class for your site.

**SPACE**=**CYLS** | **TRKS**

Specify the unit of the primary quantity (PRIMARY_Q parameter) as either:

**CYLS**

Cylinders

**TRKS**

Tracks

**PRIMARY_Q**=*1 - 99999* | *200*

This field allows you to specify the primary allocation quantity. The quantity is in a unit specified by the SPACE parameter.

**OVERFLOW**=**NO** | **YES**

Specify if you want to use an overflow data set. An overflow data set is an additional active journal data set that IMS Connect Extensions creates if all active journal data sets are full and have not yet been archived.

**OVERFLOW_SIZE**=*1 - 99* | *4*

Specifies the size of the overflow data set. This is a multiple of the primary quantity, in the unit of the primary quantity.

## Archive data set configuration

These are all the parameters that you can specify for the archive journal data sets.

**ARCHIVE_DS**=(*archive data set parameters*)

***DSN_template***

Specify the archive journal data set name. IMS Connect Extensions stores the contents of the active journals in these archives. To generate a unique data set name, enter either a generation data group (GDG) data set or a name containing symbols.

- For a GDG data set, use the name of an existing GDG base and "(+1)" to indicate a new generation. For example:

  ```
  JOURNAL.GDGBASE(+1)
  ```

- For a name containing symbols such as date and time stamps or the IMS Connect system ID, see "Naming archive journal data sets" on page 62.

**ARCHIVE_JCL**=*dsname(member)*

Specify the member name that contains the skeleton JCL to execute the Archive Manager. The name must be entered as a fully qualified data set and member name, all enclosed in quotation marks. For more information, see “Creating an archive JCL skeleton” on page 63.

**MAX_ACTIVE**=*2* **-** *32* **|** **3**

The maximum number of active journal data sets that IMS Connect Extensions will write to a single archive journal data set.

**MAX_TIME**=*1* **-** *120*

The maximum time, in seconds, that IMS Connect Extensions waits before checking to see if more active journals become full and so can be archived.

**MAX_SIZE**=*0* **-** *32767*

The maximum size, in megabytes (MB), of an Archive data set. When the size is reached, the archive data set will be closed and a new archive data set will be written.

If the archive data set reaches the maximum size, then IMS Connect Extensions writes a new archive data set. A zero means that IMS Connect Extensions does not use the value of this parameter to determine when to create a new archive.

**MAX_VOLUMES**=*1* **-** *99*

The number of tape volumes that an archive data set can span. This value is ignored if archive data sets are written to DASD. If the archive data set reaches the maximum number of volumes, IMS Connect Extensions will write to another archive data set.

**ARCHIVE_CLEANUP**=*1* **-** *365* **|** **7**

The number of days before an archive data set can be deleted or uncataloged by the archive cleanup utility and its name deleted from the repository. See “Archive Journal cleanup utility” on page 335).

*JOB_CARD1=job card line* **|** *'//CEXARCH JOB (ACCOUNT),''NAME"'*

Specifies what appears on the first line of the archive job IMS Connect Extensions submits.

*JOB_CARD2=job card line*

Specifies what appears on the second line of the archive job IMS Connect Extensions submits.

*JOB_CARD3=job card line*

Specifies what appears on the third line of the archive job IMS Connect Extensions submits.

## Commands configuration

The commands feature allows you to send IMS Connect WTOR commands and IMS Connect z/OS commands, IMS Type-1 commands, and IMS Connect Extensions commands using the command shell.

**COMMANDS**=(*command parameters*)

The parameters are:

**ACTIVE** **|** **INACTIVE**

Select whether to activate the commands feature.

**CONTROL_ACCESS**=(**YES** **|** **NO**,*options*)

Select whether to restrict access to commands by using RACF or an equivalent external security manager to authenticate users. The parameters for access restrictions are:

**SECURITY_APPLID**=*applid*

The APPLID for RACF (or equivalent). This APPLID is also used for PassTicket generation. See Chapter 12, "Configuring command access in IMS Connect Extensions," on page 129 for details.

**GEN_PASSTICKET=YES | NO**

Specify whether to use PassTickets for security authentication. Use PassTickets if you do not want user passwords to be transmitted.

## Security configuration

The security features allow you to use IMS Connect Extensions for message security validation, instead of IMS Connect.

**SECURITY**=(*security parameters*)

IMS Connect Extensions can cache ACEEs returned by RACF (or an equivalent security manager) as well as authenticate access to IMS Connect systems. The parameters are:

**ACTIVE | INACTIVE**

Select whether to activate IMS Connect Extensions security.

If you activate IMS Connect Extensions security, then you should disable IMS Connect security by specifying RACF=N in the configuration member.

**ACEE_CACHE=INACTIVE | ACTIVE**

Specify whether IMS Connect Extensions caches the ACEEs. Activating ACEE caching might significantly reduce the number of security calls required to revalidate user IDs.

**AGE_INTERVAL**=*0-1440* **| 60**

Specifies the time, in minutes, for the frequency in which IMS Connect Extensions clears the ACEE cache. The more frequently you clear the cache, the lower the storage required for ACEEs but this will also mean less overall performance benefits. Use a high value to avoid revalidating user IDs. Specify a "0" to never clear the cache.

**ACT_VAL=INACTIVE | ACTIVE**

Specify whether you want to perform additional security validation; that is, checking if the user is authorized to use the IMS Connect system through which the message or DRDA request is routed. If you activate validation you must also specify a security class.

VALIDATION is a valid synonym for the ACT_VAL keyword.

**SECURITY_CLASS**=*SAF_resource_class*

Specify the RACF (or equivalent) resource class for IMS Connect system validation.

**VAL_TYPE=IMSCONNECT | IMSCONNECT_IPADDRESS_PORT | 1 | 2**

Specify the type of validation security to be performed.

**IMSCONNECT | 1**

Perform validation based on the IMS Connect name. 1 is allowed as an alias.

**IMSCONNECT_IPADDRESS_PORT | 2**

Perform validation based on the IMS Connect name and the client IP address and IMS Connect port. 2 is allowed as an alias.

**Notes:**

1. If SECURITY and ACT_VAL are both activated, VAL_TYPE must be specified.

2. If SECURITY is activated, a valid user ID and password must be provided in each incoming message.

### Routing configuration

These fields specify which routing features are active on this system.

**OTMA-RULES_ROUTING=INACTIVE | ACTIVE**
Specifies whether to activate rules-based routing for IRM messages processed by this system.

RULES-BASED_ROUTING is a valid synonym for the OTMA-RULES_ROUTING keyword, but note that specifying both RULES-BASED_ROUTING and OTMA-RULES_ROUTING will result in an error.

**ODBM-RULES_ROUTING=INACTIVE | ACTIVE**
Specifies whether to activate rules-based routing for DRDA requests processed by this system.

**TRANSACTION_ROUTING=INACTIVE | ACTIVE**
Specifies whether to activate transaction routing for IRM messages processed by this system. If you activate transaction routing you must specify at least one application.

ROUTING is a valid synonym for the TRANSACTION_ROUTING keyword, but note that specifying both ROUTING and TRANSACTION_ROUTING will result in an error

**APPLIST=***list_of_applications*
Specifies a comma-delimited list of applications. Applications associate the IMS Connect system with the transaction codes it can route, and are used for transaction routing and other transaction level features.

*BALANCING=INACTIVE | ACTIVE*
When IMS Connect Extensions finds more than one eligible datastore to route an IRM message to, it can use capacity weights to determine how to select a datastore. Similarly, when IMS Connect Extensions finds more than one eligible ODBM target to route a DRDA request to, it can use capacity weights to determine how to select a target. This option specifies whether these weights are honored or, if inactive, whether equal weights always apply.

## ODBM routing list (ORL) ADD command

The DEFN ORL ADD command adds an ODBM routing list to a repository.

### Format

```
                                                        ┌─,REPLACE=NO──┐
►►──DEFN=(ORL,name)──┬──────────────────────┬──────────┼──────────────┼──────────────►
                     └─,DESC='description'─┘           └─,REPLACE=YES─┘

                  ┌─,─────┐
                  ▼       │
►─,TARGET_LIST=(───target─┴─)─────────────────────────────────────────────────────►◄
```

### Parameters

**DEFN**=(**ORL**,*name*)
Define an routing list called *name*. You can use any alphanumeric string of up to 8 characters. The name is not case sensitive.

**DESC**='*definition description*'
A description of the definition. A description that contains spaces must be enclosed in quotation marks. The description can be up to 30 characters long.

**REPLACE**=**NO** | **YES**
Determines what happens if you try to input a definition with the same type and name as one that already exists in the repository:

**YES** The definition replaces the existing definition.

**NO** All ADD commands directed at the repository fail.

**TARGET_LIST**=(*target1,target2,...*)
A comma-delimited list of ODBM targets that are part of this routing list. The same ODBM target can be part of more than one routing list.

## ODBM routing rule (ORR) ADD command

The DEFN ORR ADD command adds an ODBM routing rule to a repository.

### Purpose

ODBM routing rules enable you to specify routing behavior for DRDA requests. They convert an input alias into a target selected from a collection of ODBM targets that you specify. A routing rule can be assigned along with other related rules to a routing plan.

### Format

```
                                                 ┌─,REPLACE=NO──┐
►►──DEFN=(ORR,name)──┬──────────────────────┬──┼──────────────┼──────────────►
                     └─,DESC='description'─┘  └─,REPLACE=YES─┘

►──┬──────────────────────────────────────────┬──┬──────────────────────────┬──►
   └─,APPLY_RULE_TO=─┬─(ALL)───────────┬─┘  └─,PLAN_NAME=routing_plan─┘
                     ├─(SYSTEM,system)─┤
                     └─(GROUP,group)───┘

►──┬──────────────────────────────────────────────────────────────────────┬──►
   └─,CONDITION=(─┬───────────────────┬─┬────────────────────────────┬─)─┘
                  └─INPUT_ALIAS=alias─┘ └─,PSBNAME_LIST=PSB_namelist─┘

►──┬────────────────────────────────────────┬───────────────────────────────►◄
   └─,DRDA_REQUEST=(─┤ Routing options ├─)─┘
```

**Routing options:**

```
   ┌─INHERIT─┐
├──┼─ENABLE──┼──┬───────────────┬──┬────────────────────┬────────────────────────┤
   └─DISABLE─┘  └─,TARGET=target─┘  └─,FALLBACK=fallback─┘
```

## Parameters

**DEFN**=(**ORR**,*name*)
Define a routing rule called *name*. You can use any alphanumeric string of up to 8 characters. The name is not case sensitive.

**DESC**='*definition description*'
A description of the definition. A description that contains spaces must be enclosed in quotation marks. The description can be up to 30 characters long.

**REPLACE**=**NO** | **YES**
Determines what happens if you try to input a definition with the same type and name as one that already exists in the repository:

**YES** The definition replaces the existing definition.

**NO** All ADD commands directed at the repository fail.

**APPLY_RULE_TO**=(*scope*)
The scope of the rule:

**ALL** All systems

**SYSTEM**,*name*
A system called "*name*".

**GROUP**,*name*
A system group called "*name*".

**PLAN_NAME**=*routing_plan*
Assigns this routing rule to a routing plan.

A routing rule that is not assigned to any routing plan is always in effect. Where there is a conflict between an unassigned routing rule and a routing rule that is explicitly assigned to the routing plan that is currently active, the rule that is assigned to the current routing plan takes precedence. In other words, you can define unconditional rules to specify generic routing behavior and then override that behavior using specific rules that are assigned to the current routing plan.

**CONDITION**=(**INPUT_ALIAS**=*alias*,**PSBNAME_LIST**=*PSB_namelist*)
The Input Alias specifies the alias name provided by the client in the DRDA request.

When a PSB name list is specified, a routing rule is built for the alias name and each member of the PSB name list. If the PSB name list is blank, the routing rule applies to all PSB names.

The values for both the alias and PSB name relate to data in objects included in the DRDA request "ACCRDB-Access RDB" (code point 2001).

**DRDA_REQUEST**=(*routing options*)
Specifies whether the routing rule is enabled at the level specified in the APPLY_RULE_TO parameter. The options are described in "Routing options" on page 294.

## Routing options

The following parameters can be specified for the DRDA_REQUEST request type:

**ENABLE**
: This rule is enabled at the level specified in the APPLY_RULE_TO parameter. You must specify a TARGET routing list. The FALLBACK parameter is optional.

**DISABLE**
: This rule will not apply at the level specified in the APPLY_RULE_TO parameter. The TARGET and FALLBACK parameters are ignored.

**INHERIT**
: Inherit settings from rules applying to the same condition (INPUT_ALIAS and PSBNAME_LIST) at either the All systems or Group level. A Group level rule will inherit settings from the All systems level. A system level rule will inherit settings from the Group level, if one is defined, otherwise from the All systems level. The TARGET and FALLBACK parameters are ignored.

**TARGET**
: Specifies a routing list of ODBM targets to use as candidates for routing DRDA requests.

**FALLBACK**
: Specifies a routing list of ODBMs to use as candidates if the none of the ODBMs in the TARGET routing list are available.

# ODBM target (ORT) ADD command

The DEFN ORT ADD command adds an ODBM target to a repository. An ODBM target defines the IMS ODBMs and their associated aliases that are controlled by IMS Connect. ODBM targets become the building blocks for ODBM routing lists..

## Format

```
                                                   ┌─,REPLACE=NO──┐
►►──DEFN=(ORT,name)──┬───────────────────────┬─────┼──────────────┼──────────────────►
                     └─,DESC='description'───┘     └─,REPLACE=YES─┘

►──,TARGET_ODBM_NAME=(ODBM_name)──┬────────────────────────────────┬──────────────────►
                                  └─,TARGET_IMS_ALIAS=(IMS_alias)──┘

   ┌─,WEIGHTING=1──────────────────┐
►──┼───────────────────────────────┼────────────────────────────────────────────────►◄
   ├─,WEIGHTING=capacity_weight────┤
   └─,WEIGHTING=0──────────────────┘
```

## Parameters

**DEFN=(ORT,***name***)**
: Define an ODBM target called *name*. You can use any alphanumeric string of up to 8 characters. The name is not case sensitive.

**DESC**='*definition description*'

A description of the definition. A description that contains spaces must be enclosed in quotation marks. The description can be up to 30 characters long.

**REPLACE**=NO | **YES**

Determines what happens if you try to input a definition with the same type and name as one that already exists in the repository:

**YES** The definition replaces the existing definition.

**NO** All ADD commands directed at the repository fail.

**TARGET_ODBM_NAME**=(*ODBM_name*)

The name of the Open Database Manager.

**TARGET_IMS_ALIAS**=(*IMS_alias*)

The name of the IMS alias associated with the ODBM.

When the routing routine returns a target candidate and the alias name is blank, CEXROUT0 will attempt to select a target using the alias names associated with the target ODBM name. In this case, workload balancing is not used and all alias members are considered equal in weight for the request. This results in a form of round-robin processing.

**WEIGHTING**=*capacity_weight*

Specifies the capacity weight for this ODBM target. When workload balancing and ODBM routing are active and IMS Connect Extensions finds more than one candidate ODBM target to route to, the weighting determines the probability, relative to the weighting of other candidate ODBM targets, that IMS Connect Extensions will route messages to this ODBM target. The value can be a number in the range 1 - 100 or 0. A value of zero has a special meaning. It indicates that the target is not a candidate for routing.

For example, suppose the following ODBM targets are candidates for routing:

```
OD1: WEIGHTING=4
OD2: WEIGHTING=8
OD3: WEIGHTING=0
```

In this case IMS Connect Extensions will be twice as likely to route to `OD2`, with a capacity weight of 8, than to `OD1`, with a capacity weight of 4. No transactions will be routed to OD3.

## PSB name list (PSB) ADD command

The DEFN PSB ADD command adds a PSB name list to a repository. A PSB name list contains PSB names and is used in the **PSB name list** field of the ODBM Routing Rule definition panel.

When a PSB name list is used, a routing rule is built for the Alias name and each member of the PSB name list.

## Format

```
                                                      ┌─,REPLACE=NO──┐
►►──DEFN=(PSB,name)──┬─────────────────────┬──────────┼──────────────┼──────────►
                     └─,DESC='description'─┘          └─,REPLACE=YES─┘

   ┌─,──────────────────────────────────────────────┐
   ▼                                                │
►──┴─,PSBNAME=(─PSB_name──┬───────────────────────┬─)─┴───────────────────────►◄
                          └─,"PSB_description"────┘
```

## Parameters

**DEFN=(PSB,***name***)**
Define a PSB name list called *name*. You can use any alphanumeric string of up to 8 characters. The name is not case sensitive.

**DESC=**'*definition description*'
A description of the definition. A description that contains spaces must be enclosed in quotation marks. The description can be up to 30 characters long.

**REPLACE=NO | YES**
Determines what happens if you try to input a definition with the same type and name as one that already exists in the repository:

**YES** The definition replaces the existing definition.

**NO** All ADD commands directed at the repository fail.

**PSBNAME=(***name***,"***description***")**
The name and an optional description of a PSB belonging to this PSB name list.

Can be an actual PSB name or a generic name that includes a single trailing wildcard ("*"). Embedded wildcards are not allowed in generic PSB names.

When an ODBM routing rule uses a PSB List, a rule is generated for each PSB name in the list. IMS Connect Extensions will try to match the full PSB name on the incoming DRDA request with the rule. If it is unsuccessful, the granularity of the match is reduced one character at a time until a match is found. If no match is found then the incoming request cannot be routed.

# Routing plan (PLN) ADD command

The DEFN PLN ADD command adds a routing plan to a repository. Routing rules can be assigned to a routing plan so that they can subsequently be activated as a set. The association between a routing rule and a routing plan is made in the OTMA and ODBM routing rule definition.

### Format

```
                                              ┌─,REPLACE=NO──┐
►►──DEFN=(PLN,name)──┬──────────────────────┬─┼──────────────┼──────────────────────────────────►◄
                     └─,DESC='description'──┘ └─,REPLACE=YES─┘
```

### Parameters

**DEFN**=(**PLN**,*name*)
Define a routing plan called *name*.

**DESC**='*definition description*'
A description of the definition. A description that contains spaces must be enclosed in quotation marks. The description can be up to 30 characters long.

**REPLACE**=**NO** | **YES**
Determines what happens if you try to input a definition with the same type and name as one that already exists in the repository:

**YES** The definition replaces the existing definition.

**NO** All ADD commands directed at the repository fail.

## OTMA routing rule (RTG) ADD command

The DEFN RTG ADD command adds an OTMA routing rule to a repository.

### Purpose

A routing rule allows you to specify rules-based routing behavior for IRM messages. It describes which datastores to use as primary targets and fallback targets for a given original datastore identifier (DestID). A routing rule can be assigned along with other related rules to a routing plan.

You can have one master rule for a given DestID, as well as optional rules that specify different target datastores for a specified list of transaction codes.

## Format

```
                                       ┌─,REPLACE=NO──┐
►►──DEFN=(RTG,name)──┬──────────────────────┬──┼──────────────┼──────────────────────────────────►
                     └─,DESC='description'──┘  └─,REPLACE=YES─┘

►──┬───────────────────────────────────────┬──┬──────────────────────────┬──────────────────────►
   └─,APPLY_RULE_TO=──┬─(ALL)─────────┬────┘  └─,PLAN_NAME=routing_plan──┘
                      ├─(SYSTEM,system)─┤
                      └─(GROUP,group)───┘

►──┬──────────────────────────────────────────────────────────────────────────────────┬──────────►
   │                             ┌─,QUALIFIER_TYPE=NONE──────────────────────────────┐ │
   └─,CONDITION=(DESTID=destid───┼───────────────────────────────────────────────────┼─)─┘
                                 └─,QUALIFIER_TYPE=TXL,QUALIFIER_LIST_NAME=quallist──┘

►──┬─────────────────────────────────────────┬──┬─────────────────────────────────────┬─────────►
   └─,SEND_RECEIVE=(─┤ Routing options ├─)───┘  └─,SEND_ONLY=(─┤ Routing options ├─)──┘

►──┬───────────────────────────────────┬──┬──────────────────────────────────────┬──────────────►
   └─,RTPIPE=(─┤ Routing options ├─)───┘  └─,SYNC_REQUEST=(─┤ Routing options ├─)─┘

►──┬──────────────────────────────────────────┬──────────────────────────────────────────────►◄
   └─,SYNC_RESPONSE=(─┤ Routing options ├─)───┘
```

**Routing options:**

```
  ┌─INHERIT─┐
├─┼─ENABLE──┼──┬────────────────┬──┬────────────────────┬──────────────────────────────┤
  └─DISABLE─┘  └─,TARGET=target─┘  └─,FALLBACK=fallback─┘
```

## Parameters

**DEFN**=(**RTG**,*name*)
Define a routing rule called *name*. You can use any alphanumeric string of up to 8 characters. The group name is not case sensitive.

**DESC**='*definition description*'
A description of the definition. A description that contains spaces must be enclosed in quotation marks. The description can be up to 30 characters long.

**REPLACE**=**NO** | **YES**
Determines what happens if you try to input a definition with the same type and name as one that already exists in the repository:

**YES** The definition replaces the existing definition.

**NO** All ADD commands directed at the repository fail.

**`APPLY_RULE_TO`**`=(`*`scope`*`)`
The scope of the rule:

**ALL** All systems

**SYSTEM**,*name*
A system called "*name*".

**GROUP**,*name*
A system group called "*name*".

**`PLAN_NAME`**`=`*`routing_plan`*
Assigns this routing rule to a routing plan.

A routing rule that is not assigned to any routing plan is always in effect. Where there is a conflict between an unassigned routing rule and a routing rule that is explicitly assigned to the routing plan that is currently active, the rule that is assigned to the current routing plan takes precedence. In other words, you can define unconditional rules to specify generic routing behavior and then override that behavior using specific rules that are assigned to the current routing plan.

**`CONDITION`**`=(`**`DESTID`**`=`*`destid qualifier`*`)`
*destid* specifies a string that appears in the IRM_IMSDestID field on incoming messages.

This does not have to be the name of a datastore defined in the IMS Connect configuration member. It is simply used as a string to identify matching messages to which this rule applies.

*qualifier* is an optional condition that qualifies this rule. You can specify a qualifying rule without a matching master rule in the repository. If a master rule does not already exist for that DestID, an implied routing rule will be generated internally at run time.

**,QUALIFIER_TYPE**=**NONE**
If NONE is specified, this routing rule will apply to all transactions that have this original datastore ID (DestID). In other words, this will be the master routing rule for this DestID.

**,QUALIFIER_TYPE**=**TXL**,**QUALIFIER_LIST_NAME**=*quallist*
QUALIFIER_TYPE specifies the type of condition that qualifies this rule. TXL (transaction list) is the only supported type. *quallist* is the name of a defined transaction list. The values in this list are used to determine whether this rule applies to a given OTMA request.

**`SEND_RECEIVE`**`=(`*`routing options`*`)`
Specifies whether the routing rule is enabled for this message type at the level specified in the APPLY_RULE_TO parameter. The options are described in "Routing options" on page 300.

**`SEND_ONLY`**`=(`*`routing options`*`)`
Specifies whether the routing rule is enabled for this message type at the level specified in the APPLY_RULE_TO parameter. The options are described in "Routing options" on page 300.

**`RTPIPE`**`=(`*`routing options`*`)`
Specifies whether the routing rule is enabled for this message type at the level specified in the APPLY_RULE_TO parameter. The options are described in "Routing options" on page 300.

**SYNC_REQUEST**=(*routing options*)
: Specifies whether the routing rule is enabled for this message type at the level specified in the APPLY_RULE_TO parameter. The options are described in "Routing options."

**SYNC_RESPONSE**=(*routing options*)
: Specifies whether the routing rule is enabled for this message type at the level specified in the APPLY_RULE_TO parameter. The options are described in "Routing options."

### Routing options

The following routing parameters can be specified for each message type:

**ENABLE**
: This rule is enabled for this message type at the level specified in the APPLY_RULE_TO parameter. You must specify a TARGET routing list. The FALLBACK parameter is optional.

**DISABLE**
: This rule will not apply to this message type at the level specified in the APPLY_RULE_TO parameter. The TARGET and FALLBACK parameters are ignored.

**INHERIT**
: Inherit settings from rules applying to the same condition (IRM_IMSDestID) at either the All systems or Group level. A Group level rule will inherit settings from the All systems level. A system level rule will inherit settings from the Group level, if one is defined, otherwise from the All systems level. The TARGET and FALLBACK parameters are ignored.

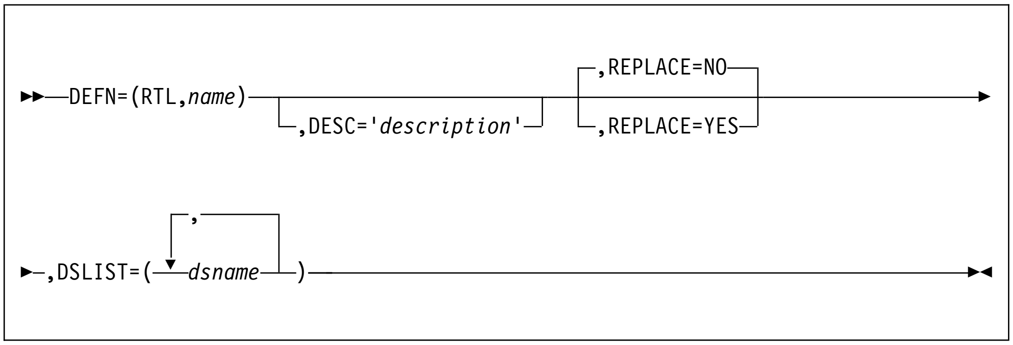
**TARGET**
: Specifies a routing list of datastores to use as candidates for routing messages of this type.

**FALLBACK**
: Specifies a routing list of datastores to use as candidates if the none of the datastores in the target routing list are available.

## OTMA routing list (RTL) ADD command

The DEFN RTL ADD command adds an OTMA routing list to a repository. A routing list allows you to define collections of datastores to use as candidates for rules-based routing of IRM messages.

### Format



### Parameters

**DEFN**=(**RTL**,*name*)
Define a routing list called *name*. You can use any alphanumeric string of up to 8 characters. The name is not case sensitive.

**DESC**='*definition description*'
A description of the definition. A description that contains spaces must be enclosed in quotation marks. The description can be up to 30 characters long.

**REPLACE**=**NO** | **YES**
Determines what happens if you try to input a definition with the same type and name as one that already exists in the repository:

**YES** The definition replaces the existing definition.

**NO** All ADD commands directed at the repository fail.

**DSLIST**=(*datastore1,datastore2,...*)
A comma-delimited list of datastores that are part of this routing list. The same datastore can be part of more than one routing list.

## Transaction list (TXL) ADD command

The DEFN TXL ADD command adds a transaction list to a repository. A transaction list is a collection of transaction codes. You can use a transaction list in an OTMA routing rule to route selected transactions to a different routing list from the one specified in the master routing rule. You only need this definition if you use routing.

### Format

```
                                                  ┌─,REPLACE=NO──┐
►►──DEFN=(TXL,name)──┬───────────────────────┬────┼──────────────┼────────────────►
                     └─,DESC='description'───┘    └─,REPLACE=YES─┘

                  ┌─,────────┐
                  ▼          │
►─,TRANLIST=(───dsname───┴─)──────────────────────────────────────────────────►◄
```

### Parameters

**DEFN**=(**TXL**,*name*)
Define a transaction list called *name*.

**DESC**='*definition description*'
A description of the definition. A description that contains spaces must be enclosed in quotation marks. The description can be up to 30 characters long.

**REPLACE**=**NO** | **YES**
Determines what happens if you try to input a definition with the same type and name as one that already exists in the repository:

**YES** The definition replaces the existing definition.

**NO** All ADD commands directed at the repository fail.

**TRANLIST**=(*trancode*)

A comma-delimited list of transaction codes for this list.

Transaction codes can be fully qualified transaction names or generic transaction names. A generic transaction name consists of one or more characters followed by a trailing asterisk (*). For example, AB* would match any transaction code starting with AB, such as AB, ABC, and ABC12. Embedded asterisks are not allowed in generic transaction names.

## Transaction code (TXN) ADD command

The DEFN TXN ADD command adds a transaction code to a repository. An application is a list of transaction codes with routing options for those codes. Application definitions associate transactions with a system. They are required for transaction routing and other transaction-based features.

### Format

```
                                                    ┌─,REPLACE=NO──┐
►►──DEFN=(TXN,name*)──┬─────────────────────────┬──┼──────────────┼──,APP=application_name────────────────►
                      └─,DESC='description'─────┘  └─,REPLACE=YES─┘

   ┌─,OVERRIDE_APP_RULES=NO──┐   (1) ┌─,OVERRIDE_TRANSACTION_TIMER=NO────────────────────────┐
►──┼─────────────────────────┼───────┼───────────────────────────────────────────────────────┼──────────►
   └─,OVERRIDE_APP_RULES=YES─┘       └─,OVERRIDE_TRANSACTION_TIMER=(YES,MSGTO=xx,ACKTO=xx)───┘

   (2) ┌─,OVERRIDE_TRANSACTION_EXPIRATION=NO─────────────────────────┐
►──────┼─────────────────────────────────────────────────────────────┼────────────────────────────────────►
       └─,OVERRIDE_TRANSACTION_EXPIRATION=(YES,F1_TRNEXP=──┬─ON──┬─)─┘
                                                           └─OFF─┘

   (3) ┌─,OVERRIDE_CLIENTID_CANCELLATION=NO─────────────────────────┐
►──────┼────────────────────────────────────────────────────────────┼─────────────────────────────────────►
       └─,OVERRIDE_CLIENTID_CANCELLATION=(YES,F3_CANCID=──┬─ON──┬─)─┘
                                                          └─OFF─┘

   ┌─,ALTERNATE_TRANSACTION_CODE=NO──────────────────────────┐
►──┼─────────────────────────────────────────────────────────┼────────────────────────────────────────────►
   └─,ALTERNATE_TRANSACTION_CODE=(YES,OFFSET=nnnn,LENGTH=n)──┘

   (4) ┌─,TRANSACTION_ROUTING=INACTIVE─┐                                                    (5)
►──────┼───────────────────────────────┼──┬────────────────────────────────────────────────────┬──────────►◄
       └─,TRANSACTION_ROUTING=ACTIVE───┘  │                               ┌─,FAILURE=ORIGINAL─┐ │
                                          └─,ROUTING_RULE=(──┬─ALL──────┬─┼───────────────────┼─)─┘
                                                             ├─DS,name──┤ └─,FAILURE=REJECT───┘
                                                             ├─DSG,name─┤
                                                             └─AFL,name─┘
```

**Notes:**

1 TRANSACTION_TIMER is a valid synonym for the OVERRIDE_TRANSACTION_TIMER keyword.

2 TRANSACTION_EXPIRATION is a valid synonym for the OVERRIDE_TRANSACTION_EXPIRATION keyword.

3 CLIENTID_CANCELLATION is a valid synonym for the OVERRIDE_CLIENTID_CANCELLATION keyword.

4 ROUTING is a valid synonym for the TRANSACTION_ROUTING keyword.

5 You must specify ROUTING_RULE if you activate transaction routing and override the application rules.

## Parameters

**DEFN**=(**TXN**,*name**)

Define a transaction code called *name*. You can use an asterisk in the name to specify a mask. For example, specify `*` to create transaction routing rules for all transactions; specify `AB*` to create transaction routing rules for all transactions beginning with `AB`.

You can specify `$NOTRAN$` to create a special transaction routing rule. For messages without a valid transaction code the $NOTRAN$ record is checked. If the $NOTRAN$ record exists, it is used to route the message, otherwise the original data set is used. Note that no wild card searching can be performed, so the record must match exactly.

**DESC**='*definition description*'

A description of the definition. A description that contains spaces must be enclosed in quotation marks. The description can be up to 30 characters long.

**REPLACE**=**NO** | **YES**

Determines what happens if you try to input a definition with the same type and name as one that already exists in the repository:

**YES** The definition replaces the existing definition.

**NO** All ADD commands directed at the repository fail.

**APP**=*application_name*

The name of the application for this transaction. A transaction can only belong to one application.

**OVERRIDE_APP_RULES**=**YES** | **NO**

Specify whether to override the transaction routing rule specified in the application.

**Note:** In order for routing to be active it must be on for both the transaction code and application regardless of this option.

**OVERRIDE_TRANSACTION_TIMER**=(**NO** | **YES**,**MSGTO**=*xx*,**ACKTO**=*xx*)

The transaction timer feature allows you to set a timeout value in the IRM header of messages for this transaction or group of transactions. Setting OVERRIDE_TRANSACTION_TIMER to YES means that the timeout values specified in the transaction definition will override the IRM setting.

When you override the transaction timer, you must also specify a message timeout value and optionally an ACK/NAK timeout value in IRM timer format. The acceptable field values are:

**00** Default

**E9** No timer

**FF** Wait indefinitely

***xx*** A hexadecimal representation of the timeout value in minutes, seconds, or hundredths of a second

For a full description of how to specify values in IRM timer format, refer to the Transaction Timer panel in the ISPF Help or see the “Timer interval specifications” topic in the *IMS Communications and Connections Guide*.

**MSGTO**=*xx*

Message timeout value.

**ACKTO**=*xx*

ACK/NAK timeout value (Optional).

**OVERRIDE_TRANSACTION_EXPIRATION=NO | YES**

IRM_F1_TRNEXP is a flag on the IRM that determines whether the transaction expiration feature is active or inactive. If active, then IMS Connect sets the expiration time for the input transaction. Setting OVERRIDE_TRANSACTION_EXPIRATION to YES means that the value specified for F1_TRNEXP in the transaction definition (either ON or OFF) will override the IRM setting.

**F1_TRNEXP=ON | OFF**

`ON` indicates that transaction expiration is enabled for this transaction.

**OVERRIDE_CLIENTID_CANCELLATION=NO | YES**

IRM_F3_CANCID is a flag on the IRM that determines whether the Client ID Cancellation feature is active or inactive. If it is active, then a message that specifies the same client ID as that of an active session will cancel the original session and then run. Setting OVERRIDE_CLIENTID_CANCELLATION to YES means that the value specified for F3_CANCID in the transaction definition (either ON or OFF) will override the IRM setting.

**F3_CANCID=ON | OFF**

`ON` indicates that cancellation of duplicate sessions is enabled for this transaction.

**ALTERNATE_TRANSACTION_CODE=(NO | YES,OFFSET=***nnnn***,LENGTH=***n***)**

Some applications use a single or common IMS transaction code, and store the actual transaction code elsewhere in the transaction payload. For example, OMEGAMON refers to the common transaction code as the umbrella transaction and the alternate transaction code as the user code.

If the Alternate Transaction Code feature is activated for this definition, IMS Connect Extensions uses the associated Offset and Length fields to locate the alternate transaction code within the transaction payload.

The offset value can be anywhere in the range 6 - 9980, but it must be found in the first application segment of the transaction. The length must be in the range 1 - 8.

**TRANSACTION_ROUTING=ACTIVE | INACTIVE**

Specifies whether transaction routing is active for this transaction. If active, then messages with a transaction code matching the name of this transaction **can** be routed. If inactive, then messages with this transaction code will never be routed. That is, IMS Connect Extensions will not alter their destination.

ROUTING is a valid synonym for the TRANSACTION_ROUTING keyword, but note that specifying both ROUTING and TRANSACTION_ROUTING will result in an error

**ROUTING_RULE**=(*candidate_datastores*)

This parameter consists of a transaction routing rule and a failure option:

A transaction routing rule consists of a list of *candidate datastores*. IMS Connect Extensions will only route messages to those datastores that are candidate datastores for both the transaction code and original destination datastore of an incoming message. You can specify *candidate_datastores* as one of the following:

**ALL** All datastores

**DS**,*name*

A datastore called "*name*".

**DSG,***name*

A datastore group called "*name*".

**AFL,***name*

An affinity list called "*name*".

The transaction routing rule also consists of a *failure option*. This option specifies what to do when a message has no eligible candidate datastores (because the transaction and original destination datastore do not share any available candidates). Choose from either:

**FAILURE=ORIGINAL**

Use the destination ID set by the message (the original datastore) as the target datastore.

**FAILURE=REJECT**

Reject the message.

Here is an example transaction routing rule:

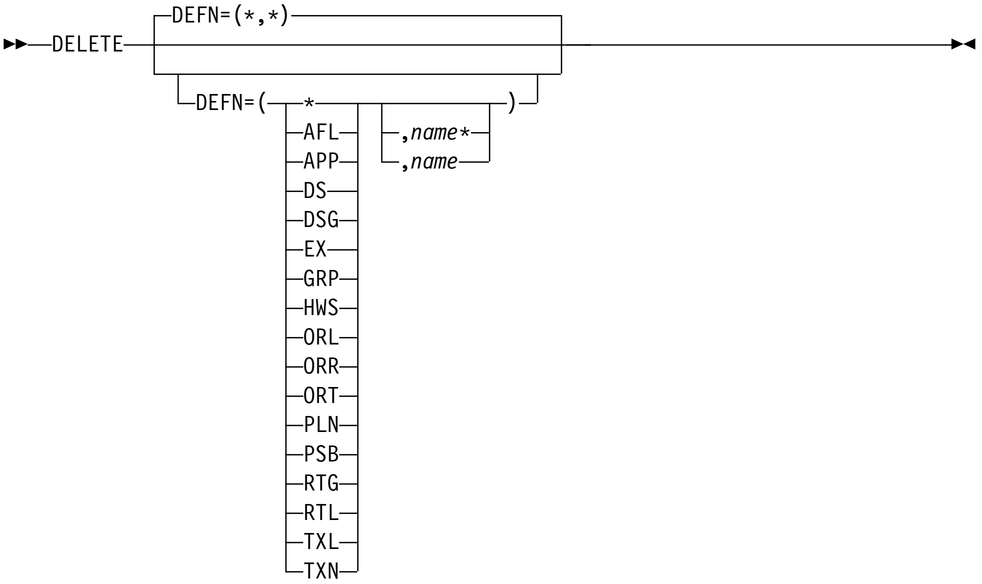
```
ROUTING_RULE=(AFL,MYATMDS,FAILURE=ORIGINAL)
```

With this rule, an incoming message for this transaction code can potentially be routed to one of the datastores in the affinity list. If the transaction routing rule for the destination datastore on the incoming message does not specify at least one candidate datastore from the MYATMDS affinity list that is also available, then, based on the failure option, the message will be passed through to IMS Connect with the original datastore.

## DELETE command

The definition maintenance utility **DELETE** command deletes definitions from the repository.

### Format



### Parameters

**DEFN**=(*type*,*name_mask*)

Specifies the type and name of the definition you want to delete. You can use a name mask to delete all definitions containing a matching substring. For example, to delete all datastore definitions beginning with AB enter:

```
DELETE DEFN=(DS,AB*)
```

**Note:** Transaction names can contain literal asterisks. When you delete transactions using an asterisk, that asterisk will always be treated as a wildcard. For example, suppose you have these transaction codes in the repository:

```
AB*
ABC
ABCD
```

If you specify the following command, all of these definitions would be deleted, not just "AB*":

```
DELETE DEFN=(TXN,AB*)
```

## SET command

Use the **SET** command to set global options for any other definition maintenance utility commands.

### Format

```
►►──SET──OPTION=(──┬─────────────┬──┬────────────────┬──)──────────────────►◄
                   └─REPLACE=YES─┘  └─WARNING=IGNORE─┘
```

### Parameters

**REPLACE**

Determines whether the definition maintenance utility replaces any existing definitions it encounters or skips any definitions that are already in the repository. The default behavior is to skip definitions that are already in the repository. To replace any existing definition:

```
SET OPTION=(REPLACE=YES)
```

If any SET command is present in the input which has the REPLACE=YES option then that is in effect for all commands whether they appear before or after the SET command.

**WARNING**

Determines whether the utility continues processing when a warning-level control card error has been detected. The default behavior is that processing ends.

If the WARNING=IGNORE option is set, the utility continues to execute.

```
SET OPTION=(WARNING=IGNORE)
```

If any SET command is present in the input which has the WARNING=IGNORE option then that is in effect for all commands whether they appear before or after the SET command.

## Definition takeup utility

The definition takeup utility reads an IMS Connect configuration member and produces definitions for that system, its datastores, and its exits. Use the definition maintenance utility to put the definitions produced by this utility into an IMS Connect Extensions repository.

The takeup utility program name is CEXDFDST. You can invoke the program by passing its name to the FUNEXEC program; you can run FUNEXEC from JCL as an executable job step program.

### Definition takeup utility - example JCL

Here is an example takeup utility JCL that generates definition **ADD** commands from a configuration member:

```
⋮
//STEP     EXEC PGM=FUNEXEC,PARM='CEXDFDST'
//STEPLIB  DD DISP=SHR,DSN=funpre.SFUNLINK
//         DD DISP=SHR,DSN=cexpre.SCEXLINK
//CONFDS   DD DISP=SHR,DSN=hwscfg.data.set
//CEXOUT   DD UNIT=SYSDA,SPACE=(CYL,(5,2)),DISP=(,PASS),DSN=&SYSIN
//MSGOUT   DD SYSOUT=*
//SYSIN    DD *
*Generate CEX ADD commands from HWS conf member

TAKEUP CONFIG=CONFDS
       MEMBER=HWSCFG1,
       CEX_PORT=1234,
       HLQ=MY.JOURNAL,
       ARCHIVE_JCL=cexpre.SCEXSAMP(CEXARCH1)

TAKEUP CONFIG=CONFPS 1
       MEMBER=HWSCFG1,
       CEX_PORT=1235,

⋮
more batch commands
/*
```

**1** You can specify multiple **TAKEUP** commands to create definitions for multiple systems at the same time. All you have to change is the name of the member and the IMS Connect Extensions console port number.

Specifying multiple **TAKEUP** commands is better than submitting multiple TAKEUP jobs because the utility will identify duplicate datastores and exits in the different IMS Connect configuration members, and so only include a unique definition for each, only if you specify the members in the same job.

### Job control statements

The job control statements in the preceding example are:

**STEP EXEC**
Specifies the program name FUNEXEC with the name of the takeup utility, CEXDFDST, as a parameter.

**STEPLIB DD**
IMS Connect Extensions batch utilities require modules in the IMS Connect Extensions (CEX) link library and its functional support (FUN) link library.

**CONFDS DD**
Specifies the name of a data set containing IMS Connect configuration members (HWSCFG).

**CEXOUT DD**
Defines a sequential data set for the command output. Use the command output as input for the maintenance utility. The data set can be written to a system output device, a tape, or a DASD volume. This data set has the following attributes:
LRECL=81
RECFM=FBA

You can override the ddname with a **SET** command.

**MSGOUT DD**
Defines a sequential data set for message output. The data set can be written to a system output device, a tape, or a DASD volume. This data set has the following attributes:
LRECL=133
RECFM=FBA

**SYSIN DD**
Defines an input sequential data set (SYSIN) containing the batch commands that you want to process. The data set can be defined as job control in-stream data (as shown in the preceding example), a tape file, or a DASD file. This data set has the following attributes:
LRECL=80
RECFM=FB

# TAKEUP command

The **TAKEUP** command takes an IMS Connect configuration member and produces **ADD** commands for the definition maintenance utilities.

## Format

```
                                                  ┌─,HOST=LOCALHOST──────────┐
►►──TAKEUP──CONFIG=ddname──,MEMBER=hwscfg─────────┼──────────────────────────┼──────────►
                                                  └─,HOST=host_name_or_IP_addr┘

►─,CEX_PORT=nnnnn──┬─────────────────────────────────────────┬────────────►◄
                   └─,HLQ=journal_hlq,ARCHIVE_JCL=skeleton(name)┘
```

## Parameters

**CONFIG**=*ddname*
The ddname of the library containing the IMS Connect configuration (HWSCFG) member.

**MEMBER**=*hwscfg*
The name of the HWSCFG member you want to takeup definitions from.

**HOST**=*host_name_or_IP_addr*
The host name or IP address of the server on which this IMS Connect system will run. The default is localhost (the local loopback address).

**CEX_PORT**=*port_number*
Specifies the IMS Connect Extensions console listener port. This TCP/IP port is used by IMS Connect Extensions clients to connect to IMS Connect Extensions enabled IMS Connect systems and provide operational control and monitoring for them. This port should not be confused with the ports IMS Connect uses to process messages. Assign an unallocated TCP/IP port using decimal port values (1 - 65535).

**HLQ**=*journal_hlq*
The high-level qualifier for the active and archive journal data sets. Omit this parameter if you do not want to activate event collection. If you specify this parameter you must also specify the name of the archiving JCL skeleton.

**ARCHIVE_JCL**=*skeleton(name)*
The name of the skeleton for the archiving job. If you are testing IMS Connect Extensions, use the sample skeleton located in the SCEXSAMP library called CEXARCH1. In the sample, change the `active.journal` qualifier in:

```
LIKE=active.journal.P01
```

to the high-level qualifier you specified in the HLQ parameter, followed by the word “ACTIVE”. For example:

```
LIKE=MYHLQ.ACTIVE.P01
```

## SET command

Use the **SET** command to set global options for any other definition takeup utility commands that follow.

### Format

```
►►──SET──┬──────────────────┬─┬──────────────────────────────────────┬──────────────►◄
         └─CEXOUT=ddname─┘   │                    ┌─ABORT──┐        │
                             └─OPTION=(WARNING=──┴─IGNORE─┴──)─┘
```

### Parameters

**CEXOUT**=*ddname*
Set the output of the command to the data set specified by *ddname*.

**WARNING**
Determines whether the utility continues processing when a warning-level control card error has been detected.

If the WARNING=IGNORE option is set, the utility continues to execute.

```
SET OPTION=(WARNING=IGNORE)
```

If WARNING=ABORT is set, execution of the utility ends. This is the default.

## Definition extract utility

The definition extract utility reads an IMS Connect Extensions repository. It can produce **ADD** commands for the definition maintenance utility or a configuration file for the Operations Console import facility.

In *extract* mode, the utility can extract and transform attributes from definitions, for example changing the name prefixes from TEST to PROD. Extract mode is indicated by the presence of a SYSIN DD statement containing one or more batch commands.

In this mode, use the utility for tasks such as:

- Migrating definitions from test to production systems.
- Copying definitions from one site to another.
- Using ISPF Edit (or similar) to modify values for definitions.

In *CSV export* mode, the utility extracts system and group definitions to an import file that can be used to distribute a standard configuration to Operations Console users. CSV export mode is indicated by the presence of a CEXCSVFI DD statement. The file is written in comma-separated values (CSV) format and can be used by the Operations Console import facility. Double quotation marks (") in the system description field are changed to single quotation marks (').

The SYSIN DD statement and CEXCSVFI DD statement are mutually exclusive.

## Example JCL: extract and transform definitions

Here is an example extract utility JCL that generates definition **ADD** commands from an IMS Connect Extensions repository:

```
⋮
//STEP     EXEC PGM=FUNEXEC,PARM='CEXDFDSO'
//STEPLIB  DD DISP=SHR,DSN=funpre.SFUNLINK
//         DD DISP=SHR,DSN=cexpre.SCEXLINK
//REPOSTRY DD DISP=SHR,DSN=cex.definition.repository
//CEXOUT   DD UNIT=SYSDA,SPACE=(CYL,(5,2)),DISP=(,PASS),DSN=&SYSIN
//MSGOUT   DD SYSOUT=*
//SYSIN    DD *
*Transform rule for extract
TRANSFORM NAME=TOPROD, TYPE=HWS, SOURCE=TEST*, TARGET=PROD*
*Extract all definitions; change hws name from test to prod
EXTRACT DEFN=* TRANSFORM(TOPROD)
⋮
more batch commands
/*
```

## Example JCL: create Operations Console import file

Here is an example extract utility JCL that generates a list of all system and group definitions in an IMS Connect Extensions repository. The presence of the CEXCSVFI DD statement indicates that the “extract to CSV file” function is required.

```
⋮
//STEP     EXEC PGM=FUNEXEC,PARM='CEXDFDSO'
//STEPLIB  DD DISP=SHR,DSN=funpre.SFUNLINK
//         DD DISP=SHR,DSN=cexpre.SCEXLINK
//REPOSTRY DD DISP=SHR,DSN=cex.definition.repository
//CEXCSVFI DD  SYSOUT=*,RECFM=FB,LRECL=160
//CEXOUT   DD UNIT=SYSDA,SPACE=(CYL,(5,2)),DISP=(,PASS),DSN=&SYSIN
//MSGOUT   DD SYSOUT=*
⋮
```

## Job control statements

The job control statements in these example are:

**STEP EXEC**
Specifies the program name FUNEXEC with the name of the take-up utility, CEXDFDSO, as a parameter.

**STEPLIB DD**
IMS Connect Extensions batch utilities require modules in the IMS Connect Extensions (CEX) link library and its functional support (FUN) link library.

**REPOSTRY DD**
Specify the name of the definitions repository from which to extract definitions. You can override the ddname with a **SET** command.

**CEXCSVFI DD**
Specify the name of a sequential data set to which a comma-separated list of system and group definitions will be written. This data set has the following attributes:

LRECL=160
RECFM=FB

**CEXOUT DD**
Defines a sequential data set for the command output. Use the command output as input for the maintenance utility. The data set can be written to a system output device, a tape, or a DASD volume. This data set has the following attributes:

LRECL=81
RECFM=FBA

You can override the ddname with a **SET** command.

**MSGOUT DD**
Defines a sequential data set for message output. The data set can be written to a system output device, a tape, or a DASD volume. This data set has the following attributes:

LRECL=133
RECFM=FBA

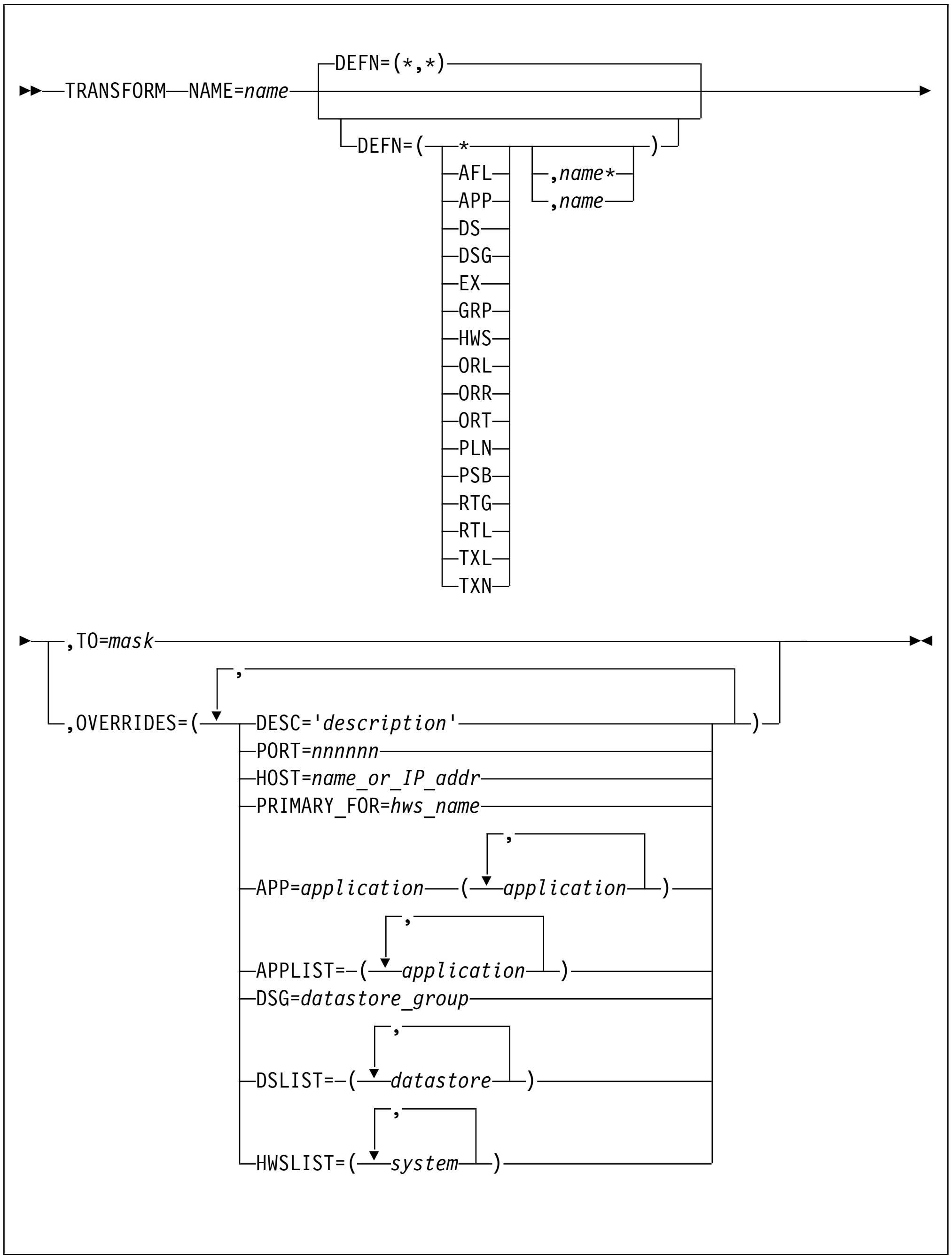
**SYSIN DD**
Defines an input sequential data set (SYSIN) containing the batch commands that you want to process. The data set can be defined as job control in-stream data (as shown in the preceding example), a tape file, or a DASD file. This data set has the following attributes:

LRECL=80
RECFM=FB

## TRANSFORM command

The **TRANSFORM** command creates a transform rule. The rule specifies how to modify a definition that is selected for extraction by the **EXTRACT** command.

## Format



## Parameters

**NAME**=*rule_name*
The name of the transform rule.

**DEFN**=(*type*,*name_mask*)
Specifies the type and name of the definition you want to transform. You can use a name mask to transform a substring from a definition. For example, to transform all definitions prefixed with DEVT enter:

```
DEFN=(*,DEVT*)
```

**Note:** Transaction names can contain literal asterisks. When you transform transactions using an asterisk, the asterisk will always be treated like a wildcard.

**TO**=*name_transform*
The name you want to transform the definition to. For example, to transform all datastore definitions from DEVT to PROD, enter:

```
TRANSFORM DEFN=(DS,DEVT*) TO=TEST*
```

**OVERRIDES**=*parameter_overrides*
An attribute, within a definition that you want to override. For example, to transform the description of every datastore definition enter:

```
TRANSFORM DEFN=(DS,*) OVERRIDES=(DESC="Added with the extract utility")
```

These are the available overrides:

**DESC**='*description*'
The description of the definition.

**PORT**=*nnnnnn*
The TCP/IP port on which IMS Connect Extensions clients connect to the console listener.

**HOST**=*name_or_IP_addr*
The name or IP address of the host running IMS Connect.

**PRIMARY_FOR**=*hws_name*
The system for which a datastore is a primary.

**APP**=*application*
The application a transaction belongs to.

**APPLIST**=(*application*,...)
The applications associated with an IMS Connect system.

**DSG**=*datastore_group*
The datastore group of a datastore.

**DSLIST**=(*datastore*,...)
The datastores in an affinity list.

**HWSLIST**=(*hws*,...)
The IMS Connect systems in a system group.

## EXTRACT command

The **EXTRACT** command extracts matching definitions from the repository; optionally applying transform rules to those definitions.

### Format

```
           ┌─DEFN=(*,*)──────────────────────────┐
►►──EXTRACT─┼──────────────────────────────────────┼──┬───────────────────────────────────┬──────────────►◄
           └─DEFN=(──┬─*───┬──┬──────────┬──)─┘  └─,TRANSFORM=─┬─rule_name────────────┤
                     ├─AFL─┤  ├─,name*───┤                      │     ┌─,──────────┐   │
                     ├─APP─┤  └─,name────┘                      └─(───▼─rule_name──┴─)─┘
                     ├─DS──┤
                     ├─DSG─┤
                     ├─EX──┤
                     ├─GRP─┤
                     ├─HWS─┤
                     ├─ORL─┤
                     ├─ORR─┤
                     ├─ORT─┤
                     ├─PLN─┤
                     ├─PSB─┤
                     ├─RTG─┤
                     ├─RTL─┤
                     ├─TXL─┤
                     └─TXN─┘
```

### Parameters

**DEFN**=(*type*,*name_mask*)
: Specifies the type and name of the definition you want to extract. You can use a name mask to extract definitions containing a matching substring. For example, to extract all definitions prefixed with DEVT enter:

```
DEFN=(*,DEVT*)
```

**Note:** Transaction names can contain literal asterisks. When you extract transactions using an asterisk, the asterisk will always be treated like a wildcard.

**TRANSFORM**=*rule_name*
: The name of the transform rule you want to apply to the definitions.

## SET command

Use the **SET** command to set global options for any other definition extract utility commands that follow.

### Format

```
►►──SET──┬──────────────────┬──┬────────────────────┬──────────────────────►◄
         └─CEXOUT=ddname─┘  └─REPOSTRY=ddname─┘
```

### Parameters

**CEXOUT**=*ddname*
: Set the output of the command to the data set specified by *ddname*.

**REPOSTRY**=*ddname*
Change the definition repository to the data set specified by *ddname*.

### Examples

The **SET** command can make the syntax of the extract command more concise:

```
⋮
/
//REPOSTRY DD DISP=SHR,DSN=REPOSTRY.ONE
//CEXOUT   DD UNIT=SYSDA,SPACE=(CYL,(5,2)),DISP=(,PASS),DSN=OUT.ONE
//REP2     DD DISP=SHR,DSN=REPOSTRY.TWO
//OUT2     DD UNIT=SYSDA,SPACE=(CYL,(5,2)),DISP=(,PASS),DSN=OUT.TWO
//MSGOUT   DD SYSOUT=*
//SYSIN    DD *
EXTRACT DEFN=(*,*) 1
SET REPOSTRY=REP2,CEXOUT=OUT2 2
EXTRACT DEFN=(*,*)
⋮
more batch commands
/*
```

**1** The first EXTRACT statement reads the repository specified by the REPOSTRY DD and extracts to the data set specified by CEXOUT.

**2** The **SET** command has switched the CEXOUT and REPOSTRY DD. The second EXTRACT statement reads the repository specified by the REP2 DD and extracts to the data set specified by OUT2.

## Definition import/export utility

You can use the import/export utility to back up, restore, and verify the IMS Connect Extensions definitions repository.

Unlike the definition extract and definition maintenance utilities, you cannot use this utility to modify or view the values for individual fields. However, this utility can be used to back up and restore IMS Connect Extensions internal records, which the other definition utilities cannot do.

### EXEC statement

The format of the EXEC statement is:

```
//stepname  EXEC PGM=FUNEXEC,PARM='CEXDFMIG'
```

### DD statements

**SEQFILE DD**
Sequential data set used as the target of the EXPORT function or the source of the IMPORT function. You can override the ddname with a **SET** command.

**MSGOUT DD**
Print data set containing processed utility statements.

**REPOSTRY DD**
Definitions data set used as the source of the EXPORT function or the target of the IMPORT function. You can override this ddname with a **SET** command.

**SYSIN DD**
Defines an input sequential data set (SYSIN) containing the batch

commands that you want to process. The data set can be defined as job control in-stream data (as shown in the preceding example), a tape file, or a DASD file. This data set has the following attributes:

LRECL=80
RECFM=FB

### CEXEXPRT sample JCL

The member CEXEXPRT in the SCEXSAMP library contains a sample job that exports records from a repository.

```
//EXPORT   EXEC PGM=FUNEXEC,PARM='CEXDFMIG'
//STEPLIB  DD DISP=SHR,DSN=funpre.SFUNLINK
//         DD DISP=SHR,DSN=cexpre.SCEXLINK
//MSGOUT   DD SYSOUT=*
//REPOSTRY DD DISP=SHR,DSN=<repository>
//SEQFILE  DD DISP=SHR,DSN=<export.output>
//SYSIN    DD *
EXPORT DEFN=(*,*),ID='Full Export',PRINTKEY=YES
/*
```

*Figure 106. Repository records migration utility JCL (export)*

### CEXIMPRT sample JCL

Another sample member, CEXIMPRT, is an example job that imports definitions into a repository. It is identical to CEXEXPRT but uses an import SYSIN deck:

```
IMPORT DEFN=(*,*),REPLACE=YES,PRINTKEY=YES
VERIFY
```

The VERIFY statement reports on the referential integrity of the repository. The sample member CEXVERFY shows this utility being used to submit a standalone VERIFY statement.

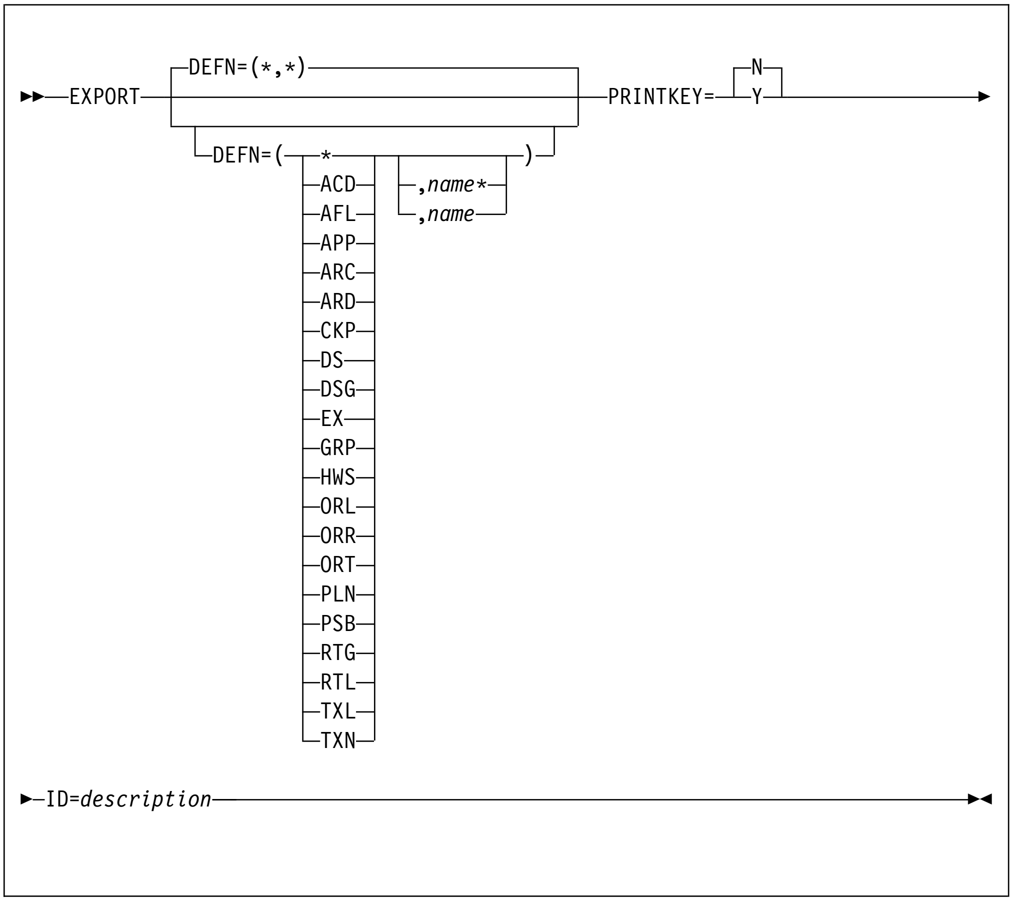
You can specify both an IMPORT and an EXPORT statement within one run, as shown in this example:

```
* EXPORT OLD REPOSITORY
SET REPOSTRY=REPOSOLD
EXPORT DEFN=(*,*)
* IMPORT NEW REPOSITORY
SET REPOSTRY=REPOSNEW
IMPORT DEFN=(*,*)
* VERIFY NEW REPOSITORY
VERIFY
```

## EXPORT command

Use the **EXPORT** command to export definitions from a repository.

## Format



```
            ┌─DEFN=(*,*)──────────────────────────┐              ┌─N─┐
►►──EXPORT──┼─────────────────────────────────────┼──PRINTKEY=───┴─Y─┴──────────────►
            └─DEFN=(──┬─*───┬──┬──────────┬──)──┘
                      ├─ACD─┤  ├─,name*───┤
                      ├─AFL─┤  └─,name────┘
                      ├─APP─┤
                      ├─ARC─┤
                      ├─ARD─┤
                      ├─CKP─┤
                      ├─DS──┤
                      ├─DSG─┤
                      ├─EX──┤
                      ├─GRP─┤
                      ├─HWS─┤
                      ├─ORL─┤
                      ├─ORR─┤
                      ├─ORT─┤
                      ├─PLN─┤
                      ├─PSB─┤
                      ├─RTG─┤
                      ├─RTL─┤
                      ├─TXL─┤
                      └─TXN─┘

►──ID=description────────────────────────────────────────────────────────────────►◄
```

## Parameters

**DEFN**=(*type*,*name_mask*)

Specifies the type and name of the definition you want to export.

You can use a name mask to select based on a substring from a definition. For example, to select all definitions prefixed with DEVT enter:

```
DEFN=(*,DEVT*)
```

You can specify multiple DEFN statements separated by commas. For example:

```
EXPORT DEFN=(ACD,*),                                              +
       DEFN=(AFL,*),                                              +
       DEFN=(TXN,*)
```

To specify a definition type, use one of the following:

| | |
|---|---|
| **ACD** | Active journal definitions |
| **AFL** | Affinity lists |
| **APP** | Applications |
| **ARC** | Archive data set records |
| **ARD** | Archive journal definitions |
| **CKP** | Journal checkpoint records |
| **DS** | Datastores |

| | |
|---|---|
| **DSG** | Datastore groups |
| **EX** | User message exits |
| **GRP** | System groups |
| **HWS** | System definitions |
| **ORL** | ODBM routing lists |
| **ORR** | ODBM routing rules |
| **ORT** | ODBM targets |
| **PLN** | Routing plans |
| **PSB** | PSB name lists |
| **RTG** | OTMA routing rules |
| **RTL** | OTMA routing lists |
| **TXL** | Transaction lists |
| **TXN** | Transactions |

**ID** 1-20 characters which are printed as an identifier in the export sequential data set.

**PRINTKEY**
Does one of the following:

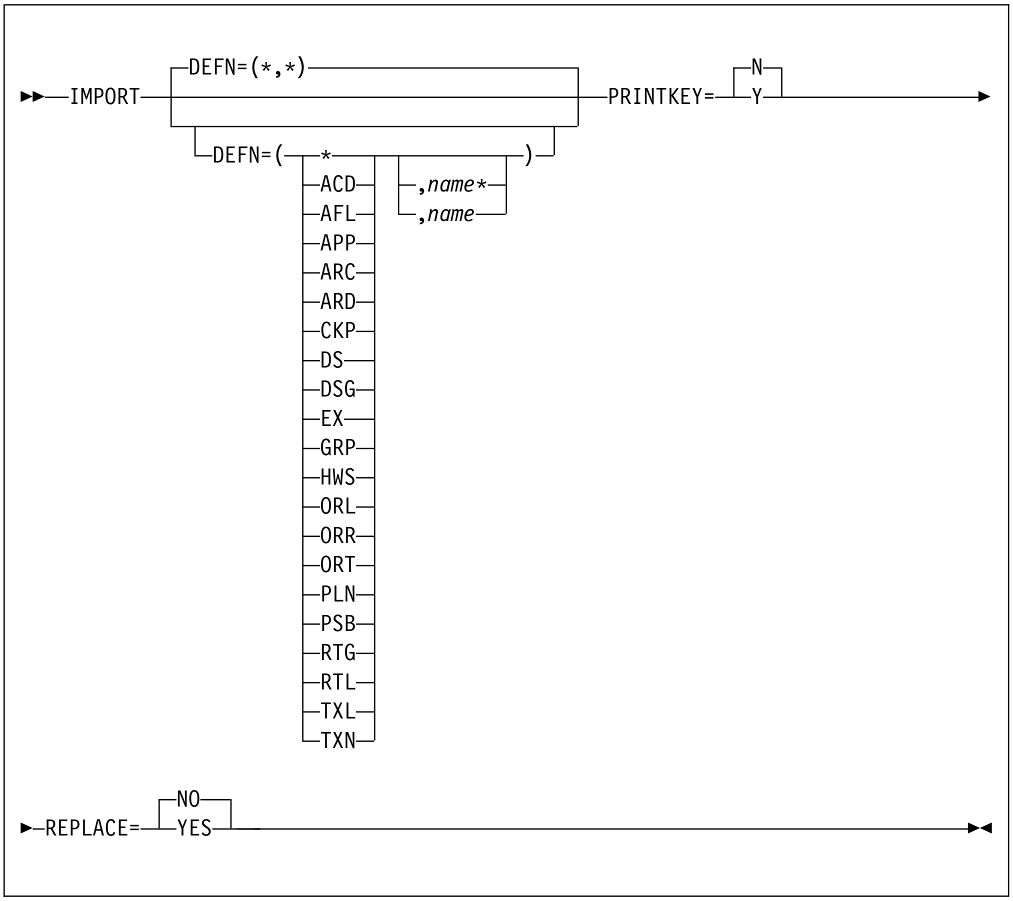
**Y** Lists a summary of records imported or exported.

**N** (Default) Does not list a summary of records imported or exported.

## IMPORT command

Use the **IMPORT** command to import definitions into a repository.

## Format



## Parameters

**DEFN**=(*type*,*name_mask*)

Specifies the type and name of the definition you want to import.

You can use a name mask to select based on a substring from a definition. For example, to select all definitions prefixed with DEVT enter:

```
DEFN=(*,DEVT*)
```

You can specify multiple DEFN statements separated by commas. For example:

```
IMPORT DEFN=(ACD,*),                                                    +
       DEFN=(AFL,*),                                                    +
       DEFN=(TXN,*)
```

To specify a definition type, use one of the following:

**ACD** Active journal definitions

**AFL** Affinity lists

**APP** Applications

**ARC** Archive data set records

**ARD** Archive journal definitions

**CKP** Journal checkpoint records

**DS** Datastores

| | |
|---|---|
| **DSG** | Datastore groups |
| **EX** | User message exits |
| **GRP** | System groups |
| **HWS** | System definitions |
| **ORL** | ODBM routing lists |
| **ORR** | ODBM routing rules |
| **ORT** | ODBM targets |
| **PLN** | Routing plans |
| **PSB** | PSB name lists |
| **RTG** | OTMA routing rules |
| **RTL** | OTMA routing lists |
| **TXL** | Transaction lists |
| **TXN** | Transactions |

**REPLACE**
Determines whether existing definitions are replaced or not.

**PRINTKEY**
Does one of the following:

**Y** Lists a summary of records imported or exported.

**N** (Default) Does not list a summary of records imported or exported.

# VERIFY command

The **VERIFY** command verifies the referential integrity of the definitions repository, and produces a report that outlines the current structure and summarizes any problems.

## Format

```
►►—VERIFY——————————————————————————————————————————————————————►◄
```

## Parameters

There are no additional parameters for this command.

# SET command

Use the **SET** command to set global options for any other import/export utility commands that follow.

## Format

```
►►──SET──┬────────────────┬──┬─────────────────┬──────────────────►◄
         └─SEQFILE=ddname─┘  └─REPOSTRY=ddname─┘
```

## Parameters

**SEQFILE**=*ddname*
Set the definitions sequence file to the data set specified by *ddname*.

**REPOSTRY**=*ddname*
Change the definition repository to the data set specified by *ddname*.

# Chapter 23. Reporting utilities

This topic describes IMS Connect event records and the options for reporting using the IMS Connect Extensions print utility and IMS Performance Analyzer.

## Active session utility

The active session utility assists in problem determination by analyzing an IMS Connect Extensions Journal data set and listing details of active sessions.

An active session is one for which a Read Prepare event for a port and socket has occurred and the Trigger event following a Close Socket event has not yet occurred. Sessions that use persistent sockets will be active following the Trigger event.

To execute the utility, perform the following steps:

1. Use the JOURNAL SWITCH command to switch the Active Journal.

   Perform this step so that you can provide the utility a journal containing recent data, but one that is not concurrently being used by the journaling process. For more information, see “Refreshing journal settings and switching the active journal” on page 244.
2. View the IMS Connect Extensions log (**L** line action from the Operations dialog) to determine the fully qualified name of the most recent Active Journal data set.
3. Run the utility against the desired journal. See “Sample JCL: CEXASUT” on page 324 for an example job.

The report includes the socket and event key for the active sessions. You can use this information with either the IMS Connect Extensions print utility or IMS Problem Investigator for z/OS to view and analyze related event records.

To analyze active sessions in real time, use the ISPF Operations dialog or Operations Console; the batch utility can only report sessions that have not ended at the time IMS Connect Extensions was writing to that journal.

**Note:** To display active session details for a system, event collection must be active and the collection level must be greater than 1.

### EXEC statement

The format of the EXEC statement is:

```
//stepname  EXEC  PGM=FUNEXEC,PARM='CEXJASKT,help,case'
```

This executes the IMS Connect Extensions active session utility, CEXJASKT. The parameters are:

***help*** Specifies whether help information is provided in the report.

**Y** Help description lines are printed in the report.

**N** Help description lines are not printed in the report.

***case*** Specify **U** to print the report in uppercase. Omit this value to print in mixed case.

## Sample JCL: CEXASUT

The following example shows typical JCL and commands used to request a report from the active session utility. You can tailor this JCL to produce active session reports in either mixed case or uppercase.

```
//CEXASUT  JOB (ACCOUNT),'NAME'
//STEP01 EXEC PGM=FUNEXEC,PARM='CEXJASKT,Y,L',REGION=0M
//STEPLIB  DD DSN=FUN.V2R4M0.SFUNLINK,DISP=SHR
//         DD DSN=CEX.V2R4M0.SCEXLINK,DISP=SHR
//SYSUDUMP DD SYSOUT=*
//MSGOUT   DD SYSOUT=*
//EVNTIN   DD DISP=SHR,DSN=journal.data.set
```

*Figure 107. JCL to run IMS Connect Extensions active session utility*

The *journal.data.set* refers to the Active or Archive Journal data set. If using an Active Journal, ensure the data set is not in use.

# Print utility

The print utility produces report output of formatted IMS Connect Extensions event records. It accepts input from an Active Journal or Archive Journal data set.

You can tailor the level of detail to print and the record selection range.

**Note:** You can only use an Active journal that is not being written to by IMS Connect Extensions.

## EXEC statement

The format of the EXEC statement is:

```
//stepname  EXEC  PGM=FUNEXEC,PARM='CEXDFPRT'
```

## SYSIN control cards

**PRINT**
Determines the output format and range of records to process. This control card is optional. If used it must be the first control card.

**RECORD**
Selects records based on event ID or event key. You cannot use this control card with the READ62 control card.

**READ62**
Selects records based on a wider range of criteria than the RECORD control card. You cannot use this control card with the RECORD control card.

If a control card is listed on more than one line, a continuation character must be placed in column 72.

## Sample JCL: CEXEVTP

The following example shows typical JCL and commands used to request a report from the print utility. You can tailor this JCL to produce multiple reports and extracts from multiple Active or Archive Journal data sets.

```
//CEXEVTP  JOB (ACCOUNT),'NAME'
//CEXEVTPR EXEC PGM=FUNEXEC,REGION=8M,
//             PARM='CEXDFPRT'
//STEPLIB  DD DISP=SHR,DSN=CEX.V2R4M0.SCEXLINK
//         DD DISP=SHR,DSN=FUN.V2R4M0.SFUNLINK
//EVNTIN   DD DISP=SHR,DSN=journal.data.set
//SYSIN    DD *
PRINT FORMAT=T
/*
//SYSUDUMP DD SYSOUT=*
//MSGOUT   DD SYSOUT=*
//*
```

*Figure 108. JCL to run the IMS Connect Extensions print utility*

## PRINT

The print utility **PRINT** command determines the output format and the range of records to process.

### Syntax

```
►►──PRINT──┬──────────────────┬──┬────────────────────────┬──────────────────────►
           │         ┌─T─┐    │  │        ┌─GMT────┐      │
           └─FORMAT=─┴─D─┴────┘  └─,ZONE=─┼─LOCAL──┼──────┘
                                          └─SERVER─┘

►──┬─┬───────────────────────┬──┬───────────────────────┬─┬──────────────────────►◄
   │ │          ┌─0─────┐    │  │          ┌─0─────┐    │ │
   │ └─,STAAFT=─┴─count─┴────┘  └─,STOAFT=─┴─count─┴────┘ │
   └─┬──────────────────────┬──┬──────────────────────┬───┘
     └─,STATIME=datetime────┘  └─,STOTIME=datetime────┘
```

### Keywords

**FORMAT**
Select one of the following:

**T** Summary formatting. This is the default value.

**D** Detail formatting

**ZONE** Select one of the following:

**GMT** Displays event times using the Greenwich Mean Time (GMT) offset. This is the default value.

**LOCAL**
Displays event times using the time offset of the machine executing the print utility.

**SERVER**
Displays event times using the time offset of the server that generated the event record.

**STAAFT**
Start processing records after a specified *count*.

**STOAFT**
Process the *count* number of records from the first record or the point defined by the STAAFT keyword.

**STATIME**
: Start processing records at a specified *datetime*. *datetime* is specified in the format *yyyy-mm-dd-hh:mm:ss.thmiju*.

**STOTIME**
: Stop processing records at a specified *datetime*. *datetime* is specified in the format *yyyy-mm-dd-hh:mm:ss.thmiju*.

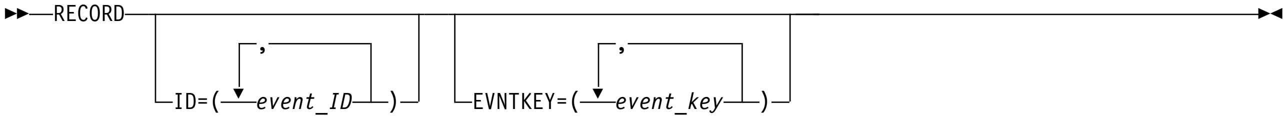
### PRINT control card example

```
PRINT FORMAT=D,                                                        +
STAAFT=2,STOAFT=10
```

## RECORD

The print utility **RECORD** command enables you to select records for the formatted output based on their ID and EVNTKEY. If you enter both ID and EVNTKEY keywords, found records will have to match one ID and one EVNTKEY.

### Syntax



### Keywords

**ID**
: Select records by event record ID. Entering more than one values will return those events matching any of the values.

**EVNTKEY**
: Select records by event record keys. Use either the 16-character event record key or the 4-character value of EVNT for Connect status event records. Entering more than one value will return those events matching any of the values specified.

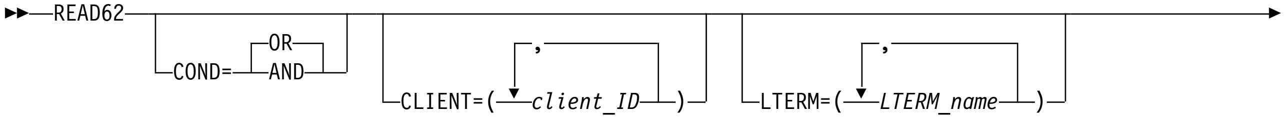
### RECORD control card example

```
RECORD  EVNTKEY=(0123456789ABCDEF,                                     +
      BD44160EB7B90300,                                                +
      BD44160C98C65201,                                                +
      EVNT)
```

## READ62

The print utility READ62 command enables you to specify conditions to select records for the formatted output.

### Syntax



```
►──┬────────────────────────────────────┬──┬──────────────────────────┬──┬──────────────────────────┬──►◄
   │          ┌─,────────────────┐      │  │        ┌─,─────────┐     │  ├─IPV4ADR──=──IP_address───┤
   │          ▼                  │      │  │        ▼           │     │  │                   (1)    │
   └─TRANCODE=(──transaction_type─┴──)──┘  └─EXIT=(──exit_name───┴──)─┘  └─IPV6ADR──=──IP_address───┘
```

**Notes:**

1 IPV4ADDR is a valid synonym for the IPV4ADR keyword. IPV6ADDR is a valid synonym for the IPV6ADR keyword.

### Keywords

**COND**
Select one of the following conditions:

**OR** Records are printed if the values in any of the keywords match.

**AND** Records are printed if they match at least one of the values in each keyword.

**CLIENT**
Select records by the client ID. Entering more than one value will return those events matching any of the values.

**LTERM**
Select records by the override LTERM name. Entering more than one value will return those events matching any of the values.

**TRANCODE**
Select records by the transaction code. Entering more than one value will return those events matching any of the values.

**EXIT** Select records by the exit name. Entering more than one value will return those events matching any of the values.

**IPV4ADR**
The IPV4 address of the remote client in dotted-decimal format (*n1.n2.n3.n4*).

**IPV6ADR**
The IPV6 address of the remote client in colon-hex format (*n1:n2:n3:n4:n5:n6:n7:n8*).

### READ62 control card example

```
READ62  TRANCODE(TRNBS001,TRNBS002)                              +
      EXIT=(HWSSMPL0)
```

## Transaction summary report

The amount of data shown on the Transaction summary report will vary depending on the collection level in place when the records were collected. Collection Level 4 will produce the most output.

The following fields are displayed on the transaction summary report:

**ID** Identifies the event record number. This is displayed in hexadecimal and decimal formats.

**V** Identifies the version of IMS Connect Extensions.

**TSKID**
Identifies the task in which the event occurred.

**TIME** The time when the event record was recorded.

**DELTA**
The time difference between the current selected event and the previous selected event.

When a single Event Key is specified on the RECORD control card, the DELTA field contains the elapsed time value between events for the same TCPIP session. When more than one Event Key is specified, the DELTA value might be calculated based on an event record from a previous session, in which case the value might not be meaningful.

**Key** The event key. This is either EVNT for Connect status event records or the token key for message-related event records.

**Description**
A short description of the event record.

```
ID=0A,010, V=12, TSKID=0105, KEY=EVNT            , BEGIN ACCEPT SOCKET
ID=3C,060, V=12, TSKID=0105, KEY=BAEBAA9E652E7C00, PREPARE READ SOCKET
ID=49,073, V=12, TSKID=0105, KEY=BAEBAA9E652E7C00, READ SOCKET
ID=49,073, V=12, TSKID=0105, KEY=BAEBAA9E652E7C00, READ SOCKET
ID=3D,061, V=12, TSKID=0105, KEY=BAEBAA9E652E7C00, MSG EXIT ENTERED
ID=3E,062, V=12, TSKID=0105, KEY=BAEBAA9E652E7C00, MSG EXIT RETURN
ID=41,065, V=12, TSKID=010E, KEY=BAEBAA9E652E7C00, MSG SENT TO OTMA
ID=42,066, V=12, TSKID=010E, KEY=BAEBAA9E652E7C00, MSG RCV FROM OTMA
ID=3D,061, V=12, TSKID=0105, KEY=BAEBAA9E652E7C00, MSG EXIT ENTERED
ID=3E,062, V=12, TSKID=0105, KEY=BAEBAA9E652E7C00, MSG EXIT RETURN
ID=4A,074, V=12, TSKID=0105, KEY=BAEBAA9E652E7C00, WRITE SOCKET
ID=49,073, V=12, TSKID=0105, KEY=BAEBAA9E652E7C00, READ SOCKET
ID=49,073, V=12, TSKID=0105, KEY=BAEBAA9E652E7C00, READ SOCKET
ID=49,073, V=12, TSKID=0105, KEY=BAEBAA9E652E7C00, READ SOCKET
ID=3D,061, V=12, TSKID=0105, KEY=BAEBAA9E652E7C00, MSG EXIT ENTERED
ID=3E,062, V=12, TSKID=0105, KEY=BAEBAA9E652E7C00, MSG EXIT RETURN
ID=41,065, V=12, TSKID=010E, KEY=BAEBAA9E652E7C00, MSG SENT TO OTMA
ID=48,072, V=12, TSKID=010E, KEY=BAEBAA9E652E7C00, TRIGGER EVENT
```

*Figure 109. Print Utility: Transaction summary report*

## Transaction detail report

The transaction detail report contains the same information as the transaction summary, plus some additional fields that are specific to the event type.

The amount of data shown on the Transaction detail report will vary depending on the collection level in place when the records were collected. Collection Level 4 will produce the most output.

The following fields are displayed on the transaction detail report:

**ID** Identifies the event record number. This is displayed in hexadecimal and decimal formats.

**V** Identifies the version of IMS Connect Extensions.

**TSKID**
Identifies the task in which the event occurred.

**TIME** The time when the event record was recorded.

**DELTA**
The time difference between the current selected event and the previous selected event.

When a single Event Key is specified on the RECORD control card, the DELTA field contains the elapsed time value between events for the same TCPIP session. When more than one Event Key is specified, the DELTA value might be calculated based on an event record from a previous session, in which case the value might not be meaningful.

**Key** The event key. This is either EVNT for Connect status event records or the token key for message-related event records.

**Description**
A short description of the event record.

**Variable Data**
Data specific to that event record.

The following fields are set for messages sent to OTMA (event X'41') and messages received back from OTMA (event X'42'):

**Type** The message type:

**DATA** A data message

**TRAN** A transaction

**RESP** A message response

**CMD** A Command

**COMT**
A Commit confirmed message

**RESP** The response type:

**ACK** A positive acknowledgment

**NACK**
A negative acknowledgment

**REQR** A response request.

**XRGR** An extended response requested

**COMT**
The response type received:

**CMTD**
Commit Confirmed received

**ABRT** Abort Confirmed received

**CRDY** Ready to Commit received

**CMD** The Command type:

**BID** Client Bid

**SVRA** Server is available

**RSYN** Resynchronization

**SIPA** Suspend input all tpipeS

**RIPA** Resume input all tpipeS

**SIPN** Suspend input for named tpipe

**RIPN** Resume input for named tpipe

**ROPN** Resume output for named tpipe

  - **ROPS** Resume output for named tpipes
  - **RCVR** Recover
  - **PREP** Prepare
  - **COMT** Commit
  - **RLBK** Rollback
  - **FORG** Forget

- **PROC** The processing type:
  - **L** Load Program
  - **T** Synchronized tpipe
  - **A** Async/Unsolicited queued messages
  - **E** Error message with NAK response
  - **Q** Message in Hold Queue
  - **S** SCI not present
- **Chain** The chain flag can have the following values:
  - **F** First In chain
  - **M** Middle in chain
  - **L** Last in chain
  - **C** Cancel Message

See “Mapping event records” on page 440 for a layout of every event record.

```
D=0B,011, V=12, TSKID=0105, KEY=EVNT            , END ACCEPT SOCKET
VAR DATA: APAR=0001, RC=0000000A, RSN=00000000
RECORD TIME=(DATE=2005.07.13, TIME=09.58.56.854472), LSN=0000000000000098

ID=0A,010, V=12, TSKID=0105, KEY=EVNT            , BEGIN ACCEPT SOCKET
TCPIB: LL=0028, BLKID=01, VVRR=0202, APAR=0001, PORT#=8801 , SOCKET#=0000
SOCKET FLAG=80, PORT FLAG=00, RMT REQ LEN=00000000, RMT ACT LEN=00000000
SPECIAL REQUEST DATA=00000000, RC=00000000, RMT RSN CODE=00000000
RECORD TIME=(DATE=2005.07.13, TIME=09.58.56.854564), LSN=0000000000000099

ID=3C,060, V=12, TSKID=0105, KEY=BD4D55966E1D7D63, PREPARE READ SOCKET
TCPIB: LL=0028, BLKID=01, VVRR=0202, APAR=0001, PORT#=8801 , SOCKET#=000A
SOCKET FLAG=40, PORT FLAG=00, RMT REQ LEN=00000020, RMT ACT LEN=00000001
SPECIAL REQUEST DATA=00000000, RC=00000001, RMT RSN CODE=00000000
RECORD TIME=(DATE=2005.07.13, TIME=09.58.56.854796), LSN=000000000000009A

ID=49,073, V=12, TSKID=0105, KEY=BD4D55966E1D7D63, READ SOCKET
TCPIB: LL=0028, BLKID=01, VVRR=0202, APAR=0001, PORT#=8801 , SOCKET#=000A
SOCKET FLAG=40, PORT FLAG=00, RMT REQ LEN=0000001F, RMT ACT LEN=0000001F
SPECIAL REQUEST DATA=00000000, RC=0000001F, RMT RSN CODE=00000000
RECORD TIME=(DATE=2005.07.13, TIME=09.58.57.136070), LSN=000000000000009B

ID=49,073, V=12, TSKID=0105, KEY=BD4D55966E1D7D63, READ SOCKET
TCPIB: LL=0028, BLKID=01, VVRR=0202, APAR=0001, PORT#=8801 , SOCKET#=000A
SOCKET FLAG=40, PORT FLAG=00, RMT REQ LEN=00000058, RMT ACT LEN=00000058
SPECIAL REQUEST DATA=00000000, RC=00000058, RMT RSN CODE=00000000
RECORD TIME=(DATE=2005.07.13, TIME=09.58.57.136147), LSN=000000000000009C

ID=3D,061, V=12, TSKID=0105, KEY=BD4D55966E1D7D63, MSG EXIT ENTERED
VAR DATA: APAR=0001, EXITN=HWSSMPL0
LEN=000A, APAR=0001, FUNC=READ, FLAG1=40, CONTENT=00
RECORD TIME=(DATE=2005.07.13, TIME=09.58.57.136179), LSN=000000000000009D

ID=3E,062, V=12, TSKID=0105, KEY=BD4D55966E1D7D63, MSG EXIT RETURN
VAR DATA: APAR=0001, RC=00000000, RSN=00000000, EXITN=HWSSMPL0
LEN=0028, APAR=0001, FUNCTION=READ, FLAG1=40, FLAG2=00, UFLAG1=00, CONTENT=98
EXPXXX RC=00000000, EXPXX RSN=00000000
IPV4: FAMILY=0002, CLIENT PORT=1313 , CLIENT IP ADDRESS=172.17.69.58
DATA LENGTH=00000234, CLIENT ID=TRCBS006
```

*Figure 110. Print Utility: Transaction detail report*

# RECORDER trace utilities

Two IMS Connect Extensions utilities can be used to process an IMS Connect-managed Recorder Trace data set (HWSRCDR). This data set stores information about the messages that are processed by IMS Connect.

**Recorder Trace conversion utility (CEXRTCNV)**
Converts the RECORDER trace data into the format of IMS Connect Extensions event records.

**Recorder Trace print utility (CEXRTPRT)**
Formats and prints the RECORDER trace event records created by the CEXRTCNV utility.

## Recorder Trace conversion utility (CEXRTCNV)

The CEXRTCNV utility converts RECORDER trace data into the format of IMS Connect Extensions event records.

## EXEC statement

The format of the EXEC statement is:

```
//stepname  EXEC  PGM=FUNEXEC,PARM=CEXRTCNV
```

This executes the RECORDER trace conversion utility.

## DD statements

The conversion utility supports the following DD statements:

**STEPLIB DD**
The IMS Connect Extensions product and support load libraries are required.

**RCDRIN DD**
The RCDRIN DD defines the RECORDER trace data set output from IMS Connect.

**Note:** The CEXRTCNV utility can only process an IMS Connect-managed Recorder Trace data set (HWSRCDR). CEXRTCNV cannot process IMS Connect trace information stored in BPE trace tables.

**RCDROUT DD**
The RCDROUT DD defines a sequential output data set containing the IMS Connect Extensions Event Records.

DCB attributes for this data set are DSORG=PS, RECFM=VB, LRECL=8192.

Each event record in the series is related via a common Token. Depending upon the RECORDER record content, three or more event records are created.

These event records (X'A0', X'A1', X'A2') contain RECORDER trace header data, IRM, OTMA, and output data.

This output data set can be printed using the print utility CEXRTPRT.

**MSGOUT DD**
Describes the output file.

It is usually defined as SYSOUT=* or SYSOUT=A.

**CEXPRINT DD**
The CEXPRINT DD defines the output file containing all IMS Connect Extensions messages. It is usually defined as SYSOUT=*. The usual DCB attributes for this data set are RECFM=VBA and LRECL=133.

## Sample JCL

```
//CEXRTCN  JOB (ACCOUNT),'NAME'
//STEP01   EXEC PGM=FUNEXEC,PARM=CEXRTCNV
//STEPLIB  DD DISP=SHR,DSN=CEX.V2R4M0.SCEXLINK
//         DD DISP=SHR,DSN=FUN.V2R4M0.SFUNLINK
//SYSUDUMP DD SYSOUT=*
//CEXPRINT DD SYSOUT=*
//MSGOUT   DD SYSOUT=*
//RCDRIN   DD DISP=SHR,DSN=your.recorder.input
//RCDROUT  DD DISP=SHR,DSN=your.recorder.output
/*
```

*Figure 111. JCL to run RECORDER trace conversion utility CEXRTCNV*

## Recorder Trace print utility (CEXRTPRT)

The CEXRTPRT utility formats and prints the RECORDER trace event records created by the CEXRTCNV utility.

### EXEC statement

The format of the EXEC statement is:

```
//stepname  EXEC  PGM=FUNEXEC,PARM=CEXRTPRT
```

This executes the print utility CEXRTPRT.

### DD statements

The print utility supports the following DD statements:

**STEPLIB DD**
The IMS Connect Extensions product and support load libraries are required.

**RCDRIN DD**
The RCDRIN DD defines a sequential output data set containing the IMS Connect Extensions Event Records.

DCB attributes for this data set are DSORG=PS, RECFM=VB, LRECL=8192.

**CEXPRINT DD**
The CEXPRINT DD defines the output file containing all IMS Connect Extensions messages. It is usually defined as SYSOUT=*. The usual DCB attributes for this data set are RECFM=VBA and LRECL=133.

**MSGOUT DD**
Describes the output file.

It is usually defined as SYSOUT=* or SYSOUT=A.

### Sample JCL

```
//CEXRTPR  JOB (ACCOUNT),'NAME'
//STEP01   EXEC PGM=FUNEXEC,PARM=CEXRTPRT
//STEPLIB  DD DISP=SHR,DSN=CEX.V2R4M0.SCEXLINK
//         DD DISP=SHR,DSN=FUN.V2R4M0.SFUNLINK
//SYSUDUMP DD SYSOUT=*
//CEXPRINT DD SYSOUT=*
//MSGOUT   DD SYSOUT=*
//RCDRIN   DD DISP=SHR,DSN=your.recorder.output
/*
```

*Figure 112. JCL to run RECORDER trace print utility CEXRTPRT*

## Sample output

```
FIRST RECORD DATE=2004.05.01, TIME=11.44.40
ID=A0,160, TOKEN=00000001
ITOC RECORD: LEN=0056 ,APAR=0001 ,ID=IT ,CONTENT=C0
SMF: LEN=0052 ,RECORD TYPE=77 ,TIME=11410582 ,SEQ#=0104122F
UOW: CLNT=TRAN0004 ,RCVT=BB26BB839516BC00 ,ENQT=BB26BB83D0512E40
DS 1ST MSG DQT=0000000000000000, DS CLR DQT=0000000000000000
ERR TIME=BB26BB83D0516E40, # MSG XMIT=0000, # MSG RCVD=0000
TIME=BB26BC2B3867F200, LSN=0000000000000001

ID=A1,161, TOKEN=00000001
ITOC REC: LEN=0010 ,APAR=0001 ,CON=80 ,CON1=02 ,TYPE=ITOC / RC / IPB
REMOTE CLIENT IRM: LLLL=X'00000071'
DUMP OF IRM HEADER FOR LENGTH=X'0058'
+0000  00580000 2A53414D 504C452A 00000000   *.......(&<......*
+0010  00000000 5452414E 30303034 00200020   *.......+........*
+0020  50415254 20202020 494D4433 20202020   *&........(......*
+0030  20202020 20202020 44565020 20202020   *..........&.....*
+0040  20202020 20202020 20202020 20202020   *................*
+0050  70656163 68202020                     *../.....        *
DUMP OF CLIENT MSG SEGMENT FOR LENGTH=X'0011'
+0000  00110000 50415254 20414E39 36304331   *....&.....+.....*
+0010  30                                    *.               *
DUMP OF CLIENT MSG SEGMENT FOR LENGTH=X'0004'
+0000  00040000                              *....            *

ID=A2,162, TOKEN=00000001
ITOC REC: LEN=0010 ,APAR=0001 ,CON=80 ,CON1=F8 ,TYPE=ITOC / RC / OPB
OTMA CTL TYPE=TRAN, RESP=NONE, COMT=NONE, CMD=NONE, PROC=        , CHAIN=FL
DUMP OF OTMA CONTROL HDR FOR LENGTH=X'0020'
+0000  01400000 00000000 00000000 0000A0F0   *. .............0*
+0010  00000000 00000000 00000000 00010000   *................*
DUMP OF OTMA STATE SECTION FOR LENGTH=X'0048'
+0000  00480020 00000000 00000000 00000000   *................*
+0010  00000000 00000000 00000000 00000000   *................*
+0020  00000000 00000000 00000000 00000000   *................*
+0030  00000000 00000000 00000000 00004040   *..............  *
+0040  40404040 40400000                     *      ..        *
DUMP OF OTMA SECURITY SECTION FOR LENGTH=X'006A'
+0000  006AC614 0902C4E5 D7404040 40400903   *..F...DVP     ..*
+0010  40404040 40404040 00000000 00000000   *        ........*
+0020  00000000 00000000 00000000 00000000   *................*
+0030  00000000 00000000 00000000 00000000   *................*
+0040  00000000 00000000 00000000 00000000   *................*
+0050  00000000 00000000 00000000 00000000   *................*
+0060  00000000 00000000 0000                *..........      *
DUMP OF OTMA USER SECTION FOR LENGTH=X'0100'
+0000  01000000 C9D4C4F3 40404040 00000000   *....IMD3    ....*
+0010  00000000 00000000 00000000 00000000   *................*
+0020  00000000 00000000 00000000 00000000   *................*
+0030  00000000 40404040 40404040 00000000   *....        ....*
+0040  00000000 97858183 88404040 00000000   *....PEACH   ....*
+0050  01000000 00000000 00000000 00000000   *................*
+0060  00000000 00000000 00000000 00000000   *................*
+0070  00000000 00000000 00000000 00000000   *................*
+0080  00000000 00000000 00000000 00000000   *................*
+0090  00000000 00000000 00000000 00000000   *................*
+00A0  00000000 00000000 00000000 00000000   *................*
+00B0  00000000 00000000 00000000 00000000   *................*
+00C0  00000000 00000000 00000000 00000000   *................*
+00D0  00000000 00000000 00000000 00000000   *................*
+00E0  00000000 00000000 00000000 00000000   *................*
+00F0  00000000 00000000 00000000 00000000   *................*
```

*Figure 113. RECORDER trace print utility: Sample output*

# Chapter 24. Archiving utilities

The archiving utilities allow you to archive active journals and to clean up old archives and records of old archives.

## Archive Journal cleanup utility

When an archive data set is created, IMS Connect Extensions enters its name in the repository. The archive journal cleanup utility deletes from the repository archive data sets that have expired.

The Archive Cleanup Utility uses the **Archive cleanup retention period** field (see Archive cleanup retention period) to determine whether to delete the name from the repository. Whether the archive data set is also deleted or uncataloged depends on its characteristics:

- The archive cleanup utility does not delete or uncatalog a GDG. Only the entry for the GDG in the repository is removed.
- A tape data set is uncataloged.
- A data set on DASD is deleted.

**Tip:** The completion message FUN2596I is displayed in the JESMSGLG. You can use this to build automated procedures to check the return codes for each system being cleaned up.

### Sample JCL: CEXCLEAN

The following example shows JCL to run the Archive cleanup utility:

```
//CEXCLEAN JOB (ACCOUNT),'NAME'
//STEP01   EXEC PGM=FUNEXEC,PARM=CEXJCLN,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=funpre.SFUNLINK
//         DD DISP=SHR,DSN=cexpre.SCEXLINK
//REPOSTRY DD DISP=SHR,DSN=your.def.file
//*
//ARCHCNTL DD *
CLEAN PRODUCT=CEX,LENGTH=8,NAME=(HWS1,HWS2),FUNCTION=R,CHECK=YES
/*
```

*Figure 114. JCL to run Archive Journal cleanup utility*

### EXEC statement

The format of the EXEC statement is:

```
//stepname  EXEC  PGM=FUNEXEC,PARM=CEXJCLN
```

### DD statements

The cleanup utility JCL must include the following DD statements:

**STEPLIB DD**
The IMS Connect Extensions product and functional support load libraries are required.

**REPOSTRY DD**

The REPOSTRY DD defines the IMS Connect Extensions Definitions data set for this IMS Connect system.

**ARCHCNTL DD**

The ARCHCNTL DD defines the control card for the utility. The syntax is:

```
CLEAN PRODUCT=CEX,LENGTH=8,NAME=(hwsid1,...hwsn),FUNCTION=L|R,   +
  CHECK=NO|YES,ZONE=GMT|LOCAL,DELERROR=N|Y
```

**LENGTH=8**

The length of the definition name. The value must be 8. If any of the system names listed in the following NAME parameter is less than 8 characters, the name is padded to be 8 characters.

**NAME=*(hwsid1,hws2,...hwsn)***

The names of one or more IMS Connect system definitions in the repository. Enter a system name of 1 - 8 characters or a comma-separated list of system names. For example, to clean systems named HWS1, HWS2, and HWS3, specify:

```
NAME=(HWS1,HWS2,HWS3)
```

If a single system is specified, the parentheses are optional, so NAME=HWS1 and NAME=(HWS1) are both valid.

**FUNCTION=L|R**

The acceptable values are:

**L** List mode. Prints to CEXPRINT a list of all archive journal records in the repository for the specified systems.

**R** Remove mode. If CHECK=NO is specified, remove any unused records and delete or uncatalog the archive data sets. If CHECK=YES is specified, print to CEXPRINT a list of the journal archive records that would have been removed.

**CHECK=NO|YES**

This parameter simulates or previews the effect of FUNCTION=R without removing any archive records. If CHECK=YES is specified, the CEXPRINT data set list will contain the journal archive records that would be deleted if CHECK=NO is specified.

This parameter can be used only with FUNCTION=R. The default value is CHECK=NO.

**ZONE=GMT|LOCAL**

This parameter specifies the time zone for the time stamp that is printed for each record in the journal archive record information. It can be used with FUNCTION=R or FUNCTION=L. If you specify ZONE=LOCAL, all time stamps are adjusted so that they show the local time of the computer that created the record. The default value is ZONE=GMT.

This parameter affects only the reported times in the body of messages FUN2570I and FUN2571I. The processing start time that is reported by FUN1003I is always shown as local time.

**DELERROR=N|Y**

If an event record is written with invalid data in the date field, the bad time stamp can create an archive entry with a time in the future. The basic CLEAN command assumes all event records have time stamps in the past, so it will not delete these error entries.

DELERROR=Y deletes any archive entry found whose ending time stamp value is higher than the current time. This parameter allows deletion of invalid entries caused by an invalid time stamp. For example:

```
CLEAN PRODUCT=CEX,FUNCTION=R,NAME=(HWS1),DELERROR=Y
```

# Archive Manager utility

The journal archive task submits archive jobs when the active journal is full or when a command is issued to switch it. You can create a JCL skeleton to generate an archiving job using CEXJARC, or you can manually submit a complete archive job.

## Sample JCL: CEXARCJ

Here is example JCL you could use to perform archiving manually. Note that when manually submitting JCL to run the Archive Manager utility:

- The ARCOUT DD statement must specify a literal data set name with no symbols ( 1 ).
- The REPOSTRY DD and the ARCHCNTL DD statements must be coded ( 2 ).

```
//CEXARCJ  JOB (ACCOUNT),'NAME'
//STEP01   EXEC PGM=FUNEXEC,PARM=CEXJARC,REGION=0M
//STEPLIB  DD DISP=SHR,DSN=funpre.SFUNLINK
//         DD DISP=SHR,DSN=cexpre.SCEXLINK
//ARCOUT   DD DISP=(,CATLG),DSN=archive.journal.data.set 1
//REPOSTRY DD DISP=SHR,DSN=definition.data.set 2
//ARCHCNTL DD *  2
CEX,08,HWS1
//*
```

*Figure 115. JCL to run the Archive Manager utility*

## EXEC statement

The format of the EXEC statement is:

```
►►—//stepname—EXEC—PGM=FUNEXEC,PARM='CEXJARC─┬──────────┬─'──────────────────►◄
                                            └─,WARNRC0─┘
```

There is one optional parameter:

**WARNRC0**
: Where a warning condition occurs that would normally return RC=4, coding this option causes the return code to be set to zero. This could be used to prevent an automated pager alert or other response being triggered when the archive utility reports a non-critical warning such as FUN2522W or FUN2533W.

## DD statements

**STEPLIB DD**
: The IMS Connect Extensions product and functional support load libraries.

**ARCOUT DD**
: The archive data set name and allocation attributes. If submitting a complete archive job, use the literal data set name.

If using the ARCOUT DD statement in an archive JCL skeleton, you have the following options:

- Use the symbol `%ARCDSN` to resolve a data set name using the Archive Journal **Data Set name** field specified in the system definition. For example:

  ```
  //ARCOUT   DD  DSN=%ARCDSN,
  //             DISP=(,CATLG,DELETE),
  //             SPACE=(CYL,(20,20),RLSE),
  //             LIKE=CEX240.ACTIVE.HWS1.P01
  ```

  This option is recommended because the **Data Set name** pattern is validated when it is saved in the ISPF dialog.
- Override the Archive Journal **Data Set name** field in the system definition by using a combination of literals and ampersand (&) prefixed symbols. For example:

  ```
  //ARCOUT   DD  DSN=USER.&ID..&DATE..&TIME,
  //             DISP=(,CATLG,DELETE),
  //             SPACE=(CYL,(20,20),RLSE),
  //             LIKE=CEX240.ACTIVE.HWS1.P01
  ```

  – You can use any of the symbols in Table 2 on page 62.

  – The archiving job will fail if the symbols and literals resolve to the name of an existing data set. For example, if you used &DATE without &TIME, a second archive job on the same day would fail.
- Override the Archive Journal **Data Set name** field in the system definition by using the name of an existing GDG base and "`(+1)`" to indicate a new generation. For example:

  ```
  //ARCOUT   DD  DSN=USER.MYGDG(+1),
  //             DISP=(,CATLG,DELETE),
  //             SPACE=(CYL,(20,20),RLSE),
  //             LIKE=CEX240.ACTIVE.HWS1.P01
  ```

  You can use the &ID symbol in a GDG name. This symbol resolves to the IMS Connect system name. A GDG base with the IMS Connect system name must exist for the job to complete successfully.

**REPOSTRY DD**

The IMS Connect Extensions repository. This data set is used to obtain control information. Do not use this statement in an archive JCL skeleton.

**ARCHCNTL DD**

The control card for the utility. Enter these options, separated by commas:

**Product code**

Use `CEX`.

**Length**

The length of the definition name. The value must be 08.

**Name** The System Definition name.

For example:

```
//ARCHCNTL DD  *
CEX,08,HWS1
/*
```

Do not use this statement in an archive JCL skeleton.

# Chapter 25. Installation verification programs (IVP)

The installation verification programs (IVP) generate test sessions to verify the installation of IMS Connect and IMS Connect Extensions. This section provides reference information that you can use to customize the sample IVP jobs or to write your own.

## IVP reference: OTMA workloads

The installation verification program (IVP) generates IMS Connect client sessions. The test cases send an OTMA transaction to IMS and receive the response from IMS Connect.

You can use the OTMA IVP to:
- Verify the installation of IMS Connect and IMS Connect Extensions.
- Do basic stress testing of IMS Connect.
- Test IMS Connect and IMS Connect Extensions security.

The sample IVP jobs CEXIVP1 to CEXIVP9 are supplied in the SCEXSAMP library. They execute the corresponding test cases CEXCSE01 to CEXCSE09. Follow the instructions that are contained in each member to replace the placeholders in the sample job with the appropriate values for your system. After you have tailored each sample, submit it.

### EXEC statement

The format of the EXEC statement is:

```
//stepname  EXEC PGM=FUNEXEC,PARM='CEXDFIVP'
```

### DD statements

**MSGOUT**
Contains a report about the execution of the IVP. This includes a formatted view of data returned from IMS or the service exit.

**CEXPRINT**
Contains error and debug information about the IVP execution. Refer to this data set to troubleshoot any execution problems.

**SYSIN DD**
Data set containing the utility control cards.

## CASE control card

The CASE IVP statement selects an IVP test case.

```
                                  ┌─,CTR=0────────────┐
►►──CASE──NAME=test_case_name─────┤                   ├──────────────────────────►◄
                                  └─,CTR=1 - 99999999─┘
```

## Parameters

**NAME**

The name of the test case. You must use one of the test case names from the following table:

*Table 7. Available IVP test cases*

| Name | Commit mode | SYNCLEVEL | Transaction | Exit | Persistent socket? |
|---|---|---|---|---|---|
| CEXCSE01 | 1 | None | PART | HWSSMPL1 | |
| CEXCSE02 | 1 | Confirm | PART | HWSSMPL1 | |
| CEXCSE03 | 1 | None | PART | HWSSMPL1 | Yes |
| CEXCSE04 | 1 | Confirm | PART | HWSSMPL1 | Yes |
| CEXCSE05 | 1 | Confirm | N/A | CEXSVC01 | |
| CEXCSE06 | 1 | None | PART | HWSSMPL0 | |
| CEXCSE07 | 1 | Confirm | PART | HWSSMPL0 | |
| CEXCSE08 | 1 | None | PART | HWSSMPL0 | Yes |
| CEXCSE09 | 1 | Confirm | PART | HWSSMPL0 | Yes |

**CTR** The number of times to rerun the test case. CTR=0, the default, means the test case is not rerun. A value of 1 means the test case is rerun once (so the test case will run twice in total). The maximum value is 99999999.

# HOST control card

The HOST IVP statement specifies the host and port for IMS Connect.

```
►►──HOST──┬─DNSNAME=DNS_name─────┬──PORT=ICON_port──────────────────────────►
          ├─IPV4ADR=IPv4_address─┤
          └─IPV6ADR=IPv6_address─┘

►─HWSID=IMS_Connect_system──────────────────────────────────────────────────►◄
```

## Parameters

**DNSNAME**

The name of the host for the IMS Connect system. For example, `localhost`.

**IPV4ADR**

The IPv4 address of the host.

**IPV6ADR**

The IPv6 address of the host.

**PORT** A TCP/IP port IMS Connect is listening on.

**HWSID**

The name of the IMS Connect system.

# IRM control card

The IRM IVP statement specifies values for fields in the IRM.

```
►►──IRM──DESTID=destid_for_target_ds──────────────────────────────────────────────────────►

►─┬──────────────────────────────────────────────────────────────────────────────────────┬─►
  │                                                                                 (1)   │
  └─,USERID=RACF_userid─,PASSWORD=─┬─RACF_password─┬─,GROUP=RACF_group─,APPLID=appl_name───┘
                                   └─PASSTIK───────┘

►─┬───────────────────────────┬──┬──────────────────────┬──┬───────────────────┬───────────►◄
  └─,LTERM=logical_terminal───┘  └─,CLIENT=client_name──┘  │          ┌─E9───┐ │
                                                           └─,TIMER=──┴─flag─┴─┘
```

**Notes:**

1 Specify security fields if IMS Connect or IMS Connect Extensions security is active.

## Parameters

**DESTID**
The destination ID of the target datastore.

**USERID**
RACF user ID.

**PASSWORD**
The RACF password or PassTicket (“PASSTIK”) for the user ID. The PASSWORD parameter is required in the following cases:

- If IMS Connect security is enabled (RACF=Y is specified in the IMS Connect configuration file).
- If IMS Connect Extensions security is enabled in the system definition.

**GROUP**
RACF group.

**APPLID**
The appl name.

**LTERM**
Logical terminal name.

**CLIENT**
The client name.

**TIMER**
IRM timer flag field value.

# SEND control card

The SEND IVP statement specifies how the transaction data sent to IMS Connect is printed by the IVP.

```
            ┌─N─┐
►►──SEND──,PRINT=─┼─Y─┼──────────────────────────────────────────────────►◄
            └─A─┘
```

### Parameters

**PRINT**

Print messages in one of the following formats:

**N** Do not print (default).

**Y** Dump format.

**A** EBCDIC and ASCII dump format.

## RECEIVE control card

The RECEIVE IVP statement specifies how the transaction data received from IMS Connect is printed by the IVP.

```
               ┌─N─┐
►►──RECEIVE──,PRINT=─┼─Y─┼───────────────────────────────────────────────►◄
               ├─F─┤
               └─A─┘
```

### Parameters

**PRINT**

Print messages in one of the following formats:

**N** Do not print (default).

**F** Character format.

**Y** Dump format.

**A** EBCDIC and ASCII dump format.

# IVP reference: Open Database workloads

The installation verification programs (IVP) generate IMS Connect client sessions. The test cases send DRDA requests to an IMS sample database and receive the response from IMS Connect.

The sample IVP jobs CEXIVPO1 and CEXIVPO2 are supplied in the SCEXSAMP library.

These IVP samples use a REXX program named CEXIVPOD which drives the DRDA protocol. CEXIVPOD formats all the DRDA requests and interprets and formats the DRDA replies. It also establishes the TCP/IP environment and maintains the TCP/IP sockets.

## EXEC statement

The format of the EXEC statement is:

```
//stepname  EXEC PGM=IRXJCL,PARM='CEXIVPOD'
```

## DD statements

**SYSTSPRT**
Contains a report about the execution of the IVP. This includes a formatted view of data returned from IMS.

**SYSTSIN DD**
Data set containing the DRDA command sequences.

## Parameter validation

This following table lists the variables that must be set before running either of the sample IVP jobs. If validation fails for any reason the return code is set to 12 and the program terminates.

| Variable | Purpose | Default value | Validation performed |
|---|---|---|---|
| IPVers | Internet Protocol version | IPV4 | IPV4 or IPV6 |
| IPAddr | DNS name or IP address of the IMS Connect system | none | 1. Must be set to a DNS name or an IP address.<br>2. Whether the IP address is valid will be determined when it is used to establish a connection. |
| Port | IMS Connect DRDA port | none | Port > 0 & Port < 65536 |
| USRID | An authorized user ID | none | length(USRID) < 9 |
| PASSWORD | The RACF password or PassTicket ("PASSTIK") for the user ID | none | length(PASSWORD) < 9<br>The PASSWORD parameter is required in the following cases:<br>• If IMS Connect security is enabled (RACF=Y is specified in the IMS Connect configuration file).<br>• If IMS Connect Extensions security is enabled in the system definition.<br>Otherwise, PASSWORD is optional. |
| ODAlias | (Optional) The datastore alias for the target ODBM | none | length(ODAlias) < 5 |
| SRVNAM | (Optional) The server name | MVS1 | none |

# Chapter 26. CEXPAPI: Publish to an external application

The publisher API is an assembler API that publishes IMS Connect Extensions and IMS Connect data.

This section documents intended Programming Interfaces that allow the customer to write programs to obtain the services of IMS Connect Extensions.

Clients can access the following types of data through the publisher API:

- IMS Connect Extensions event data
- IMS Connect systems that use IMS Connect Extensions
- Many of the tasks in an IMS Connect region
- Active sessions
- IMS Connect configuration data
- Datastores and their status (active or inactive)

With this information you can highlight delays in session processing, show resource utilization for IMS Connect tasks, and build records for analysis.

The publisher API is implemented by the CEXPAPI macro, which is in the USERMAC library. It provides the following functions:

**INIT** Initializes the CEXPAPI interface.

**QUERY**
Lists all eligible IMS Connect systems. An eligible IMS Connect system is one that has activated the IMS Connect Extensions publisher API.

**CONNECT**
Connects the client region to the target IMS Connect region.

**READSYNC**
Lists tasks and active sessions for an IMS Connect system.

**STARTMON**
Starts a process that collects event records from IMS Connect Extensions.

**READASYN**
Receives IMS Connect Extensions event data by reading from the buffer created by the STARTMON function and resets this buffer.

**STOPMON**
Stops the process started by the STARTMON function.

**DISCONN**
Disconnects the client region from IMS Connect.

**TERM** Terminates the CEXPAPI interface.

**ETERM**
Terminates the CEXPAPI interface and discards all other pending functions.

The following figure shows the relationship between these functions.

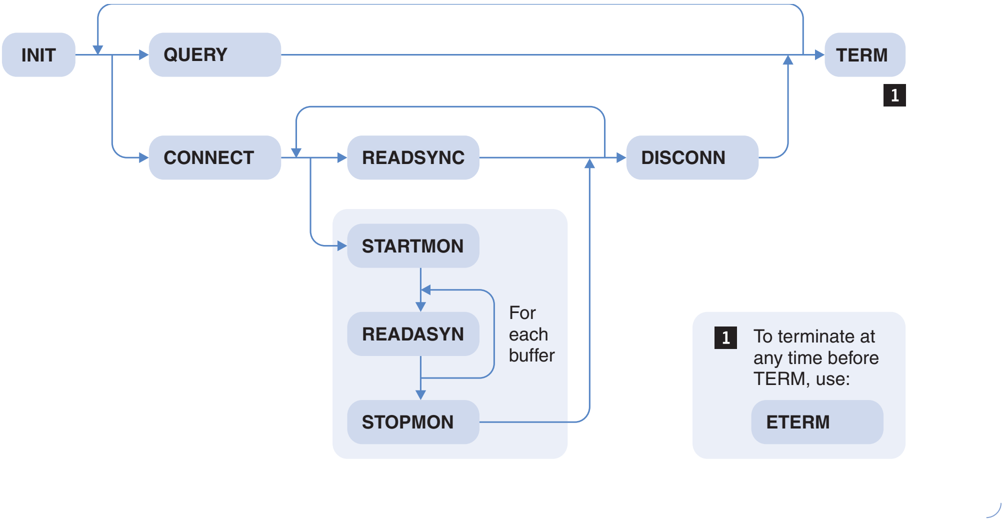


*Figure 116. The relationship between the functions of the CEXPAPI macro*

## Programming requirements and environment

This topic provides reference information to help you develop and run clients using the publisher API.

- Generate the equates needed by the publisher API using the CEXAPEQU macro, which is in the USERMAC library.
- To issue READSYNC requests, you must generate the request control block (RQB) DSECT using the following statement:

  ```
  CEXAPRQB REQ=DSECT
  ```

  Or, if you want to use the RQB in working storage, use the following statement instead:

  ```
  CEXAPRQB REQ=AREA
  ```

- It is recommended that you establish an extended specify task abnormal exit (ESTAE) routine prior to connecting to the server.
- Your ESTAE routine must issue the ETERM function if it cannot recover. The ETERM function protects the client address space and ensures that clients can connect to the server when the address space is restarted.
- The client interface requires approximately 50 kilobytes of virtual storage within the client's private storage. You can free this storage using the TERM function.
- The ECSA cleanup utility requires the system to assign a protection key of 7. Therefore you must add a PPT entry for CEXAPIMN to your system configuration. For a sample, see CEXPPT in the SCEXSAMP library.

### Environment

**Minimum authorization**
Authorized state, any PSW key

**Dispatchable unit mode**
Task

**Cross memory mode**
None (PASN=HASN=SASN)

**AMODE**
31-bit

**ASC mode**
Primary

**Interrupt status**
Enabled

**Locks** None required

## Termination procedure

How you terminate the client depends on why you need it to terminate:

**You want to initiate a termination (normal termination)**
Do the following:

1. If monitoring is active (STARTMON), issue the STOPMON function.
2. Issue the DISCONN function.

   This disconnects the client address space from the server address space and releases cross memory resources held in the client address space.
3. Issue the TERM function.

   This releases virtual storage and other resources used by the API management routine.

**The server initiates a termination**
What you do depends on how the server initiates termination:

- If the server posts the termination ECB (TRMECB), issue the ETERM function.
- If you issued a READASYN function and the server replies with return code 4 and reason code IIRSN_SVRCLS (server closing) or IIRSN_CLSCMD (stop client), perform a normal termination (STOPMON, DISCONN, TERM).

**Your ESTAE routine gets control because of a region ABEND**
If you cannot recover, issue the ETERM function.

## Recovery and error handling

The CEXPAPI macro sets its own ESTAE. If, at the time of entry, the ESTAE finds that the API is processing a request, it issues the ETERM function on behalf of the client. After issuing the ETERM function, the ESTAE passes control to the interface recovery routine so that the client can receive return and reason codes indicating the error and the ETERM processing that has occurred. You can choose to issue the TERM function to free virtual storage resources associated with the current session.

If, at the time of entry, CEXPAPI finds that the API is not processing a request, it percolates to the client ESTAE.

## Restrictions and limitations

- Include the IMS Connect Extensions SCEXLINK data set as a STEPLIB data set in your program's startup JCL.

  Or:

  Place the module CEXMAPI in Pageable Link Pack Area (PLPA) or any other LOAD library available to your program.
- Each MVS task must establish its own environment using the INIT function.

- An MVS task can be connected to only one IMS Connect region at a time, but other tasks in the same region can connect to different IMS Connect regions.

## Security

The client application must run as an authorized task. No additional security checking is performed for the client once connected to an IMS Connect region.

## Input register information

The general purpose register, R13, must point to the standard 18-word save area. Registers 0, 1, 14, 15 are used as work registers. You do not have to place anything into any other register, unless specifying parameters in register notation.

## Output register information

When control returns to the caller, the general purpose registers (GPR) are:

**0** Contains the reason code when the return code is non-zero. Otherwise used as a work register where contents are unpredictable.

**1** If non-zero, contains the address of the request feedback block. For the INIT function, contains the address of the token.

**2-13** Contents are unchanged.

**14** Work register.

**15** Return code.

## Reason and return codes

See the member CEXAPEQU for reason and return codes and their explanation.

## Performance implications

- A publisher API client can prevent IMS Connect from shutting down. Your program must respond correctly when the TRMECB is posted to allow IMS Connect to terminate.
- To ensure that the client can issue the READASYN request before the server buffer is full, you may need to adjust the MVS priority of the job.
- The event collection level set by the STARTMON function affects the number of event records passed to the client and thus the resources required by the client for processing.

## List form syntax

```
►►—label—CEXPAPI—MF=L———————————————————————————————————————————►◄
```

## Samples

These sample members in the SCEXSAMP library will help you create an API client and then assemble and execute it:

**CEXSPAPI**
Sample publisher API client.

**CEXAPAPI**
Assembles and links the CEXSPAPI program.

**CEXXPAPI**
Executes CEXSPAPI.

## Mapping macros

The mapping macros for the publisher API are in the USERMAC library.

**CEXAPEQU**
Defines the equates for the publisher API interface.

**CEXAPIBH**
Defines the client buffer header. The header must be mapped by this macro to receive data from the interface.

**CEXAPIRL**
The returned buffer list header. This header provides information on the buffer and its content.

**CEXAPCLE**
Maps IMS Connect configuration data returned by the **READSYNC** command.

**CEXAPDLE**
Maps IMS Connect datastore list returned by the **READSYNC** command.

**CEXAPQLE**
Maps buffer element list for active IMS Connect systems from the QUERY request.

**CEXAPRQB**
Provides the data type and filtering information for the READSYNC request.

**CEXAPTLE**
Maps buffer element list for IMS Connect task information from the READSYNC request.

**CEXCSKT**
Maps buffer element list for active IMS Connect sessions from the READSYNC request.

**CEXIICB**
Maps the token that is returned by the INIT function.

**CEXCERE**
Maps returned IMS Connect event records. This macro has a different syntax from other macros in this list. See “Mapping event records” on page 440 for details.

Each mapping macro has the following syntax:

```
►►—label—macro_name—REQ=—┬DSECT┬—,PFX=prefix————————————————————►◄
                         └AREA─┘
```

These are the keywords:

**REQ=AREA | DSECT**
(Required) One of the following:

**DSECT**
Assembler mapping.

**AREA** Assembler storage area generation.

**PFX=***prefix*
: (Optional) The prefix for all labels generated with the macro expansion. Each mapping macro has a default expansion.

## CEXPAPI functions

The topics in this section list all CEXPAPI functions, their purpose, syntax, keywords, and specific return and reason codes.

### CONNECT

The CONNECT function connects the client region and the target IMS Connect region. The function selects a target IMS Connect system and establishes the buffer size for the READSYNC function.

#### Format

```
►►──label──CEXPAPI──CONNECT──,AGENT=agent──,AUTHID=authorization_ID──────────────►

►─,BUFSIZE=READSYNC_buffer_size──,TARGET=target_hws──,TOKEN=token────────────────►

►─,TRMECB=terminate_ECB──┬──────────────────────┬──────────────────────────────►◄
                         └─,MF=(E─,parm_area─)─┘
```

#### Parameters

**AGENT=***address* **|** (*r2-r12*)
: (Required) The address of the 16-byte agent name of the client. The agent name is included in messages produced by the server. It must be unique for each CONNECT request.

**AUTHID=***address* **|** (*r2-r12*)
: (Required) The address of the 8-byte authorization ID of the client. The authorization ID is not currently in use.

**BUFSIZE=***address* **|** (*r2-r12*)
: (Required, must be non-zero) The address of the fullword server buffer size. The buffer is used for READSYNC requests only.

**TARGET=***address* **|** (*r2-r12*)
: (Required) The address of the 8-byte HWSID of the target IMS Connect system. All subsequent calls: STARTMON, READASYN, STOPMON, READSYNC or DISCONN, are performed with the *target* IMS Connect system.

**TOKEN=***address* **|** (*r2-r12*)
: (Required) The address of a full-word containing the token address. The token is returned by the INIT function and must be passed in all subsequent API requests.

**TRMECB=***address* **|** (*r2-r12*)
: (Required) The address of a fullword containing the termination ECB. This ECB is posted if the server terminates while the client is connected.

  Constantly monitor this ECB and perform an immediate shutdown when this ECB is posted. See "Termination procedure" on page 347 for details.

**MF=(E,***address* **|** (*r1-r12*))
: (Required if reentrant) The address of the parameter list area used by your program. This parameter list must be constructed in modifiable storage by

calling the list form of the CEXPAPI macro using the MF=L parameter. See "List form syntax" on page 348 for details.

```
*-------------------------------------------------------------------*
    Connect to the target IMS Connect region
*-------------------------------------------------------------------*

         CEXPAPI CONNECT,                                            +
               TOKEN=@TOKEN,                                         +
               AUTHID=AUTHID,                                        +
               AGENT=CLIENTID,                                       +
               TARGET=HWSID,                                         +
               BUFSIZE=BUFSIZE,                                      +
               TRMECB=@TERMECB,                                      +
               MF=(E,PARMLST)
         LTR   R15,R15                 Successful?
         BNZ   ...                     No, error

@TOKEN   DS    A                       Address of token
@TERMECB DC    A(TERMECB)              Address of API termination ECB

AUTHID   DC    CL8' '                  Authorization ID
CLIENTID DC    CL16'SAMPLE CLIENT'     Client ID
HWSID    DC    CL8'HWS1'               IMS Connect System ID
BUFSIZE  DC    F'32768'                Buffer size
TERMECB  DC    F'0'                    API termination ECB
```

## DISCONN

The DISCONN function disconnects the client region from the target IMS Connect region.

### Format

```
►►──label──CEXPAPI──DISCONN──,TOKEN=token──┬──────────────────────┬──────────►◄
                                           └─,MF=(E─,parm_area─)──┘
```

### Parameters

**TOKEN=***address* | (*r2*-*r12*)
: (Required) The address of a full-word containing the token address. The token is returned by the INIT function and must be passed in all subsequent API requests.

**MF=**(**E**,*address* | (*r1*-*r12*))
: (Required if reentrant) The address of the parameter list area used by your program. This parameter list must be constructed in modifiable storage by calling the list form of the CEXPAPI macro using the MF=L parameter. See "List form syntax" on page 348 for details.

### Example

```
*-------------------------------------------------------------------*
    Disconnect client region from IMS Connect
*-------------------------------------------------------------------*

         CEXPAPI DISCONN,                                            +
               TOKEN=@TOKEN,                                         +
               MF=(E,PARMLST)
         LTR   R15,R15                 Successful?
         BNZ   ...                     No, error

@TOKEN   DS    A                       Address of token
```

## ETERM

The ETERM function terminates the CEXPAPI interface and discards all other pending functions. The request disconnects the client from the server and releases all cross memory resources.

Use this function if the remote client task becomes unusable and you need to perform an unconditional termination. This is any condition in which the ESTAE routine would get control.

### Format

```
►►—label—CEXPAPI—ETERM—,TOKEN=token——┬————————————————————┬——————————————————►◄
                                     └—,MF=(E—,parm_area—)┘
```

### Parameters

**TOKEN=***address* **|** (*r2-r12*)
: (Required) The address of a full-word containing the token address. The token is returned by the INIT function and must be passed in all subsequent API requests.

**MF=(E,***address* **|** (*r1-r12*))
: (Required if reentrant) The address of the parameter list area used by your program. This parameter list must be constructed in modifiable storage by calling the list form of the CEXPAPI macro using the MF=L parameter. See "List form syntax" on page 348 for details.

```
*----------------------------------------------------------------------*
    Terminate CEXPAPI interface and discard all pending functions
*----------------------------------------------------------------------*

         CEXPAPI ETERM,                                               +
               TOKEN=@TOKEN,                                          +
               MF=(E,PARMLST)
         LTR   R15,R15                  Successful?
         BNZ   ...                      No, error

@TOKEN   DS    A                        Address of token
```

## INIT

The INIT function initializes the CEXPAPI interface, allowing you to perform CEXPAPI functions.

After a successful INIT request, register 1 contains the address of the task token. This token address must be returned with all subsequent requests. The token cannot be shared between different MVS tasks.

### Format

```
►►—label—CEXPAPI—INIT——┬————————————————————┬——————————————————————————————►◄
                       └—,MF=(E—,parm_area—)┘
```

### Parameters

**MF=(E,***address* **|** (*r1-r12*))
: (Required if reentrant) The address of the parameter list area used by your program. This parameter list must be constructed in modifiable storage by

calling the list form of the CEXPAPI macro using the MF=L parameter. See “List form syntax” on page 348 for details.

```
*--------------------------------------------------------------------*
    Initialize CEXPAPI interface
*--------------------------------------------------------------------*

         CEXPAPI INIT,                                                 +
               MF=(E,PARMLST)
         LTR   R15,R15                 Successful?
         BNZ   ...                     No, error
                                       After INIT,
         ST    R1,@TOKEN               R1 contains address of token

PAPILST  CEXPAPI MF=L                  Parameter list

@TOKEN   DS    A                       Address of token
```

## QUERY

The QUERY function lists all eligible IMS Connect systems. An eligible IMS Connect system is one that has activated the IMS Connect Extensions publisher API option in the System Definition.

### Format

```
►►──label──CEXPAPI──QUERY──,BUFFER=client_buffer──,TOKEN=token──────────────►

►──┬──────────────────────────┬──────────────────────────────────────────────►◄
   └─,MF=(E─,parm_area─)─┘
```

### Parameters

**BUFFER=***address* | (*r2*-*r12*)
: (Required) The address of a full-word containing the client buffer address. The buffer must contain a buffer header mapped by the macro CEXAPIBH.

**TOKEN=***address* | (*r2*-*r12*)
: (Required) The address of a full-word containing the token address. The token is returned by the INIT function and must be passed in all subsequent API requests.

**MF=(E,***address* | (*r1*-*r12*))
: (Required if reentrant) The address of the parameter list area used by your program. This parameter list must be constructed in modifiable storage by calling the list form of the CEXPAPI macro using the MF=L parameter. See “List form syntax” on page 348 for details.

```
*--------------------------------------------------------------------*
    List all eligible IMS Connect systems
*--------------------------------------------------------------------*

         CEXPAPI QUERY,                                                +
               TOKEN=@TOKEN,                                           +
               BUFFER=@BUFFER,                                         +
               MF=(E,PARMLST)
         LTR   R15,R15                 Successful?
         BNZ   ...                     No, error

@TOKEN   DS    A                       Address of token
@BUFFER  DS    A                       Address of buffer

BUFFER   DSECT ,                       Buffer
```

## READASYN

The READASYN function reads IMS Connect Extensions event data. The function does this by transferring data from the buffer created by the STARTMON function to its own buffer.

To use the READASYN function:

1. Issue a **STARTMON** command, creating the ECB for the server buffer threshold.

   When the server buffer size reaches the threshold, the ECB defined by the ECBADDR keyword of the STARTMON function is posted.
2. Issue a READASYN function to transfer the server buffer to your client buffer area.

   This clears the server buffer.
3. Clear the client buffer and wait for the next server buffer threshold.

**Note:**

1. You can issue a READASYN request at any time, even before the ECB is posted.
2. If this function gives a return code 4 reason code IIRSN_SVRCLS (server closing) or IIRSN_CLSCMD (stop client), perform an immediate shut down. See “Termination procedure” on page 347 for details.
3. During periods of low activity, you can implement a timer to issue the READASYN function at predefined intervals .
4. It is recommended that you use the MVS SPOST macro (or similar) to ensure a post is not pending to the ECB prior to issuing the READASYN function.

### Format

```
►►──label──CEXPAPI──READASYN──,BUFFER=client_buffer──,TOKEN=token──────────────►

►─┬───────────────────────┬──────────────────────────────────────────────────►◄
  └─,MF=(E─,parm_area─)─┘
```

### Parameters

**BUFFER=***address* | (*r2-r12*)
(Required) The address of a full-word containing the client buffer address. The buffer must contain a buffer header mapped by the macro CEXAPIBH.

**TOKEN=***address* | (*r2-r12*)
(Required) The address of a full-word containing the token address. The token is returned by the INIT function and must be passed in all subsequent API requests.

**MF=(E,***address* | (*r1-r12*))
(Required if reentrant) The address of the parameter list area used by your program. This parameter list must be constructed in modifiable storage by calling the list form of the CEXPAPI macro using the MF=L parameter. See “List form syntax” on page 348 for details.

```
*-----------------------------------------------------------------*
    Read IMS Connect Extensions event data into client buffer
*-----------------------------------------------------------------*

         CEXPAPI READASYN,                                         +
               TOKEN=@TOKEN,                                       +
               BUFFER=@BUFFER,                                     +
               MF=(E,PARMLST)
         LTR   R15,R15                Successful?
         BNZ   ...                    No, error

@TOKEN   DS    A                      Address of token
@BUFFER  DS    A                      Address of buffer

BUFFER   DSECT ,                      Buffer
```

## READSYNC

The READSYNC function lists information about an IMS Connect system.

### Purpose

READSYNC lists the following type of information:

- Tasks associated with an IMS Connect system.
- IMS Connect active tasks listening on ports.
- Active sessions for an IMS Connect system.
- Datastores for an IMS Connect system.
- IMS Connect configuration data.

The type of information returned by this function depends on contents of the request control block (RQB). Table 8 summarizes the settings for the RQB and the corresponding elements that are returned.

*Table 8. RQB requests*

| Request | RQB_PRIREQ | RQB_SECREQ | RQB_FILTER | Mapping |
|---|---|---|---|---|
| All tasks | IIPRI_TASK | IISEC_ALL | IIFLT_NOFILTER | CEXAPTLE |
| Port tasks only | IIPRI_TASK | IISEC_PORT | IIFLT_NOFILTER | CEXAPTLE |
| All datastores | IIPRI_DSTORE | IISEC_ALL | IIFLT_NOFILTER | CEXAPDLE |
| Active datastores only | IIPRI_DSTORE | IISEC_ALL | IIFLT_ACTIVE | CEXAPDLE |
| All active sessions | IIPRI_SESSIONS | IISEC_ALL | IIFLT_NOFILTER | CEXCSKT |
| Active sessions that are not in Read Prepare state | IIPRI_SESSIONS | IISEC_ALL | IIFLT_NOTWAIT | CEXCSKT |
| Configuration member | IIPRI_CONFIG | IISEC_ALL | IIFLT_NOFILTER | CEXAPCLE |

### Format

```
►►—label—CEXPAPI—READSYNC—,BUFFER=client_buffer——————————————————————►

►—,RQB=request_control_block—,TOKEN=token—┬——————————————————————┬——————►◄
                                          └—,MF=(E—,parm_area—)—┘
```

### Parameters

**BUFFER=***address* **|** (*r2-r12*)
(Required) The address of a full-word containing the client buffer address. The buffer must contain a buffer header mapped by the macro CEXAPIBH.

**RQB=***address* **|** (*r2-r12*)
(Required) The address of the fullword containing the request control block (RQB). The RQB defines what type of information is retrieved from the server.

**TOKEN=***address* **|** (*r2-r12*)
(Required) The address of a full-word containing the token address. The token is returned by the INIT function and must be passed in all subsequent API requests.

**MF=(E,***address* **|** (*r1-r12*))
(Required if reentrant) The address of the parameter list area used by your program. This parameter list must be constructed in modifiable storage by calling the list form of the CEXPAPI macro using the MF=L parameter. See "List form syntax" on page 348 for details.

```
*--------------------------------------------------------------------*
    List tasks and active sessions for the IMS Connect system
*--------------------------------------------------------------------*

         CEXPAPI READSYNC,                                              +
               TOKEN=@TOKEN,                                            +
               BUFFER=@BUFFER,                                          +
               RQB=@RQB,                                                +
               MF=(E,PARMLST)
         LTR   R15,R15                  Successful?
         BNZ   ...                      No, error

@TOKEN   DS    A                        Address of token
@BUFFER  DS    A                        Address of buffer
@RQB     DC    A(RQB)                   Address of request feedback

RQB      CEXAPRQB REQ=AREA,PFX=RQB      API request block

BUFFER   DSECT ,                        Buffer
```

# STARTMON

The STARTMON function starts the process of collecting event records from IMS Connect Extensions.

Records are accumulated on the server until you issue a READASYN call, which transfers the server buffer to the client and then resets the server buffer. If the READASYN is not received by the time the buffer is full, event records are discarded until the buffer is cleared. If records were discarded, it is indicated in the response your program receives to the READASYN request.

## Format

```
►►—label—CEXPAPI—STARTMON—,BUFSIZE=server_buffer_size———————————————————►

►—,ECBADDR=ECB_address—,LEVEL=event_collection_level————————————————————►

►—,THRESH=server_buffer_threshold—,TOKEN=token—┬——————————————————┬——————►◄
                                               └,MF=(E—,parm_area—)┘
```

## Parameters

**BUFSIZE=***address* **|** (*r2-r12*)
(Required) The address of the fullword server buffer size. The buffer is only used for READASYN requests.

**ECBADDR=***address* **|** (*r2-r12*)
(Required) The address of the fullword containing the monitoring ECB. The monitoring ECB is posted when the buffer size threshold (see the THRESH

keyword) is reached. The ECB must be in the same storage key as the client's primary key and addressable in the 'Home' address space.

**LEVEL=***address* **|** (*r2*-*r12*)
(Required) The address of a 4-byte field containing the collection level. The collection level determines what sort of events are spooled. See Chapter 4, "Event collection," on page 55 for details on collection levels and event records.

**THRESH=***address* **|** (*r2*-*r12*)
(Required) The address of a 4-byte field containing the buffer full threshold. The value is expressed as a percentage. For example, a value of 70 means that the ECB is posted when the server buffer has reached 70 percent of the size specified with the BUFSIZE keyword.

**TOKEN=***address* **|** (*r2*-*r12*)
(Required) The address of a full-word containing the token address. The token is returned by the INIT function and must be passed in all subsequent API requests.

**MF=(E,***address* **|** (*r1*-*r12*))
(Required if reentrant) The address of the parameter list area used by your program. This parameter list must be constructed in modifiable storage by calling the list form of the CEXPAPI macro using the MF=L parameter. See "List form syntax" on page 348 for details.

```
*---------------------------------------------------------------------*
    Start event collection from IMS Connect Extensions
*---------------------------------------------------------------------*

         CEXPAPI STARTMON,                                             +
               TOKEN=@TOKEN,                                           +
               BUFSIZE=BUFSIZE,                                        +
               LEVEL=LEVEL,                                            +
               THRESH=THRESH,                                          +
               ECBADDR=@BUFFECB,                                       +
               MF=(E,PARMLST)
         LTR   R15,R15                 Successful?
         BNZ   ...                     No, error

@TOKEN   DS    A                       Address of token
@BUFFECB DC    A(BUFFECB)              Address of buffer full ECB

BUFSIZE  DC    F'32768'                Buffer size
LEVEL    DC    F'4'                    Event collection level
THRESH   DC    F'80'                   Buffer full threshold
BUFFECB  DC    F'0'                    Buffer full ECB
```

# STOPMON

The STOPMON function stops collecting IMS Connect events by ending the STARTMON process. Subsequent READASYN requests are rejected by the server until another STARTMON is issued.

## Format

```
►►──label──CEXPAPI──STOPMON──,TOKEN=token──┬──────────────────────┬──────────────►◄
                                           └─,MF=(E─,parm_area─)─┘
```

### Parameters

**TOKEN=***address* **|** (*r2-r12*)
(Required) The address of a full-word containing the token address. The token is returned by the INIT function and must be passed in all subsequent API requests.

**MF=(E,***address* **|** (*r1-r12*))
(Required if reentrant) The address of the parameter list area used by your program. This parameter list must be constructed in modifiable storage by calling the list form of the CEXPAPI macro using the MF=L parameter. See "List form syntax" on page 348 for details.

```
*-------------------------------------------------------------------*
    Stop event collection from IMS Connect Extensions
*-------------------------------------------------------------------*

         CEXPAPI STOPMON,                                            +
               TOKEN=@TOKEN,                                         +
               MF=(E,PARMLST)
         LTR   R15,R15                 Successful?
         BNZ   ...                     No, error

@TOKEN   DS    A                       Address of token
```

## TERM

The TERM function terminates the CEXPAPI interface.

### Format

```
►►—label—CEXPAPI—TERM—,TOKEN=token——┬————————————————————————┬——————————————►◄
                                    └—,MF=(E—,parm_area—)—┘
```

### Parameters

**TOKEN=***address* **|** (*r2-r12*)
(Required) The address of a full-word containing the token address. The token is returned by the INIT function and must be passed in all subsequent API requests.

**MF=(E,***address* **|** (*r1-r12*))
(Required if reentrant) The address of the parameter list area used by your program. This parameter list must be constructed in modifiable storage by calling the list form of the CEXPAPI macro using the MF=L parameter. See "List form syntax" on page 348 for details.

```
*-------------------------------------------------------------------*
    Terminate CEXPAPI interface normally
*-------------------------------------------------------------------*

         CEXPAPI TERM,                                               +
               TOKEN=@TOKEN,                                         +
               MF=(E,PARMLST)
         LTR   R15,R15                 Successful?
         BNZ   ...                     No, error

@TOKEN   DS    A                       Address of token
```

# ECSA cleanup utility

The ECSA cleanup utility identifies storage and cross memory linkages used by the IMS Connect Extensions publisher API and publisher API clients and frees them.

Use the ECSA cleanup utility to:

- Free ECSA storage, if you permanently stop using the publisher API.
- Collect information for API problem determination.
- Recover from API errors that prevent the execution of either the API client or server.

**Attention:** Shut down IMS Connect Extensions and all publisher API clients before executing this utility.

## Sample JCL: CEXCESA

The member CEXCESA in the SCEXSAMP sample library contains an example job to list related control blocks in ECSA:

```
//CEXCESA  JOB (ACCOUNT),'NAME'
//STEP01   EXEC PGM=CEXAPIMN,PARM='CEXDFPUT',REGION=8M
//STEPLIB  DD DISP=SHR,DSN=funpre.SFUNLINK
//         DD DISP=SHR,DSN=cexpre.SCEXLINK
//MSGOUT   DD SYSOUT=*
//SYSIN    DD *
CONTROL MVSID=MVS1,CASE=LOWER
LIST JOBNAME=ALL
//*
```

*Figure 117. Publisher API cleanup utility*

## EXEC statement

The format of the EXEC statement is:

```
//stepname  EXEC PGM=CEXAPIMN,PARM='CEXDFPUT'
```

## SYSIN control cards

**CONTROL**
General control settings.

**LIST** List publisher API related system entries.

**VERIFY**
Check the current status of a system entry.

**DELETE**
Delete a system entry.

## Examples

Here are some example SYSIN control cards:

- List all system entries. Does not release locks:

  ```
  CONTROL MVSID=MVS1
  LIST JOBNAME=ALL
  ```

- List, verify and delete a job entry:

```
CONTROL MVSID=MVS1
VERIFY JOBNAME=USRADRV2
DELETE JOBNAME=USRADRV2
LIST JOBNAME=USRADRV2
```

- Delete system linkage header and name services (with FORCE option):

```
CONTROL MVSID=MVS1
VERIFY JOBNAME=TOKEN
DELETE JOBNAME=TOKEN,FORCE=YES
```

# CONTROL syntax

Here is the syntax for the CONTROL control card of the ECSA cleanup utility.

```
►►──CONTROL──MVSID=MVS_ID──┬─────────────────────┬──┬─────────────────────┬──────────►◄
                           │         ┌─LOWER─┐   │  │         ┌─KEEP──┐   │
                           └─,CASE=──┴─UPPER─┴───┘  └─,LOCK=──┴─CLEAR─┴───┘
```

## Parameters

**MVSID**
The value of the MVSID of the current system.

**CASE** Determines whether the utility output is printed in upper or lowercase.

**LOWER**
Lowercase

**UPPER**
Uppercase

**LOCK** Determines how the utility handles locks.

**KEEP** Locks are kept. The utility execution terminates when it finds a lock held by another task.

**CLEAR**
Locks are released unconditionally and the utility continues processing.

**Note:** Use this option only when IMS Connect Extensions and its clients are not active. This option is intended for use only when a lock is not released due to a program error, or when the IMS Connect Extensions task abnormally terminates due to a region cancel.

# LIST syntax

Here is the syntax for the LIST control card of the ECSA cleanup utility.

```
►►──LIST──JOBNAME=──┬─job_name─┬──────────────────────────────────────────────►◄
                    └─ALL──────┘
```

## Parameters

**JOBNAME**
The name of the job for which control block information is listed. Use `JOBNAME=ALL` to list control block information for all publisher API related jobs.

## VERIFY syntax

Here is the syntax for the VERIFY control card of the ECSA cleanup utility.

```
►►──VERIFY──JOBNAME=─┬─job_name─┬──────────────────────────────────────────►◄
                     └─TOKEN────┘
```

### Parameters

**JOBNAME**
The name of the job for which to verify active status. Use `JOBNAME=TOKEN` to verify the status of the system header and name services token.

## DELETE syntax

Here is the syntax for the DELETE control card of the ECSA cleanup utility.

```
►►──DELETE──JOBNAME=─┬─job_name─┬──FORCE=─┬─NO──┬──────────────────────────►◄
                     └─TOKEN────┘         └─YES─┘
```

### Parameters

**JOBNAME**
The name of the job whose ECSA storage area should be deleted. Use `JOBNAME=TOKEN` to delete the system header and name services token.

**FORCE**
Determines what action to take for active entries:

**NO** Delete an entry only when you have verified it and found that it was not active.

**YES** Delete the entry even if it is active.

**Note:** Use this option only when IMS Connect Extensions and its clients are not active. This option is intended for use only when a lock is not released due to a program error.

# Publisher API console

These topics how to use the publisher API console to manage connected publisher API clients.

The publisher API console allows you to:

- List summary information about connected publisher API clients.
- View detailed information about clients you select.
- Set the maximum number of clients that can connect.
- Stop and force-stop clients.

## Listing connected clients

The **PU** line command lists summary information about connected publisher API clients.

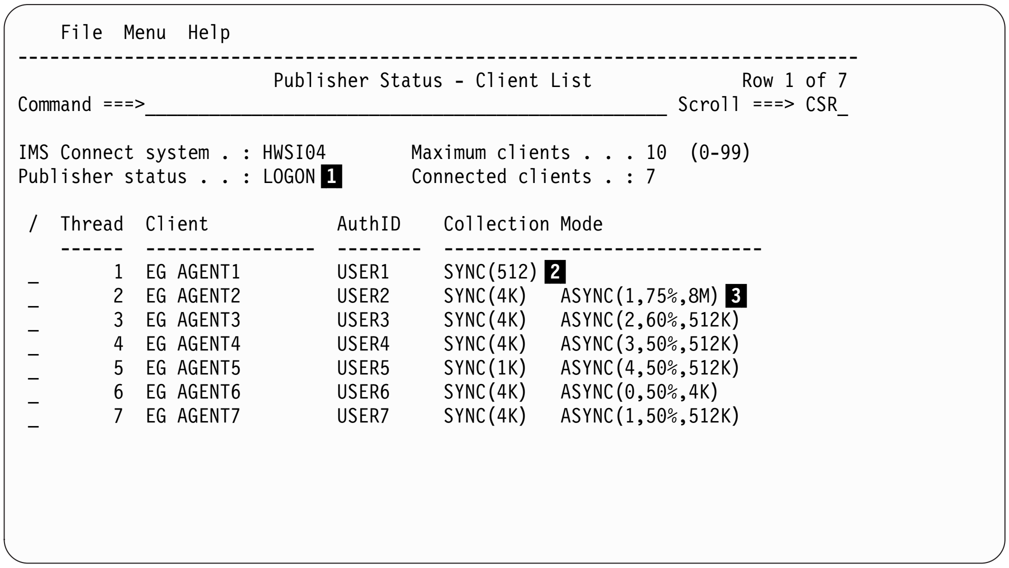
### Procedure

1. Select primary menu option 2 **Operations**.

2. Find the system for which you want to list clients, and then enter **PU** next to that system. The Publisher Status - Client List appears.

### Results

The following figure shows the Publisher Status - Client List with an example list of clients.



```
    File  Menu  Help
-------------------------------------------------------------------------------
                        Publisher Status - Client List              Row 1 of 7
Command ===>_________________________________________________ Scroll ===> CSR_

IMS Connect system . : HWSI04        Maximum clients . . . 10  (0-99)
Publisher status . . : LOGON 1       Connected clients . : 7

 /  Thread  Client           AuthID    Collection Mode
    ------  ---------------- --------  ------------------------------
 _       1  EG AGENT1        USER1     SYNC(512) 2
 _       2  EG AGENT2        USER2     SYNC(4K)   ASYNC(1,75%,8M) 3
 _       3  EG AGENT3        USER3     SYNC(4K)   ASYNC(2,60%,512K)
 _       4  EG AGENT4        USER4     SYNC(4K)   ASYNC(3,50%,512K)
 _       5  EG AGENT5        USER5     SYNC(1K)   ASYNC(4,50%,512K)
 _       6  EG AGENT6        USER6     SYNC(4K)   ASYNC(0,50%,4K)
 _       7  EG AGENT7        USER7     SYNC(4K)   ASYNC(1,50%,512K)
```

*Figure 118. Publisher Status - Client List*

**1** The status of the server-side of the Publisher API:

**LOGON**
Accepting client connections.

**NOLOGON**
Rejecting new client connections because **Maximum clients** is set to zero.

**SHUTDOWN**
IMS Connect shutting down.

**2** The data collection mode “SYNC(*nnnn*)” with the size of the synchronous buffer in kilobytes.

**3** The data collection mode “ASYNC(*c*,*xx*, *nnnn*)”. Where the following parameters from the STARTMON function appear:

***c*** Event collection level (LEVEL).

***xx*** Buffer-full threshold percentage (THRESH).

***nnnn*** Size of the STARTMON buffer (BUFSIZE).

## Viewing client details

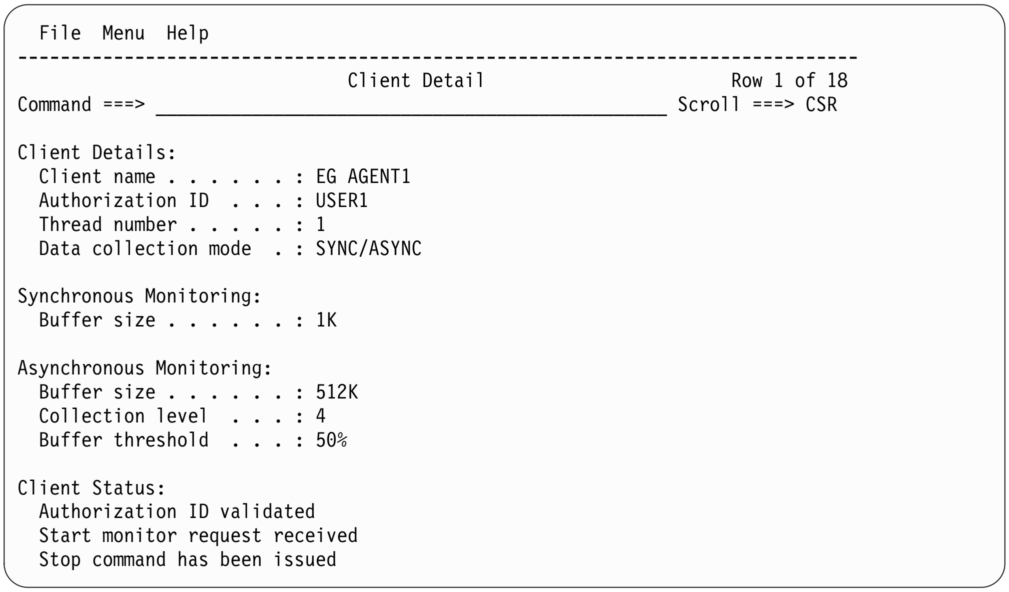
This topic explains how to view information about a connected publisher API client.

### Procedure

1. List connected clients. See “Listing connected clients” on page 361.
2. Enter an **S** next to one or more clients to select them.

3. Press Enter. The Client Detail panel appears. Figure 119 shows a typical display for this panel when a SYNC/ASYNC client is selected. For a description of each field on this panel, see the online help.

### Results



```
  File  Menu  Help
--------------------------------------------------------------------------------
                               Client Detail                    Row 1 of 18
Command ===> ________________________________________________ Scroll ===> CSR

Client Details:
  Client name . . . . . . : EG AGENT1
  Authorization ID  . . . : USER1
  Thread number . . . . . : 1
  Data collection mode  . : SYNC/ASYNC

Synchronous Monitoring:
  Buffer size . . . . . . : 1K

Asynchronous Monitoring:
  Buffer size . . . . . . : 512K
  Collection level  . . . : 4
  Buffer threshold  . . . : 50%

Client Status:
  Authorization ID validated
  Start monitor request received
  Stop command has been issued
```

*Figure 119. Client details panel*

As shown in the following table, most fields on the Client Detail panel map to publisher API functions.

*Table 9. Publisher API client details and equivalent API functions and parameters*

| Field | API function | Parameter |
|---|---|---|
| Client name | CONNECT | AGENT |
| Authorization ID | CONNECT | AUTHID |
| Thread number | Generated by IMS Connect Extensions | N/A |
| Data collection mode | SYNC/ASYNC when STARTMON established, otherwise SYNC | N/A |
| Buffer size (synchronous monitoring) | CONNECT | BUFSIZE |
| Buffer size (asynchronous monitoring) | STARTMON | BUFSIZE |
| Collection level | STARTMON | LEVEL |
| Buffer threshold | STARTMON | THRESH |
| Client status | Generated by IMS Connect Extensions | - |

## Setting the maximum number of clients

You select the maximum number of publisher API clients in the System Definition panel and the Publisher Status - Client List panel.

### About this task

- The **Maximum clients** option in the system definition sets the initial value for maximum clients. The maximum number of clients is set to this value every time you restart IMS Connect.
- The Publisher Status - Client List panel sets a value at runtime.

- If the maximum number of clients is reached, new connections are rejected.
- If you change the maximum number of clients to a value less than the number currently connected, new clients will not be able to connect, but existing clients will be unaffected until they disconnect.
- If you set the maximum clients to zero, the status of the publisher API changes to NOLOGON.

## Stopping a client

Stop a publisher API client when you want it to disconnect. Force-stop a client when a stop fails.

### About this task

- A stop tells the client to disconnect.
- A force-stop clears the client's entry from IMS Connect Extensions.

**Attention:** If you force-stop a client that has not disconnected that client may ABEND as cross memory resources were not properly released. For the client to be able to reconnect again, you may need to restart its region.

For details on how clients disconnect from IMS Connect Extensions, see "Termination procedure" on page 347.

**Note:** You can also stop and force-stop clients using the commands **P** and **F**. See the online help for details.

### Procedure

To stop a client:

1. List connected clients. See "Listing connected clients" on page 361.
2. Enter **P** next to one or more clients.
3. Press Enter. If the client is not stopped, use the force-stop command:
   a. Enter **F** next to a client you tried to stop.
   b. Press Enter.

# Part 7. Troubleshooting

These topics provide technical reference information to help you to troubleshoot and diagnose problems with IMS Connect Extensions.

**Topics:**

# Chapter 27. Troubleshooting connection failures

The information in this section can be used to help you diagnose and solve problems relating to connection failures between IMS Connect and IMS Connect Extensions components.

## Connection failure between a client and a console listener

If an attempt by a client to connect to the console listener on an IMS Connect system was unsuccessful, check for messages.

Symptoms of a connection failure depend on the client. They include:

- In the ISPF Operations dialog: the system appears with a status of INACT.
- In the Operations Console: the system appears with a red "box" status icon, indicating a failed connection attempt, or a red "exclamation point" status icon, indicating that the client did not receive a correctly formatted reply from a console listener. Error messages may also be displayed in the Console view.
- After running a batch command utility:
  - Message CEX5046E appears in the JES message log and in CEXPRINT, or:
  - Message HWSP1445E appears in the log of the IMS Connect task, indicating that an unknown exit has been specified with `MSGID=*CONSOL*`.

The most likely causes are:

- The specified system name is incorrect. Check the name used in the IMS Connect configuration member.
- The system is not active. Check that the system is active.
- The console port number specified in the system definition for the client does not match the port number that the console listener is using. Make sure you are using the port defined in IMS Connect Extensions, not one of the IMS Connect ports defined in the configuration member.

The following explanations may also apply if the error appears in IMS Connect Extensions Operations Console:

- The specified IMS Connect system is running a version of IMS Connect Extensions earlier than 2.2.0. This is indicated by a message "Invalid response received" in the Console view.
- The user ID or password specified in your connection details or in the details for this system is incorrect.
  - Select **Connection details** from the **View** menu in the Navigation view to check or reenter your IMS Connect Extensions user ID and password.
  - Right-click the system and select **Edit** to check or reenter the user ID and password in the system details.

  **Note:** Passwords must conform to the **Activate Mixed Case Passwords** setting in the system definition and the SECURITY PWCASE control option. If the connection fails because the user is not authenticated, check if you are using mixed-case passwords, and if so ensure the password case is correct.

After you have corrected the system or connection details, click **Refresh Connection Statuses** from the **View** menu in the Navigation view to attempt to reconnect using the new settings.

**Related reference**:

"System Definition" on page 190
Use the System Definition panel to define the options and features for an IMS Connect system. You must create one system definition in IMS Connect Extensions for every IMS Connect system that you wish to manage.

"Parameters" on page 443
The following keywords are supported in the CEXCTLIN DD statement.

## IMS Type-1 command failure due to TCP/IP connection error

The information in this section can be used to help you troubleshoot and diagnose problems when issuing IMS Type-1 commands from IMS Connect Extensions.

The following messages can be displayed when a user issues IMS Type-1 commands from IMS Connect Extensions:

```
CEX5050I CEX1 action : IMS Command Shell - failed  : TCP/IP CONNECT error, see log message CEX5401E
CEX5401E Unexpected TCPIP response. IP operation was BPX1CON, ERRNO was 1128, reason 76630291
CEX5415E Connection request failed, hostname P390, port 3707, errno 1128 connection refused
```

The command shell attempts to connect to an available IMS Connect message port to submit IMS Type-1 commands using message exit HWSJAVA0 to IMS. These messages indicate that the socket connection is being refused.

IMS Connect Extensions uses the local host name to derive a network address. It then issues a socket connection request to the first available message port at that address. This request has failed.

To diagnose the problem, perform the following checks.

- The TCP/IP *errno* denotes the reason for the failure.

  | errno | Description |
  |---|---|
  | **111** | Permission is denied. |
  | **1128** | Connection refused. The attempt to connect to a socket was refused because there was no process listening or because the queue of connection requests was full and the underlying protocol does not support retransmissions. |
  | **Other** | See "Return Codes (Errnos) Listed by Value" in *z/OS UNIX System Services Messages and Codes*. |

- Ensure that IMS Connect has an open message port defined in the configuration file. Check that the port is not protected. SSL ports are not supported.
- Check that the port is available on the host name address. Use the following TSO command to check the network address:

  ```
  PING hostname
  ```

  Use the following TSO command to ensure that IMS Connect is listening on the port at that address:

  ```
  NETSTAT ALLCONN (PORT nnnnn
  ```

- Ensure that any modifications to message exit HWSJAVA0 do not impact client message flows.

# Chapter 28. Messages and codes

Use the information in these messages to help you diagnose and solve IMS Connect Extensions problems.

## Return codes

The following return codes are set by IMS Connect Extensions:

**0**
Operation was successful.

**4**
Operation completed, but a warning (**W**) message was issued during processing.

**8**
Operation may be incomplete. A failure (**F**) message was issued and IMS Connect Extensions continues processing.

**12**
Operation may be incomplete. An error (**E**) condition occurred.

**16**
Operation is incomplete. A severe error (**S**) condition occurred.

## Return and reason codes supplied in the RSM

When IMS Connect Extensions rejects messages, it gives its return and reason code to the client in these RSM fields:

**RSM_RETCOD**
The return code is always 8, meaning the message was rejected.

**RSM_RSNCOD**
Possible reason codes are shown in the following table.

*Table 10. Reason codes generated by IMS Connect Extensions*

| OMUSR reason code | RSM (decimal value) | Description |
|---|---|---|
| SECFAIL | 40 | Security failure: access denied |
| DESTUNA | 72 | Transaction routing: destination unavailable |
| PACELMT | 150 | Transaction pacing: pacing threshold exceeded |
| N/A | 153 | Message exit not active |
| SECINOP | 154 | Security failure: user ID, password, or group information missing or invalid |
| N/A | 156 | Message exit deleted |

## Reason codes in internal error messages

Internal reason codes are used by IBM to debug internal errors. Only IBM can interpret these codes. Internal reason codes consist of 8 hexadecimal digits:

```
mmmmcccc
```

where:

**mmmm (the first four digits)**
Identifies the internal function or module that reported the error.

**cccc (the last four digits)**
Identifies the condition that triggered the internal error. Different modules may use the same four digits to indicate different error conditions.

## Message format

Messages begin with a unique identifier, followed by message text which may contain symbol information to identify the circumstance which caused the message:

```
CEXnnnnx Message text with symbol information
```

The message identifier has the format **CEXnnnnx** or **FUNnnnnx** where:

**CEX** Identifies the message as an IMS Connect Extensions **product** message.

**FUN** Identifies the message as an IMS Connect Extensions **functional support** message.

**nnnn** A unique four-digit message identification number.

**x** Indicates the severity of the message and the return code. The severity levels, from least to most severe, are:

**I** Information. No action required.

(Corresponds to a return code 0.)

**W** Warning. Possible problems occurred that you should evaluate.

(Corresponds to a return code 4.)

**F** Failure. Your action may be required as processing is incomplete.

(Corresponds to a return code 8.)

**E** Error. Your action required. Processing is incomplete.

(Corresponds to a return code 12.)

**S** Severe. Internal logic error requiring your attention.

(Corresponds to a return code 16.)

Each message also includes the following information:

**Explanation:**
The Explanation section explains what the message text means, why it occurred, and what its variables represent.

**System action:**
The System action section explains what the system will do in response to the event that triggered this message.

**User response:**
The User response section describes whether a response is necessary, what the appropriate response is, and how the response will affect the system or program.

## Messages written to SYSLOG

The following IMS Connect Extensions messages are written to SYSLOG:
CEX5006-5011
CEX5019
CEX5029-5030

CEX5034-5036
CEX5041-5042
CEX5048-5050
CEX5058, CEX5067, CEX5097
CEX5112-5113
CEX5131-5135
CEX5189-5190
CEX5192
CEX5194-5195
CEX5198-5199
CEX5400, CEX5404, CEX5407
CEX5587-5588
CEX5594

FUN1011-1012

FUN1015-1016

FUN2539

FUN2596

FUN2604

FUN2702

FUN2708-2711

### Controlling which messages are written to SYSLOG

You can use the MESSAGE control option to control which messages are written to SYSLOG. The ADD_WTO suboption writes additional messages to SYSLOG. The DROP_CEXPRINT suboption stops specified CEX and FUN messages being written to CEXPRINT.

### REXX messages written to SYSTPRT

To ensure that messages are written to SYSTPRT, include the `OPTION MSGLVL=VERBOSE` statement in the REXX exec after initializing the CEX host command environment.

## How to look up message explanations

You can use several methods to search for messages and codes.

### Searching an information center

In the search box that is located in the top left toolbar of any Eclipse help system, such as the IBM Information Management Software for z/OS Solutions Information Center, enter the number of the message that you want to locate. For example, you can enter DFS1065A in the search field.

Use the following tips to help you improve your message searches:

- You can search for information on codes by entering the code; for example, enter -327.
- Enter the complete or partial message number. You can use the asterisk wildcard character (*) to represent multiple characters, and you can use the question mark wildcard character (?) to represent a single character.

The information center contains the latest message information for all of the information management products that are included in the information center.

### Searching for messages on the Web

You can use any of the popular search engines that are available on the Web to search for message explanations. When you type the specific message number or code into the search engine, you will be presented with links to the message information in IBM information centers.

## CEX-prefixed messages

This topic describes messages with the CEX*nnnnx* message identifier format.

**CEX5004I IMS Connect Extensions status change detected :** *message*

**Explanation:** Indicates that a change in the status of the IMS Connect Extensions system has been detected. *message* identifies what has changed:

- IMS Connect Extensions now Active
- IMS Connect Extensions now Inactive
- Event Collection now Active
- Event Collection now Inactive
- Advanced Features now Active
- Advanced Features now Inactive
- Severe Error has occurred
- Event Collection level changed to *new* from *old*
- Trace recording level changed to *new* from *old*

**System action:** IMS Connect Extensions continues.

**User response:** None. Informational message only.

**CEX5005I Refresh affinity list=***afl_name* **a datastore entry referencing the affinity list for Supermember routing was not found**

**Explanation:** A datastore entry could not be found that references the affinity list member as a Supermember routing list.

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5006E ERROR DETECTED, RC=** *rc***, RSN=** *rsn*

**Explanation:** IMS Connect Extensions detected an internal error that prevents services and functions from continuing. The message comes in two forms: ERROR DETECTED, or CEX FAILED TO INITIALIZE. IMS Connect Extensions services might not have been available when the failure was detected. The CEXPRINT data set might contain more information about the failure.

**System action:** IMS Connect Extensions quiesces.

**User response:** Gather the following documentary evidence at the time the problem occurs: copy of the IMS Connect job output, the CEXPRINT data set and a portion of the JES syslog. Contact IBM Software Support.

**CEX5007E PROCESSING ERROR. SEE CEXPRINT DATA SET**

**Explanation:** IMS Connect Extensions detected an internal error that prevents services and functions from continuing. The CEXPRINT data set might contain more information about the failure. IMS Connect Extensions services might not have been available when the failure was detected.

**System action:** IMS Connect Extensions quiesces.

**User response:** Gather the following documentary evidence at the time the problem occurs: copy of the IMS Connect job output, the CEXPRINT data set and a portion of the JES syslog. Contact IBM Software Support.

**CEX5008E Event Collection Init failed, RSN=***rsn*

**Explanation:** IMS Connect Extensions detected an internal error that prevents services and functions from continuing. The CEXPRINT data set might contain more information about the failure.

**System action:** IMS Connect Extensions quiesces.

**User response:** If the reason code *rsn* has a value of 20, increase the region size and restart IMS Connect. Otherwise gather the following documentary evidence at the time the problem occurs: copy of the IMS Connect job output, the CEXPRINT data set and a portion of the JES syslog. Contact IBM Software Support.

**CEX5009E Can not process collection request, RSN=***rsn*

**Explanation:** IMS Connect Extensions detected an internal error that prevents services and functions from continuing. The CEXPRINT data set might contain more information about the failure.

**System action:** IMS Connect Extensions quiesces.

**User response:** If the reason code *rsn* has a value of 32, increase the region size and restart IMS Connect. Otherwise gather the following documentary evidence at the time the problem occurs: copy of the IMS Connect job output, the CEXPRINT data set and a portion of the JES syslog. Contact IBM Software Support.

**CEX5010I Pacing *type* threshold has been exceeded for object *object*.**

**Explanation:** Indicates that the number of incoming messages has exceeded the threshold value defined for an object.

*type* identifies the type of threshold:

- Warning
- Reject

*object* identifies the name of the object. The object is one of the following:

- HWS - the IMS Connect system
- DSG - Datastore Group
- DS - datastore

**System action:** IMS Connect Extensions continues.

**User response:** None. Informational message only. You may want to adjust the pacing thresholds for the object definitions.

**CEX5011I Pacing *type* threshold for object *object* is no longer being exceeded.**

**Explanation:** Indicates that the number of incoming messages is no longer exceeding the threshold value defined for an object.

*type* identifies the type of threshold:

- Warning
- Reject

*object* identifies the name of the object. The object is one of the following:

- HWS - the IMS Connect system
- DSG - Datastore Group
- DS - datastore

**System action:** IMS Connect Extensions continues.

**User response:** None. Informational message only.

**CEX5012I End of file on Recorder Trace input data set**

**Explanation:** The Recorder Trace conversion utility has reached the end of the recorder trace data set used as input.

**System action:** The Recorder Trace conversion utility stops processing.

**User response:** None. Informational message only.

**CEX5013E Input record is not a recorder trace ICON record**

**Explanation:** The Recorder Trace conversion utility has encountered a record in the input data set that is not a valid ITOC record.

**System action:** The Recorder Trace conversion utility stops processing.

**User response:** Look in the MSGOUT data set containing a hex dump of the record in error. Verify that the input data set is a valid recorder trace data set and try again.

**CEX5014E Input ICON record could not be converted**

**Explanation:** The Recorder Trace conversion utility has encountered an ITOC record but could not format the record correctly.

**System action:** The Recorder Trace conversion utility stops processing.

**User response:** Look in the MSGOUT data set containing a hex dump of the record in error. Verify that the input data set is a valid recorder trace data set and try again.

**CEX5015E Input ICON record header is not a type RC, SN, ER, or TO**

**Explanation:** The Recorder Trace conversion utility has encountered an ITOC record that does not contain a valid type identifier.

**System action:** The Recorder Trace conversion utility stops processing.

**User response:** Look in the MSGOUT data set containing a hex dump of the record in error. Verify that the input data set is a valid recorder trace data set and try again.

**CEX5016E Recorder trace ended early due to invalid record in input data set**

**Explanation:** The Recorder Trace conversion utility has encountered an invalid record in the input data set and is ending execution before reaching end of file on the input data set.

**System action:** The Recorder Trace conversion utility stops processing.

**User response:** Look in the MSGOUT data set containing a hex dump of the record in error. Verify that the input data set is a valid recorder trace data set and try again.

**CEX5017E Data set for DDname=*ddn* failed to open**

**Explanation:** This message indicates that the data set associated with the DD statement failed to open. *ddn* identifies the ddname.

**System action:** The utility stops processing.

**User response:** Verify that the listed DD statement is present and points to a valid data set.

**CEX5018E Data set for DDname=*ddn* failed to close**

**Explanation:** The data set associated with the named DD statement: *ddn*, failed to close.

**System action:** IMS Connect Extensions continues.

**User response:** Verify that the listed DD statement is present and points to a valid data set name.

**CEX5019I** *storage pool condition*

**Explanation:** This is an informational message indicating that one of the storage pools used for IMS Connect Extensions processing encountered the condition shown in the message.

**System action:** Active session information and details from event collection may be lost. Some IMS Connect Extensions functions may not operate. Another CEX5019I message will be issued to inform you when the problem has been resolved.

**User response:** If you receive reason code **00000215** "The pool attempted to expand and failed", increase the region size. Otherwise, contact IBM Software Support.

**CEX5020I IMS Connect Extensions initialized and *status*, console port is *port***

**Explanation:** Indicates that IMS Connect Extensions has initialized and shows its status, either Active or Inactive.

**System action:** IMS Connect Extensions continues.

**User response:** If IMS Connect Extensions is Inactive, refer to other messages and restart IMS Connect. Otherwise this is an information message only.

**CEX5021I Advanced features *status*, transaction routing *status*, workload balancing *status*, statistics collection *status*, event collection *status* with collection level *level*, publisher API *status*, OTMA routing rules *status*, ODBM routing rules *status*, session message limit *status*, session message limit threshold *threshold***

**Explanation:** This message complements CEX5020I. It details the Advanced feature fields and system options that were in effect at startup, including their status (Active or Inactive).

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5022I Pacing ACTIVE|INACTIVE, with interval count *icnt*, warning threshold *wtv*, reject threshold *rtv***

**Explanation:** Indicates that IMS Connect Extensions has initialized and Pacing is active at the IMS Connect system level. *icnt* identifies the pacing interval count, *wtv* the warning threshold and *rtv* the reject threshold.

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5023I Security *status*, ACEE caching *status*, security validation *status*, validation type is *type*, validation class is *class*, cache ageing interval is *nn* min.**

**Explanation:** Indicates that IMS Connect Extensions has initialized and Security is active.

- *status* identifies either Active or Inactive.
- *type* identifies the validation type, either "IMS Connect" or "IMS Connect + IP Address + Port".
- *class* identifies the RACF security class used to validate access to IMS Connect.
- *nn* identifies the time interval for clearing aged user ID information from the ACEE cache.

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5024I Publisher API Client *client*, *state***

**Explanation:** Information indicating that a client has connected to the publisher API.

*client* The client name or thread number or both.

*state* The state of the connection. For example:

**has connected**
Connection established.

**has started**
Connection established and transfer started.

**disconnected normally**
Connection ended normally

**has failed**
The connection failed.

**System action:** Processing continues.

**User response:** In most cases the message is informational only. Some states might warrant some more investigation or action. For example:

**could not be found**
Check if the client recently disconnected or if its thread number was reused.

**was force-stopped by the API console**
The client was requested to terminate by a user request issued from the publisher API console. The actual client disconnect from a previous API console request is still outstanding. This request was ignored. If this situation persists for more than a few minutes then the client may no longer be listening for modification requests. In this case, consider using the FORCE option.

**CEX5025I** **ECSA Cleanup utility execution error, the utility is** *state*

**Explanation:** This message is issued when the IMS Connect Extensions CSA cleanup utility encounters an error during execution. *state* can be either of the following:

- not executing in an Authorized state
- not executing in protect key 7

**System action:** Utility execution ends.

**User response:** The utility must execute in an authorized state and in key 7. Correct the problem causing the utility to execute in a non-authorized state and rerun the utility. If the utility is not executing in storage key 7 make sure that a Program Properties table entry exists for module CEXAPIMN.

**CEX5026I** *control statements from SYSIN DD*

**Explanation:** Lists the control statements in the SYSIN DD.

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5027I** **CEX Journal Print Utility has reached the stop after point STOAFT=***position*

**Explanation:** CEX Journal Print Utility has completed processing to the position defined by the STOAFT parameter and this position is before the end-of-file.

**System action:** Processing continues.

**User response:** None. Informational message only.

| **CEX5028E** **Error code** *nn* **-** *API_error*

**Explanation:** An internal error occurred during initialization or termination processing, causing the publisher API to terminate.

**System action:** The publisher API terminates or does not start.

**User response:** Contact IBM Software Support.

**CEX5029I** *API_status*

**Explanation:** Indicates the status of close processing for the publisher API with one of the following messages:

- CEX publisher API closing with *number* clients still connected
- | CEX publisher API closing with *number* clients to disconnect from the API
- CEX publisher API waiting for *number* clients to disconnect from the API
- CEX publisher API waiting for *number* PC routines to end before closing

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5030I** **All CEX Publisher API clients have disconnected**

**Explanation:** All API clients have disconnected from the publisher API.

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5031I** **ARR Recovery successful for client=***client***, PSW=***program_status_word*

**Explanation:** The publisher API recovered from an error but the client request did not complete. The API provides the client with return and reason codes.

**System action:** IMS Connect Extensions continues.

**User response:** Contact IBM Software Support. An entry was recorded in SYS1.LOGREC.

**CEX5032I** **CEX Publisher API has initialized; Maximum clients** *number_of_clients*

**Explanation:** The publisher API has initialized the maximum number of clients that can connect is listed.

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5033F** **Print Utility control card error;** *control card*

**Explanation:** Control card error detected. The *control card* variable provides additional information.

**System action:** The Journal Print Utility terminates processing.

**User response:** Correct the control card causing the error and re-issue the job.

**CEX5034I Second copy of HWSTECL0 in STEPLIB**

**Explanation:** A second copy of HWSTECL0 was found in the STEPLIB concatenation. IMS Connect Extensions will initialize the second copy and allow it to process all IMS Connect instrumentation events.

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5035I Control passed to the second copy of HWSTECL0**

**Explanation:** A second copy of HWSTECL0 was found in the STEPLIB concatenation. IMS Connect Extensions will initialize the second copy and allow it to process all IMS Connect instrumentation events.

**System action:** IMS Connect Extensions continues.

**User response:** None. Informational message only.

**CEX5036I Second copy of HWSTECL0** *has/has not initialized*; **RC=***rc* **RSN=***rsn*

**Explanation:** A second copy of HWSTECL0 was found in the STEPLIB concatenation. IMS Connect Extensions either failed or succeeded in initializing it.

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5037F Command parsing error** *rsn*. **INFO=***information*

**Explanation:** Journal Print Utility parser error. Additional information in this message helps diagnose location of the error.

**System action:** Journal Print Utility processing terminates.

**User response:** Correct the control statement.

**CEX5038F Print utility execution ended due to error in control card**

**Explanation:** Journal Print Utility control card error was found. This message may be preceded by one or more error messages explaining the errors in the control statements.

**System action:** Journal Print Utility processing terminates.

**User response:** Correct the control statement.

**CEX5039W Print Utility did not find any event records that qualified for formatting**

**Explanation:** The Journal Print Utility did not produce an output, because it did not find any records. It may be that the data set did not contain any records or that no records matching the selection criteria were found.

**System action:** None

**User response:** None. Warning only..

**CEX5040I Exit** *exit_name* **loaded at** *entry_point*, **length** *module_length*

**Explanation:** A user message exit has been located.

*exit_name*
The name of the user message exit which has been located.

*entry_point*
The entry point address.

*module_length*
The length of the module.

**System action:** None

**User response:** None. Informational message only.

**CEX5041E Load for MODULE=***module_name* **failed, second HWSTECL processing bypassed**

**Explanation:** The module, *module_name*, failed to load. Client HWSTECL0 will not be initialized. The *module_name* is one of the following:

**FUNXLOD2**
This is the extended loader. It is required to load HWSTECL0.

**HWSTECL0**
The client version of HWSTECL0.

**System action:** Processing continues. Client version of HWSTECL0 is not loaded.

**User response:** Contact IBM Software Support.

**CEX5042E FUNXLOD2 failed to initialize, second HWSTECL0 processing bypassed**

**Explanation:** The extended loader failed to initialize.

**System action:** Processing continues. Client version of HWSTECL0 is not loaded.

**User response:** Contact IBM Software Support.

**CEX5043I** *batch command control statements*

**Explanation:** Lists the batch command utility control statements in the SYSIN data set.

**System action:** Processing continues

**User response:** None. Informational message only.

**CEX5044E** *parsing error information*

**Explanation:** The CEX host command environment and CEXDFCMD utility both enable you to submit IMS Connect Extensions commands in batch. While parsing a statement one of the following syntax errors was detected:

- **The SYSIN data set does not contain any valid control cards**

  Provide control cards to the CEXDFCMD utility.
- **CONNECT command error;**
  - **At least one of the following keywords must be specified; HOST, DNSNAME, IPV4ADR or IPV6ADR**

    Specify the IMS Connect system using one of these keywords: DNSNAME (DNS name), IPV4ADR (IPv4 formatted address), IPV6ADR (IPv6 formatted address), or (CEX host command environment only) HOST (host name).
  - **Only one of the following keywords can be specified; HOST, DNSNAME, IPV4ADR or IPV6ADR**

    Use only one keyword to specify the IMS Connect system.
  - **The PORT keyword must be specified**

    Include the console port (PORT) of the IMS Connect system in the CONNECT statement.
  - **The HWSID keyword must be specified**

    Include the system ID (HWSID) of the IMS Connect system in the CONNECT statement.
  - **A password can not be specified without a USERID unless a value of "PASSTIK" is used**

    Either provide a USERID on the CONNECT statement or specify PASSWD=PASSTIK.
  - **CONNECT command required before commands that communicate with IMS Connect**

    You must connect to the target IMS Connect system before issuing any command that requires a connection, such as SWITCH or TRACE.
- **TRACE command error;**
  - **Invalid conditional trace IP_ADDRESS name specified**

    The IP_ADDRESS conditional resource type name must be a valid IPv4 address, IPv6 address, or host name.
  - **Invalid conditional trace resource name specified**

    The value specified in the condition was too long or contained invalid characters. For example, CLIENT_NAME, TRANSACTION, MESSAGE_EXIT, and LTERM must not start with a number and must not be longer than 8 characters.
  - **Invalid port number specified**

    The PORT keyword must specify a decimal value in the range 1 - 65535.
- **CLEAR command error;**
  - **Invalid user ID or user ID filter**

    The USERID keyword must specify a string that represents a valid user ID, an asterisk, or a partial user ID ending in an asterisk (*). No other wildcards are permitted.

**System action:** Command not submitted.

**User response:** Correct the REXX host command or batch utility control card and resubmit the job.

**CEX5045W** *command utility warnings*

**Explanation:** One of the following:

- *No valid commands were found to process*

  A connection was made but no commands were specified. For example, specify SWITCH to switch the active journal.
- *Read socket request detected exceptional condition*

  A TCP/IP exception condition was detected. Resubmit the job.
- *HWSID name specified did not match the HWSID name of the target system*

  The IMS Connect system listening on the port and address you specified has a different system identifier than the one you specified. Use the correct HWSID for the target system.
- *The target IMS Connect system may not be active*

  The target IMS connect system is not responding either because it is not active or because it is not accepting connections from the utility.

**System action:** Commands may not have been processed.

**User response:** If required, correct the control card and resubmit the job.

**CEX5046E** **TCPIP request ended with error, request:***command***, Return code:***RC* **Error number:***ERRNO*

**Explanation:** Utility detected TCPIP error.

*command*
: The type of TCPIP request.

*RC*
: The return code from TCPIP (generally FFFFFFFF).

*ERRNO*
: The socket error number.

**System action:** Commands may not have been processed.

**User response:**

1. Check that you specified the port number of the IMS Connect Extensions console port, and not an

IMS Connect message port. See “Connection failure between a client and a console listener” on page 367.

2. Using the error number (ERRNO), refer to *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference* to diagnose the error.

**CEX5047I CEX Publisher API clients are now being forcibly terminated in order for shutdown to proceed**

**Explanation:** IMS Connect Extensions is forcibly terminating publisher API clients that have not responded to the shutdown request. This is to allow IMS Connect to shutdown.

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5048W IMS Connect shutdown is now waiting indefinitely for all Publisher API clients to disconnect.**

**Explanation:** This message appears if you reply NO to message CEX5049W. The message means that IMS Connect will not shutdown until all Publisher API clients disconnect.

**System action:** System prepares to shutdown.

**User response:** No action required.

**CEX5049W Publisher API clients still connected. Reply "YES" to force disconnection or "NO" to wait for client disconnect.**

**Explanation:** IMS Connect is trying to shut down but Publisher API clients are still connected. These clients have not responded to the shutdown request.

**System action:** System prepares to shutdown.

**User response:** Reply with either "YES" or "NO":

**YES** Publisher clients will be forced to disconnect. Data they are accumulating may be lost and their memory region may become corrupt.

**NO** IMS Connect will wait indefinitely (issuing message CEX5048W).

**CEX5050I** *user type message*

**Explanation:** An IMS Connect Extensions command message. *user* identifies the User ID who issued the command. *type* identifies whether the command request is about to be issued (type Request) or has just been processed (type Action).

*message* contains the text of the message. This represents the command that was issued. Additionally, if the type of request is Action, the result of the command request will indicate whether the command was “successful”, “warning”, or “failed”. “Warning” and “failed” commands will contain an explanation.

Unsuccessful command requests contain a reason code text to aid problem determination.

**System action:** Processing continues.

**User response:** For warning or failed messages check the reason message for an explanation and if required verify the command and try again.

**CEX5051I IMS Connect XML adapter has been detected**

**Explanation:** IMS Connect Extensions has detected that IMS Connect message exits are using a parameter macro with support for the XML adapter.

**System action:** Processing continues.

**User response:** No action required.

**CEX5052W Exit** ***xname*** **not defined to IMS Connect Extensions, advanced features disabled for this message exit**

**Explanation:** The IMS Connect message exit *xname* is not defined in IMS Connect Extensions.

**System action:** Processing continues. Advanced features in IMS Connect Extensions will not be available for messages processed by the exit.

| **User response:** If you need to use advanced functions
| in IMS Connect Extensions that depend on this exit,
| load it using the **LOAD** primary command in the User
| Exits definition dialog (option 1.2).

**CEX5053I** *command*

**Explanation:** Echoes the command received by the IMS Connect Extensions command shell.

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5054I** *command*

**Explanation:** Echoes the modify command received by the IMS Connect Extensions command shell.

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5055I** *hwsid* **Command complete**

**Explanation:** Confirms command completion for the IMS Connect system identified by *hwsid*.

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5056I** *response*

**Explanation:** Contains the response from IMS Connect to a command issued from IMS Connect Extensions.

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5057F** **Reload of user exit** *exit_name* **failed during INIT call. Return code:** *rc* **Reason code:** *rsn*

**Explanation:** User exit reload processing was in progress and the INIT call to the new version of the exit ended with a non-zero return code. *exitname* identifies the user exit, *rc* and *rsn* the return and reason codes.

**System action:** The RELOAD command is terminated. The old exit is not replaced by the new version. Processing continues.

**User response:** Investigate why the new user exit program issued the non-zero return code. Correct the problem and then retry the reload command.

**CEX5058E** **CEX Command processing activation failed**

**Explanation:** IMS Connect Extensions could not activate command processing. Message CEX5006E provides additional information to help you resolve this issue.

**System action:** Processing continues.

**User response:** Restart IMS Connect Extensions. If the issue recurs, contact IBM Software Support, including the contents of CEX5006E.

**CEX5059I** **Command processing** *status information*

**Explanation:** Echoes the status of commands and PassTicket generation.

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5060I** **Datastore name** *dsn***,***datastore attributes*

**Explanation:** Echoes the definition for a datastore. This message does not show whether the datastore is active in IMS Connect.

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5061I** *datastore group attributes*

**Explanation:** Echoes the definition for a datastore group and lists the datastores that are part of that group.

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5062I** **Application=***application_name attributes*

**Explanation:** Echoes the definition for an application.

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5063I** **Transaction=***txn attributes*

**Explanation:** Echoes the definition for a transaction code.

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5064I** *affinity list members*

**Explanation:** Echoes the datastores in an affinity list.

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5065I** **Affinity list=***aflname* **is a SuperMember routing list for datastore=***dst*

**Explanation:** Indicates that the affinity list is being used to determine candidates for the Resume TPIPE messages routed to *dst*.

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5066W** *hwsid reason*

**Explanation:** Indicates command processing has failed for the named IMS Connect system (*hwsid*) and shows the IMS Connect reason code or IMS Connect Extensions internal reason code associated with the failure.

**System action:** Processing continues.

**User response:** Correct the problem and reissue the command.

**CEX5067E** **Insufficient storage available**

**Explanation:** IMS Connect Extensions was unable to acquire sufficient storage.

**System action:** IMS Connect Extensions stops processing. No further event collection or advanced functionality is possible.

**User response:** Increase the region size parameter for the IMS Connect job and restart IMS Connect.

**CEX5068I** **Mixed case passwords ACTIVE | INACTIVE**

**Explanation:** IMS Connect Extensions has been initialized. The status of the mixed-case passwords option is shown in this message.

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5069E** **Insufficient storage. Statistics collection unavailable**

**Explanation:** IMS Connect Extensions was unable to acquire sufficient storage to collect statistics.

**System action:** IMS Connect Extensions continues without statistics collection. The Status Monitor is unavailable.

**User response:** Increase the region size parameter for the IMS Connect job and restart IMS Connect.

**CEX5070E** **Refresh SuperMember affinity list=***aflname condition*

**Explanation:** IMS Connect Extensions detected an error when trying to refresh an affinity list that is being used in SuperMember routing.

**System action:** Processing continues.

**User response:** Correct the error.

**CEX5071E** *condition dstn* **affinity list** *aflname*

**Explanation:** IMS Connect Extensions detected an error trying to refresh an affinity list that is being used in SuperMember routing.

**System action:** Processing continues. Resume TPIPE messages referring to datastores in this affinity list may not be routed.

**User response:** There may be other messages in CEXPRINT that provide more information on this error. If these messages indicate problems with existing resource definitions, correct the definitions and reissue the refresh request. If the error condition indicates that a needed resource definition could not be found, report the problem to IBM.

**CEX5072I** *information*

**Explanation:** Provides information about one of the following dynamic changes to definitions:

- SuperMember affinity list=*aflname* was added as the SuperMember routing list for datastore=*dstname*
- SuperMember affinity list=*aflname* was removed as a SuperMember routing list and is no longer referenced by a datastore entry.
- Affinity list member=*aflname* was added during refresh processing.
- Supermember routing disabled for Datastore=*dstname* correct and refresh the affinity list then refresh the datastore.

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5073E** *error information*

**Explanation:** The IVP detected an error. The error can either be due to a control card error or an execution error. The message text identifies the precise problem.

**System action:** The IVP ends.

**User response:** Correct the error and rerun the IVP.

**CEX5074I** **Message translation activation commencing for exit** *xname* **using client CCSID** *clid* **and server CCSID** *svid*

**Explanation:** IMS Connect Extensions is attempting to initialize message translation.

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5075I** **CEX trace is active | inactive, level=***level***, tracing by** *tracing_options*

**Explanation:** This message displays the current status of the IMS Connect Extensions OTMA trace. If tracing is active the message includes the tracing level (1 or 2) and the status of any conditional trace resources.

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5076I** **Message translation activation successful for exit** *xname*

**Explanation:** Message translation activation successful.

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5077E** **Message translation activation failed for exit** *xname***, unable to detect a supported translation technique in the exit RSN=***rsn*

**Explanation:** IMS Connect Extensions was unable to activate message translation for the exit.

**Note:** Internal reason codes are not documented here as they are used by IBM to debug internal errors. Only IBM can interpret these codes.

**System action:** Depending on the error option in the exit definition, IMS Connect Extensions will either disable the exit or allow messages to use the original translation routine in the exit.

**User response:** Verify that the exit contains a translation routine that is similar to the one used in HWSSMPL1. If it does, contact IBM Software Support.

**CEX5078E Message translation activation failed for exit *xname*, client-server CCSID pairing not defined to z/OS Unicode Services**

**Explanation:** IMS Connect Extensions could not find conversion in z/OS Unicode Services from the client CCSID to the server CCSID, or from the server CCSID to the client CCSID.

**System action:** Depending on the error option in the exit definition, IMS Connect Extensions will either disable the exit or allow messages to use the original translation routine in the exit.

**User response:** Contact your system administrator and request that the conversion image be updated to include the required conversions.

**CEX5079E Message translation activation failed for exit *xname*, client-server CCSID pairing not an ASCII-to-EBCDIC conversion**

**Explanation:** Translation activation failed because the CCSIDs create character code sets that do not share alphanumeric characters with ASCII and EBCDIC.

**System action:** Depending on the error option in the exit definition, IMS Connect Extensions will either disable the exit or allow messages to use the original translation routine in the exit.

**User response:** You cannot use custom message translation with a client CCSID that does not have alphanumeric characters in common with ASCII or a server CCSID that does not have alphanumeric characters in common with an EBCDIC character code set. Select a compatible CCSID for the client, server, or both.

**CEX5080E Message translation activation failed for exit *xname*, client-server CCSID pairing not a single byte character set (SBCS) conversion**

**Explanation:** IMS Connect Extensions detected that one or more CCSIDs specify a multibyte character code set. IMS Connect Extensions does not support translation for multibyte character sets (MBCS). You can only use message translation with single-byte character sets (SBCS). All Unicode encoding schemes (UTF 8, 16 , 32) are examples of a multibyte character sets, with UTF 8 and 16 having variable byte lengths.

**System action:** Depending on the error option in the exit definition, IMS Connect Extensions will either disable the exit or allow messages to use the original translation routine in the exit.

**User response:** Select only single-byte code pages.

**CEX5081E Message translation activation failed for exit *xname*, z/OS Unicode Services error RC=*rc*, RSN=*rsn***

**Explanation:** z/OS Unicode Services returned an unexpected return and reason code.

**System action:** Depending on the error option in the exit definition, IMS Connect Extensions will either disable the exit or allow messages to use the original translation routine in the exit.

**User response:** To interpret the return and reason code, refer to *z/OS Support for Unicode: Unicode Services*.

**CEX5082E Message translation activation failed for exit *xname*, z/OS Unicode Services interface error RC=*rc*, RSN=*rsn***

**Explanation:** A failure occurred in the IMS Connect Extensions interface for z/OS Unicode Services.

**Note:** Internal return and reason codes are not documented here as they are used by IBM to debug internal errors. Only IBM can interpret these codes.

**System action:** Processing continues.

**User response:** Refer to Chapter 29, "Gathering diagnostic information," on page 421, and then contact IBM.

**CEX5083E Message translation activation failed for exit *xname*, z/OS Unicode Services inactive, callable service routine not found**

**Explanation:** Message translation activation has failed for the user exit *xname*. IMS Connect Extensions could not find the z/OS Unicode Services callable service routine. If stub routine CUNLCNV is statically linked then this is probably due to a linkedit error in SMP/E during product installation. If the stub routine is dynamically located via the CVT then this is due to a z/OS environmental error.

**System action:** Processing continues. The status of the message exit depends upon the error options specified in the message exit definition.

**User response:** If stub routine CUNLCNV is statically linked then verify that the Unicode Services library ACUNMOD (SYS1.ACUNMOD) has been defined to SMP/E and that no errors have occurred during product installation. If the stub routine is dynamically located via the CVT then consult your system administrator to ensure that z/OS Unicode Services is properly installed and configured.

**CEX5084W Exit *xname* disabled due to errors during message translation activation**

**Explanation:** An error has occurred in setting up message translation and based on the error processing option for this exit definition, IMS Connect Extensions has disabled the exit.

**System action:** Processing continues.

**User response:** Previous error messages indicate the cause of the error.

**CEX5085W Exit *xname* continues processing without IMS Connect Extensions message translation**

**Explanation:** An error occurred in activating message translation and based on the error processing option for this exit definition, IMS Connect Extensions allows the original translation routine of the message to be used instead.

**System action:** Processing continues.

**User response:** Previous error messages indicate the cause of the error.

**CEX5086W Inconsistent specification of mixed case password usage, IMS Connect setting is ACTIVE|INACTIVE and IMS Connect Extensions setting is ACTIVE|INACTIVE**

**Explanation:** The mixed-case password settings for IMS Connect and IMS Connect Extensions are inconsistent.

**System action:** Processing continues.

**User response:** Review the mixed-case password settings in the IMS Connect Extensions system definition and in IMS Connect, and correct any inconsistency.

**CEX5087E Could not add|delete persistent trace record to|from the repository, RC=*rc*, RSN=*rsn*, *error_text***

**Explanation:** A problem was encountered when trying to add or delete the persistent trace member in the repository. The following cases might be reported:

- The record could not be deleted because the trace member is missing from the repository.
- The record could not be deleted or added for some other reason.

**System action:** Processing continues. Whether tracing will be active when IMS Connect next restarts cannot be determined.

**User response:** Contact IBM Software Support. Report the return code (*rc*) and reason code *rsn*.

**CEX5088E Could not add|delete persistent trace record to|from the repository, member is in use, RC=*rc*, RSN=*rsn*, trace may be|may not be active after restart**

**Explanation:** A problem was encountered when trying to add or delete the persistent trace member in the repository. The member that needs to be added or updated is in use by another user. Whether tracing will be active when IMS Connect next restarts cannot be determined.

**System action:** Processing continues. The IMS Connect Extensions trace is active. The trace might not automatically start when IMS Connect is restarted.

**User response:** Release the member and try the action again.

**CEX5089I The CEX trace has been restarted as requested**

**Explanation:** In a previous execution of IMS Connect Extensions tracing was active and the “Reactivate tracing after system restart” option was selected.

**System action:** Processing continues. The IMS Connect Extensions trace is active.

**User response:** None. Informational message only.

**CEX5090W The Routing Rule for Destid=*dest*, Transaction qualifier=*quallist*, message type=*message_type*, from member=*member1* was superseded by member=*member2***

**Explanation:** When building the routing descriptor block for the named DestID and optionally a qualifier, the block was updated by more than one member of the same scope. The order in which one rule supersedes another is determined by the sort order of the repository member names for the two rules.

**System action:** Processing continues.

**User response:** Ensure that the update by multiple members was intentional.

**CEX5091I** *OTMA routing rule descriptions*

**Explanation:** This message is used to list the OTMA routing rules that are in effect following initial startup or refresh of a routing descriptor. The entries can include a master rule and qualifying rules for a DestID. When a qualifying rule is loaded for which there is no corresponding master rule in the repository, an implied master rule is generated internally at run time.

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5092E** *reason*

**Explanation:** This message is used to show errors that were detected while building the OTMA routing descriptor blocks. Possible errors are:

- A required group member=*member*, is missing
- A required OTMA routing list member=*member*, is missing

**System action:** Processing continues.

**User response:** Correct the reason that the group member appears to be missing.

**CEX5093I** *OTMA routing rule descriptions*

**Explanation:** This message is used to list the target list names that are being used by OTMA routing. Use this to verify that target lists and target references are correctly set up for OTMA routing.

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5095W** **The Routing Rule for Alias=***alias***, PSBNAME=***psbname***, request type=***request_type***, was also updated by member=***member*

**Explanation:** When building the ODBM routing descriptor block for the named alias and PSB name, the block was updated by more than one member of the same scope (that is, Group or System).

**System action:** Processing continues. The ODBM routing descriptor block is built.

**User response:** This condition is permissible, though unusual. Check the ODBM routing rules descriptions in CEXPRINT to ensure that the update by multiple members was intentional.

**CEX5096F** **IP Address keyword error:** *reason*

**Explanation:** The print utility READ62 command found an error in the format of a specified IP address. The text of this message explains the error. IPV4 addresses must be specified in dotted decimal format. IPV6 addresses must be specified in colon hex format.

**System action:** Print utility processing terminates.

**User response:** Correct the IPV4ADR or IPV6ADR keyword in the READ62 control card and rerun the job.

**CEX5097I** **IMS Connect Extensions Advanced Functions unavailable due to previous error**

**Explanation:** IMS Connect Extensions advanced functions have been disabled due to a previous error.

**System action:** No IMS Connect Extensions advanced features will run.

**User response:** Previous error messages explain the reason for the error and help diagnosis.

**CEX5098E** **Internal error** *number* **INFO=***info* **INFO2=***info2*

**Explanation:** IMS Connect Extensions has detected an internal processing error.

**Note:** Internal reason codes are not documented here as they are used by IBM to debug internal errors. Only IBM can interpret these codes.

**System action:** IMS Connect Extensions functionality may be impacted.

**User response:** Review CEXPRINT for other messages. Gather the following documentary evidence at the time the problem occurs: copy of the IMS Connect job output, the CEXPRINT data set and a portion of the JES syslog. Contact IBM Software Support.

**CEX5099S** **Internal logic error** *number* **INFO=***info* **INFO2=***info2*

**Explanation:** IMS Connect Extensions has detected an internal logic error.

**Note:** Internal reason codes are not documented here as they are used by IBM to debug internal errors. Only IBM can interpret these codes.

**System action:** IMS Connect Extensions functionality may be impacted.

**User response:** Review CEXPRINT for other messages. Gather the following documentary evidence at the time the problem occurs: copy of the IMS Connect job output, the CEXPRINT data set and a portion of the JES syslog. Contact IBM Software Support.

**CEX5100F** **No System Definition exists for IMS Connect system** *hwsid*

**Explanation:** IMS Connect Extensions failed to find a System Definition for the IMS Connect system.

*hwsid* identifies the IMS Connect system name.

**System action:** IMS Connect Extensions quiesces. No further event collection or advanced functionality is possible. IMS Connect continues with no IMS Connect Extensions functionality.

**User response:** Define a System Definition using the

ISPF interface and restart the IMS Connect system. The System Definition name **must** be the same as the HWS ID parameter in the IMS Connect configuration member.

**CEX5101F Datastore *ds* has not been defined**

**Explanation:** IMS Connect Extensions found the datastore *ds* had not been defined to IMS Connect Extensions in the Definitions data set.

**System action:** IMS Connect Extensions continues.

**User response:** If you want to use any advanced functions, use the ISPF interface to define the named datastore, then stop and restart the IMS Connect system.

**CEX5102F Datastore *ds* has an undefined Datastore Group *dsg***

**Explanation:** IMS Connect Extensions found the Datastore Group *dsg* had not been defined to IMS Connect Extensions in the Definitions data set.

**System action:** IMS Connect Extensions continues.

**User response:** Using the ISPF interface update the named datastore definition and with a valid Datastore Group. To implement the change either stop and restart the IMS Connect system or refresh the datastore using the REFRESH command.

**CEX5103F Transaction *txn* has an undefined affinity list *aflist***

**Explanation:** IMS Connect Extensions found the Affinity List *aflist* had not been defined to IMS Connect Extensions in the Definitions data set.

**System action:** IMS Connect Extensions continues.

**User response:** Using the ISPF interface define the Affinity List or update the transaction definition with an existing Affinity List. To implement the change either stop and restart the IMS Connect system or refresh the transaction or the Affinity List using the REFRESH command.

**CEX5104F Datastore *ds* has an undefined Affinity list *aflist***

**Explanation:** IMS Connect Extensions found the Affinity List *aflist* had not been defined to IMS Connect Extensions in the Definitions data set.

**System action:** IMS Connect Extensions continues.

**User response:** Using the ISPF interface define the Affinity List or update the datastore definition with an existing Affinity List. To implement the change either stop and restart the IMS Connect system or refresh the datastore using the REFRESH command.

**CEX5105F Application *appl* has not been defined**

**Explanation:** IMS Connect Extensions found the Application *appl* had not been defined to IMS Connect Extensions in the Definitions data set.

**System action:** IMS Connect Extensions continues.

**User response:** Using the ISPF interface define the Application or update the System Definition with an existing Application. To implement the change either stop and restart the IMS Connect system or refresh the Application or the System Definition using the REFRESH command.

**CEX5106E Exit limit exceeded. Allowed=*allow* Defined=*define***

**Explanation:** IMS Connect Extensions detected that too many exits were defined for the IMS Connect system. *allow* identifies the number of exits allowed and *define* the number defined in the IMS Connect configuration member.

**System action:** IMS Connect Extensions quiesces.

**User response:** Reduce the number of exits defined in the IMS Connect configuration member and restart IMS Connect.

**CEX5107W Message exit=*exitname* failed to load**

**Explanation:** During initialization processing IMS Connect Extensions attempted to load one of the message exits defined in the IMS Connect message exit list. The message exit failed to load properly. If the message has been defined to IMS Connect Extensions, the exit status is set to 'disabled' and no advanced function is supported for the message exit.

**System action:** IMS Connect Extensions quiesces.

**User response:** If the exit can be made available for IMS Connect Extensions to load, move the message exit to the proper STEPLIB data set and use the IMS Connect Extensions exit RELOAD command to load the exit.

**CEX5111E Affinity resource *name* type *type* not found on this system**

**Explanation:** IMS Connect Extensions found the Affinity resource *name* had not been defined to IMS Connect Extensions in the Definitions data set.

*type* the type of definition:

- DSG - Datastore Group
- DS - datastore
- AFL - Affinity List

**System action:** IMS Connect Extensions continues.

**User response:** Using the ISPF interface, correct the Affinity resource name for the particular definition type and stop and restart the IMS Connect system.

**CEX5112W USERID specified** *user_id1* **has been ignored in favor of the job step user ID** *user_id2*

**Explanation:** When PassTicket generation is requested, the USER parameter on the job card takes precedence over a user ID that is specified in the USERID parameter on the CONNECT card.

**System action:** Processing continues.

**User response:** Either remove or correct the USERID parameter on the CONNECT card.

**CEX5113I Datastore Member=***datastore* **has been AUTORESUMED.**

**Explanation:** A Datastore Available event was received for a datastore that was in Drained status. The datastore status also reflected that Autoresume was to occur. The datastore has been removed from Drained status and is now available for routing use.

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5114I IMS Connect Extensions inactive on this IMS Connect system** *hwsid* **as per user specification**

**Explanation:** IMS Extensions is not active for this IMS Connect system. The Activate IMS Connect Extensions option on the System Definition is "Inactive". *hwsid* identifies the IMS Connect system.

**System action:** IMS Connect continues without IMS Connect Extensions.

**User response:** None. Informational message only.

**CEX5115I In Datastore Group** *datastore_group*, **datastore** *datastore* **is the primary for system** *HWSID*

**Explanation:** Messages with the destination ID (DESTID) of the datastore group *datastore_group* will be routed to the datastore *datastore* if handled by the IMS Connect system *HWSID*, and if the datastore is available.

**System action:** IMS Connect continues.

**User response:** None. Informational message only.

**CEX5116I In Datastore Group** *datastore_group*, **datastore** *new_datastore* **has replaced** *old_datastore* **as the primary for system** *HWSID*

**Explanation:** A new datastore *new_datastore* is now the primary for the IMS Connect system *HWSID*. The datastore replaces *old_datastore* as the primary.

**System action:** IMS Connect continues.

**User response:** None. Informational message only.

**CEX5117I CEXCTLIN Command:** *option_name*=*value*

**Explanation:** This message provides a list of the values that have been set either from the contents of the CEXCTLIN data set or from the value defaults. The message is displayed once for each CEXCTLIN option.

**System action:** IMS Connect Extensions continues.

**User response:** None. Informational message only.

**CEX5118W No records found in file** *ddname*

**Explanation:** The CEXCTLIN file is empty.

**System action:** IMS Connect Extensions continues.

**User response:** Check that the CEXCTLIN statement in the IMS Connect startup job refers to the correct data set and that any required control options are specified.

**CEX5120I** *ddname* **Line** *record_number***:***input_record*

**Explanation:** This message is used to list the contents of various SYSIN data sets.

**User response:** None. Informational message only.

**CEX5125I DRAIN command AUTORESUME feature is not available until IMS Connect APAR is applied.**

**Explanation:** The DRAIN command AUTORESUME feature requires that IMS Connect publish the correct DATASTORE ID in events 16 and 17. It appears that events 16 and 17 still contain the DATASTORE TMEMBER name instead of the DATASTORE ID.

**System action:** Processing continues.

**User response:** To use the AUTORESUME feature with IMS Connect V12 systems, apply the fix for APAR PM85243. To use the AUTORESUME feature with IMS Connect V11 systems, apply the fix for APAR PM70277.

**CEX5126E Password validation failed for user** *user*, **SAF return code=** *safrc* **/** *racfrc* **/** *racfrsn*

**Explanation:** An attempt was made to validate the password, password phrase, or PassTicket. The security system rejected the request. *user* identifies the user ID, *safrc* is the SAF return code from the SAF VERIFY. *racfrc* and *racfrsn* are the return code and reason code from RACF or the installation security exit.

**System action:** The transaction is rejected.

**User response:** Rerun the transaction, specifying a valid user ID and password or password phrase. If the user ID and password or password phrase are valid, refer to the manual for your security system and check the list of SAF return codes.

**CEX5127E** **Access TO SAF resource** *safcl*,*safnm* **failed for user** *safusr*, **SAF return code=***safrc*/*racfrc*

**Explanation:** A request was made to validate access to IMS Connect system *safnm* using security class *safcl* for user ID *user*. The security system rejected the request.

*safrc* is the SAF return code from the SAF FASTAUTH call. *racfrc* / *racfrsn* is the return/reason code from RACF or the installation security exit.

**System action:** The transaction is rejected.

**User response:** If the user requires access to the IMS Connect system, update the security rules to allow access. Either restart IMS Connect or reload the security rules using the IMS Connect Extensions SECURITY command. Refer to the manual for your installation security system for a list of SAF return codes.

**CEX5128E** **ACEE cache error, Function=***function*, **Return code=***rc*, **Reason code=***rsn*

**Explanation:** A request to the ACEE cache manager failed. The return and reason codes can only be interpreted by IBM support.

**System action:** The ACEE caching feature is disabled and processing continues without caching.

**User response:** Refer to Chapter 29, "Gathering diagnostic information," on page 421, and then contact IBM.

**CEX5129W** **PassTicket generation failed for userid** *userid*, **resource class PTKTDATA, application ID** *applid*, **return code=***rc*

**Explanation:** Could not generate a PassTicket for the user ID and application ID. The warning may indicate that the user is not authorized or that there is a problem with PassTicket generation. The return code is one of the following:

**04** TESTAUTH failed.

**08** Call to RCVTPTGN failed.

**System action:** Processing continues.

**User response:** If appropriate, perform the PassTicket generation configuration steps and restart IMS Connect.

**CEX5130W** **ICON_CONTROL PORT** *port* **not defined in HWSCFG or invalid port type.**

**Explanation:** If an ICON_CONTROL PORT statement is provided in the CEXCTLIN file, the port number is checked during initialization. If the port number is not defined in the IMS Connect configuration file or if it is not a valid port type, this warning message is written to CEXPRINT.

**System action:** IMS Connect Extensions continues. PORT is set to the default value of 0. This means that the first port of the correct type that is defined in HWSCFG will be used.

**User response:** Ensure that the ICON_CONTROL PORT control option specifies a dedicated IMS Connect port which is to be used for routing IMS Type-1 commands that are issued from IMS Connect Extensions. *port* must be an OTMA port that is defined in HWSCFG. Ensure that the port is not subject to TCP/IP routing. Restart IMS Connect.

**CEX5131I** **ROUTE DRAIN|RESUME for datastore** *ds*, *autoresume_setting*

**Explanation:** One of these messages is displayed on the JES syslog to show the result when a ROUTE ACTION command is issued:

- ROUTE DRAIN for datastore *ds*, AUTORESUME is set|is not set
- ROUTE RESUME for datastore *ds*

**System action:** Processing continues.

**CEX5132W** **ROUTE RESUME for datastore** *ds*. **Datastore is not currently draining.**

**Explanation:** This message is displayed on the JES syslog when a ROUTE ACTION=RESUME command fails because the datastore is not drained or is not currently draining.

**System action:** Processing continues.

**CEX5133I** **QUERY DS(***datastore***): Routing status is** *status* **with** *count* **responses pending.**

**Explanation:** This message is displayed when a QUERY PENDING_RESPONSES command is issued against a single datastore or a routing list. *count* indicates the number of messages that are pending a response from IMS. For a routing list, one message is displayed for each datastore in the list, then followed by CEX5134I and optionally CEX5135W.

**System action:** Processing continues.

**CEX5134I** **QUERY RLIST(***routing_list***):** *count* **pending clients.**

**Explanation:** This message is displayed when a QUERY PENDING_RESPONSES command is issued against a routing list. *count* indicates the number of messages that are pending a response from IMS.

**System action:** Processing continues.

**CEX5135W QUERY RLIST(*routing_list*): Not all datastores SUSPENDED.**

**Explanation:** This message is displayed when a QUERY PENDING_RESPONSES command is issued against a routing list, and one or more of the datastores in the list is not suspended (that is, the status is not Suspended, SusCWRZero, SusAutoRes, or AutoRes).

**System action:** Processing continues.

**User response:** Check for other messages that might indicate why datastores were not suspended.

**CEX5136W No matching rule for CEXROUTE RBR_NODEST=*destid* | RBR_NOALIAS=*alias***

**Explanation:** This message is displayed either at initialization or as a result of a refresh when a matching rule cannot be found for the specified DestID or alias.

**System action:** Processing continues. Routing will be performed as if the PASS option had been specified.

**User response:** Check that the DestID or alias is correctly specified in the CEXROUTE control option. Ensure that a routing rule is defined that matches this DestID or alias. In the case of RBS_NOALIAS ensure that the matching routing rule is unqualified: that is, it does not refer to a PSB name list.

**CEX5151I The message handler exit is now ACTIVE | INACTIVE**

**Explanation:** The MESSAGE control option can be used to stop specified messages being written to CEXPRINT and to write additional messages to SYSLOG. This message is displayed in CEXPRINT to show when the message handler exit was active. Messages issued while the message handler exit is inactive cannot be suppressed or redirected.

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5187E CEXAUTH0 is configured as the Security Exit for this IMS Connect system but IMS Connect Extensions is not active**

**Explanation:** The security exit failed to connect to an active IMS Connect Extensions environment.

**System action:** The security exit is not enabled.

**User response:** If IMS Connect DB security authorization is required, ensure that IMS Connect Extensions is active.

**CEX5188I ODBM access defined, CEX Activate Security is selected, and CEXAUTH0 is defined as an ODBMAUTH exit program in the EXITDEF statement.**

**Explanation:** The following options relating to ODBM security were detected during IMS Connect Extensions initialization:

- The ODACCESS command is specified in the system configuration file.
- The Activate Security option is selected in the system definition.
- The IMS Connect Extensions ODBM security exit (CEXAUTH0) was specified in the ODBMAUTH EXITDEF statement in the PROCLIB member pointed to by the BPECFG EXITMBR statement.

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5189W IMS Connect Extensions security will not apply for ODBM workload as CEXAUTH0 is not defined in the EXITDEF statement.**

**Explanation:** The following options relating to ODBM security were detected during IMS Connect Extensions initialization:

- The ODACCESS command is specified in the system configuration file.
- The Activate Security option is selected in the system definition.

However, the IMS Connect Extensions ODBM security exit (CEXAUTH0) was not specified in the ODBMAUTH EXITDEF statement in the PROCLIB member pointed to by the BPECFG EXITMBR statement.

**System action:** Processing continues. However, ODBM security authorization will not be performed by IMS Connect Extensions.

**User response:** If you require ODBM security authorization to be performed by IMS Connect Extensions, specify CEXAUTH0 as the ODBMAUTH exit program. If not, this message can be ignored.

**CEX5190E CEXROUT0 is ODBM routing exit but CEX is no longer active. CEXROUT0 will use default routing**

**Explanation:** CEXROUT0 is the IMS Connect Extensions routing exit. CEXROUT0 is specified as the DB routing exit in the BPE configuration member and was initialized, but IMS Connect Extensions is no longer active and therefore can't perform routing.

**System action:** The input alias is copied as the output alias and the ODBM name is set to blanks. This is also the behavior of the HWSROUT0 exit supplied with IMS Connect.

**User response:** The CEXPRINT data set might contain more information about why IMS Connect Extensions is not active. If IMS Connect DB security authorization is required, ensure that IMS Connect Extensions is active.

**CEX5191I** *ODBM routing rule descriptions*

**Explanation:** This message is used to list the ODBM routing rules that are in effect following initial startup or refresh of a routing descriptor.

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5192E** **CEXROUT0 exit could not connect to a resource needed for routing. ODBM routing rules are not active. R=***resource*

**Explanation:** CEXROUT0 has been specified as the ODBM routing exit, indicating that ODBM routing is required. IMS Connect Extensions cannot perform ODBM routing because the named resource, which is needed for ODBM routing, was not available. The CEXPRINT data set might contain more information about the failure.

**System action:** CEXROUT0 uses the default routing action.

**User response:**

- If ODBM routing by IMS Connect Extensions is not required, remove CEXROUT0 from the BPE configuration member and disable ODBM routing in the system definition.
- If the CEXPRINT data set does not indicate a problem that has caused *resource* to be unavailable, contact IBM Software Support.

**CEX5193I** *ODBM target list names*

**Explanation:** This message is used to list the target list names that are being used by ODBM routing. Use this to verify that target lists and target references are correctly set up for ODBM routing.

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5194E** **CEXROUT0 is configured as the Routing Exit for this IMS Connect system but IMS Connect Extensions is not active.**

**Explanation:** CEXROUT0 is specified as the ODBM routing exit in the BPE configuration member but IMS Connect Extensions is not active. The CEXPRINT data set might contain more information about why IMS Connect Extensions is not active. This message is issued by WTO in CEXROUT0 as IMS Connect Extensions cannot issue messages while it is not active.

**System action:** CEXROUT0 uses the default routing action.

**User response:**

- Correct the problem that prevented IMS Connect Extensions from becoming active and restart IMS Connect.
- If ODBM routing by IMS Connect Extensions is not required, remove CEXROUT0 from the BPE configuration member and clear "Activate ODBM rules" in the system definition.

**CEX5195W** **CEXROUT0 is configured as the Routing Exit for this system, but ODBM Routing is not active in the System Definition**

**Explanation:** CEXROUT0 is specified as the ODBM routing exit in the BPE configuration member but IMS Connect Extensions is not able to perform ODBM routing because ODBM routing is not activated in the system definition. This message can also be issued if ODBM routing was active but CEXROUT0 detects that it is no longer active.

**System action:** CEXROUT0 uses the default routing action.

**User response:**

- If ODBM routing is required, select "Activate ODBM rules" in the system definition, and then refresh the system definition to cause ODBM routing to begin.
- If ODBM routing by IMS Connect Extensions is not required, remove CEXROUT0 from the BPE configuration member.

**CEX5196I** **Exit CEXROUT0 has detected that ODBM Routing is now active. ODBM routing rules will be used for calls to CEXROUT0.**

**Explanation:** The CEXROUT0 exit has detected that ODBM routing is now active in the system definition when it was previously inactive.

**System action:** CEXROUT0 begins routing requests using ODBM routing rules.

**User response:** None. Informational message only.

**CEX5198W** **IMS Connect Extensions routing will not apply for ODBM workload as CEXROUT0 is not defined in the EXITDEF statement.**

**Explanation:** During initialization, "Activate ODBM rules" is selected in the system definition. However, the ODBM routing exit (CEXROUT0) was not specified as the ODBMROUT exit member.

**System action:** Processing continues. However, ODBM routing will not be performed as configured in IMS Connect Extensions.

**User response:**

- If ODBM routing is required, specify CEXROUT0 as the ODBMROUT exit member. The exit member is specified in the ODBMROUT EXITDEF statement in the PROCLIB member pointed to by the EXITMBR statement in the BPECFG member.
- If ODBM routing by IMS Connect Extensions is not required, this message can be ignored.

**CEX5199W** *exit* **detected in HWS configuration file TCPIP command. Not allowed. Results unpredictable.**

**Explanation:** During initialization, CEXROUT0 or HWSROUT0 was detected in the HWS configuration file TCPIP command. This is not allowed.

**System action:** Processing continues but the results are unpredictable. It is possible that IMS Connect will abend.

**User response:** Remove any instances of CEXROUT0 and HWSROUT0 from all of the HWS configuration file TCPIP commands and restart IMS Connect.

**CEX5200I** **----** *dow*, *dd mmm yyyy* **----**

**Explanation:** Shows the current date. This entry is written to the IMS Connect Extensions log when the first activity for a new day is logged. Therefore this entry marks the start of a new days processing and can be used to search the log for messages for a particular day.

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5315W** **Invalid affinity type** *type* **for transaction** *txn*

**Explanation:** The Affinity type for a transaction is invalid. *txn* identifies the name of the transaction and *type* the type of affinity:

- 1 - All datastores
- 2 - A specific datastore
- 3 - A Datastore Group
- 4 - An Affinity List

**System action:** IMS Connect Extensions continues.

**User response:** Define a valid affinity type for the transaction.

**CEX5316I** **IMS Connect Extensions is beginning shutdown processing**

**Explanation:** The IMS Connect Extensions system has received the shutdown request from IMS Connect and has begun the orderly termination process.

**System action:** IMS Connect Extensions begins orderly termination.

**User response:** None. Informational message only.

**CEX5400E** **Unable to BIND to IP address** *ipaddress***:***port***, IMS Connect Extensions command processing unavailable**

**Explanation:** IMS Connect Extensions was unable to connect using the TCP/IP address and port number; *ipaddress* identifies the TCP/IP address and *port* identifies the port number.

Before writing this message to the log, the Listener reports the problem to the system operator and prompts for a suitable response. The format is:

```
CEX5400E Listener unable to bind to console port
correct error then reply GO to retry or STOP to
end listener
```

The system operator can use the GO command to retry or the STOP command to stop the Listener from trying to bind to the port.

**System action:** IMS Connect Extensions continues but the console is unable to connect. IMS Connect Extensions commands and Status Monitor are not available.

**User response:** Check the IP address and Port number, then stop and restart the IMS Connect system. Do not use an existing port number listed in the IMS Connect configuration member.

**CEX5401E** **Unexpected TCPIP response. IP operation was** *operation***, ERRNO was** *errno*

**Explanation:** IMS Connect Extensions experienced a bad TCP/IP response. *operation* identifies the function and *errno* the error code. A common error is:

```
CEX5401E Unexpected TCPIP response.
IP operation was INITAPI,
ERRNO was 1011
```

This error indicates that IMS Connect Extensions could not locate the TCP/IP stack on the system, probably due to a TCP/IP configuration problem.

**System action:** IMS Connect Extensions continues.

**User response:**

**If you received an INITAPI error with ERRNO 1011**

Because CEX is requesting the default TCP/IP stack, this error generally indicates that a valid

TCPDATA file cannot be found. To solve this problem, insert the following DD statement in the IMS Connect start up JCL:

```
//SYSTCPD DD DISP=SHR,DSN=tcpip.data
```

Replace *tcpip.data* with the name of the data set that contains the TCPDATA parameters for the system. This data set may be a sequential data set or a member of a partitioned data set.

SYSTCPD explicitly identifies the data set that is to be used to obtain the TCP/IP parameters and bypasses the default search order. The default search order for TCPDATA is documented in *z/OS Communications Server IP Configuration Guide*.

**For other errors**

Look up the sockets return codes (ERRNOs) in *z/OS Communications Server IP Sockets Application Programming Interface Guide and Reference*.

**CEX5402E Datastore *ds* has no datastore group defined, datastore bypassed**

**Explanation:** IMS Connect Extensions detected a datastore with no Datastore Group defined. *ds* identifies the datastore.

**System action:** IMS Connect Extensions continues.

**User response:** Using the ISPF interface update the named datastore definition and with a valid Datastore Group. To implement the change either stop and restart the IMS Connect system or refresh the datastore using the REFRESH command.

**CEX5403E IMS Connect configuration member *member* not found in PROCLIB**

**Explanation:** The IMS Connect configuration *member* was not found in the IMS Connect Proclib library defined in the startup JCL deck. IMS Connect Extensions is unable to initialize for this IMS Connect system.

**System action:** IMS Connect Extensions fails to initialize.

**User response:** Ensure that the IMS Connect configuration member in question can be found in the Proclib library named in the PROCLIB DD statement of the IMS Connect job.

**CEX5404E IMS Connect Extensions console has been lost, restart IMS Connect to recover console**

**Explanation:** An IMS Connect Extensions console module has ended abnormally. Communications with IMS Connect Extensions no longer possible.

This message also appears in your SYSLOG.

**System action:** IMS Connect Extensions waits for IMS Connect to be restarted.

**User response:** Restart IMS Connect when possible.

**CEX5405I IMS Connect Extensions task, *task_name*, has ended before shutdown. Completion code=*completion_code***

**Explanation:** An IMS Connect Extensions console module has ended abnormally. Communications with IMS Connect Extensions no longer possible.

**System action:** IMS Connect Extensions waits for IMS Connect to be restarted.

**User response:** Restart IMS Connect when possible.

**CEX5406I Listener initialized; console services now available**

**Explanation:** Shows that the console listener is active and listening on the designated port.

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5407I IMS Connect Extensions initialization complete**

**Explanation:** IMS Connect Extensions initialization complete

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5408E TCP/IP error, socket descriptor *socket* exceeds MAXSOC *maxsoc***

**Explanation:** TCP/IP has returned a socket descriptor that exceeds the MAXSOC specification in the HWS configuration member.

**System action:** The IMS Connect Extensions console listener task attempts to recover from the error. If recovery fails, then the console listener terminates, and console clients can no longer communicate with IMS Connect.

**User response:** If this error persists, review the MAXSOC specification and increase accordingly.

**CEX5409E TCP/IP error, Unix System Services callable service *func* not found**

**Explanation:** TCP/IP Unix System Services callable service *func* was not found. This is a z/OS environmental error.

**System action:** The IMS Connect Extensions listener stops and console services are lost.

**User response:** Consult your z/OS system administrator to ensure that Unix System Services has been properly installed and configured.

**CEX5410E TCP/IP network management interface activation failed, load module EZBNMIFR not found, RC=*rc*, RSN=*rsn***

**Explanation:** TCP/IP error, network management interface module EZBNMIFR was not found. This is a z/OS environmental error.

**System action:** The IMS Connect Extensions network status function is disabled.

**User response:** Consult your z/OS system administrator to ensure that the TCP/IP network management interface is properly installed and configured.

**CEX5411E TCP/IP network management request failed, RC=*rc*, RSN=*rsn***

**Explanation:** TCP/IP error, network management request failed.

**System action:** The IMS Connect Extensions network status function is disabled until IMS Connect is restarted.

**User response:** Consult your z/OS system administrator to ensure that the TCP/IP network management interface is properly installed and configured.

**CEX5412E XML error, z/OS XML System Services callable service *func* not found**

**Explanation:** z/OS XML System Services callable service *func* was not found. This is a z/OS environmental error.

**System action:** The IMS Connect Extensions console listener stops XML operations and the GUI client services are lost until the system is restarted.

**User response:** Consult your z/OS system administrator to ensure that z/OS XML System Services has been properly installed and configured.

**CEX5413E XML system services request failed, RC=*rc*, RSN=*rsn***

**Explanation:** XML system services request failed.

**System action:** The IMS Connect Extensions console listener stops XML operations and the GUI client services are lost until the system is restarted.

**User response:** Refer to the appendixes on return codes and reason codes in *z/OS XML System Services User's Guide and Reference*, SA23-1350.

**CEX5414E IMS command failed in exit HWSJAVA0 RC=*rc*, RSN=*rsn***

**Explanation:** IMS command failed in exit HWSJAVA0. This message occurs when the transport (send-receive) process of the Type-1 command fails.

**System action:** The IMS command is rejected.

**User response:** Refer to the sections on IMS Connect return codes and reason codes in *IMS Messages and Codes, Volume 4: IMS Component Codes*.

**CEX5415E Connection request failed, hostname *host*, port *port*, errno *errno***

**Explanation:** The command shell is attempting to connect to an available IMS Connect message port to submit IMS Type-1 commands using message exit HWSJAVA0. IMS Connect Extensions uses the local hostname to derive a network address. It then issues a socket connection request to the first available message port at that address. This request has failed.

**System action:** The IMS command is rejected.

**User response:** To check the TCP/IP configuration, see "IMS Type-1 command failure due to TCP/IP connection error" on page 368.

For more information about the *errno*, refer to "Return Codes (Errnos) Listed by Value" in *z/OS UNIX System Services Messages and Codes*.

**CEX5416E IMS Connect Extensions console port number *port* is in use by IMS Connect; console services are unavailable**

**Explanation:** The IMS Connect configuration member specifies a port with the same port number as the IMS Connect Extensions console port. The console port number must be unique.

**System action:** IMS Connect Extensions initialization fails. IMS Connect initialization continues.

**User response:** Assign a unique port number to the IMS Connect Extensions console port in the system definition and then restart IMS Connect.

**CEX5500I *line control statement***

**Explanation:** Echoes migration utility control statements and their line number.

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5501I *value* completed successfully**

**Explanation:** Command executed successfully.

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5502F** *value* **failed**

**Explanation:** Command failed.

**System action:** Processing continues.

**User response:** Check the data sets specified by the CEXPRINT and MSGOUT ddnames for more information on the cause of the failure.

**CEX5503W** **No records selected for** *value*

**Explanation:** The command completed successfully, but no records were actually processed. This may be due to the filtering selection criteria.

**System action:** Processing continues.

**User response:** Check MSGOUT for more information.

**CEX5504E** **Data set for DDname=***ddn* **failed to open**

**Explanation:** Could not open the named data set.

**System action:** Utility processing ends

**User response:** Verify that the DD statement points to a valid data set.

**CEX5505E** *error information*

**Explanation:** Provides information about errors found while parsing control statements for the migration utility.

**System action:** Utility processing ends

**User response:** Check the CEXPRINT DD for more information.

**CEX5506E** **Load request for module** *value* **failed; Return code** *rc***, reason code** *rsn***, info** *info*

**Explanation:** Utilities failed to load a required module.

**System action:** Utility processing ends.

**User response:** Verify that the STEBLIB data sets refer to valid IMS Connect Extensions and functional support libraries.

**CEX5507W** **VERIFY completed with warnings**

**Explanation:** The VERIFY command completed processing successfully but found one or more warnings in the repository.

**System action:** Processing continues.

**User response:** See the MSGOUT DD for additional information.

**CEX5508F** **VERIFY completed with errors**

**Explanation:** The VERIFY command completed processing successfully but found one or more errors in the definition data set.

**System action:** Processing continues.

**User response:** See MSGOUT for details of errors. Correct the error and rerun the utility

**CEX5509E** *ddn* **DD is missing**

**Explanation:** The named DD was not specified in the JCL.

**System action:** Utility processing ends.

**User response:** Specify the DD in the JCL and then retry.

**CEX5510E** **Transformation not defined. TRANSFORM=***trans*

**Explanation:** The transformation rule *trans* does not exist.

**System action:** Utility processing ends.

**User response:** Check the spelling of the transformation rule name or define the rule.

**CEX5511E** **Transformation error; Result too long. TRANSFORM=***trans***, VALUE=***value*

**Explanation:** If the named transformation is applied to the value the result is greater than the maximum allowed length

**System action:** Utility processing ends.

**User response:** Correct the transform rule.

**CEX5512E** **Transformation error; No result. TRANSFORM=***trans***, VALUE=***value*

**Explanation:** If the named transformation is applied to the value the result is of length zero.

**System action:** Utility processing ends.

**User response:** Correct the transform rule.

**CEX5513I** *phase details*

**Explanation:** This is an informational message describing the phases of IVP processing.

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5514I IMS Connect Extensions assembler testing remote client: V*version***

**Explanation:** This is an informational message showing the start of execution of the testing remote client.

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5515I Processing command, name=*name***

**Explanation:** Processing of IVP client command *name* started.

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5516I Command ended, name=*name rc rsn***

**Explanation:** Execution of command ended with the specified return and reason codes.

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5517I Refer to the CEXPRINT dataset for additional messages**

**Explanation:** Additional information about IVP execution available in the CEXPRINT data set.

**System action:** Processing continues.

**User response:** Refer to the messages in CEXPRINT (data set referred to by that ddname).

**CEX5518I Client could not connect to IMS Connect system=*name***

**Explanation:** IVP client could not connect to the specified IMS Connect system.

**System action:** Execution of batch client ends.

**User response:** Verify that the system is active and listening on the port you specify and rerun the utility.

**CEX5519I Client has connected to IMS Connect system=*name***

**Explanation:** Client has connected to the specified IMS system.

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5520E Invalid import record type detected: X'*type*'**

**Explanation:** An invalid record type/definition was detected on the input data set during import processing.

**System action:** The import request is terminated with RC=12.

**User response:** If possible, locate and correct the invalid type/definition in the input data set.

**CEX5521W EXTRACT completed with exceptions**

**Explanation:** The command to extract definitions to a CSV file completed successfully, but one of the following conditions occurred which you might want to check:

1. LOCALHOST (case independent) was found in an IP address.
2. 127.*.*.* (127 in 1st node) was found in an IP address.
3. Double quotation marks (") in the system description field were changed to single quotation marks (').

**System action:** Processing continues.

**User response:** Refer to the MSGOUT data set for additional information.

**CEX5522I received message type=*rtyp*, expected type=*etyp***

**Explanation:** The testing client received an unexpected response from IMS Connect. The type expected and the type received appear in the message. The following are the various types of responses:

| | |
|---|---|
| **ANY** | Any response |
| **CSM** | Commit status message |
| **RSM** | Request status message |
| **DCO** | Deallocate commit |
| **DAB** | Deallocate abort |
| **TMO** | OTMA timeout |
| **PNG** | Ping |
| **DFS** | DFS message |
| **RMM** | Request mod message |
| **ERR** | Unknown type: error |

**System action:** Processing continues.

**User response:** Refer to the data set specified by the CEXPRINT ddname for additional information.

**CEX5523I CSMOKY received, state information; Ack=*ack*, Conv=*conv*, Async=*asyn***

**Explanation:** The testing client has received a CSMOKY response from IMS Connect. This message serves as notification of the CSMOKY receipt and gives the values contained in the CSMOKY state flag. The state information of the flag shows whether an ack is required, a conversation is active, and whether asynchronous output is present.

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5524I RSM received, state information; Ack=*ack*, Conv=*conv*, Async=*asyn***

**Explanation:** The testing client has received a RSM response from IMS Connect. This message serves as notification of the RSM receipt and gives the values contained in the RSMY state flag. The state information of the flag shows whether an ack is required, a conversation is active, and whether asynchronous output is present.

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5525I label not found,name=*label***

**Explanation:** During execution of the testing client, a label that was the target of a branch type command could not be located. The name of the label that could not be located is shown.

**System action:** Testing client execution ends.

**User response:** This is an internal error. Contact IBM Software Support. Have the output from this job available.

**CEX5526I RSM compare error RC=*rc*, RSN=*rsn*, expected RC=*exprc*, RSN=*exprsn***

**Explanation:** The RSM received from IMS Connect does not match the expected RSM. The actual and expected RSM return codes and reason codes are shown in the message.

**System action:** Processing continues.

**User response:** Determine the cause of the problem by identifying the return codes and reason codes produced in the RSM and resubmit the IVP job.

**CEX5527I Start of formatted data**

**Explanation:** Start of formatted data (only appears when FORMAT=F).

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5528I End of formatted data**

**Explanation:** End of formatted data (only appears when FORMAT=F).

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5529I RSM Response detected, RC=*rc*, RSN=*rsn*, in decimal**

**Explanation:** The testing client has detected an RSM response from IMS Connect. The return and reason codes in the RSM are shown in the message text in decimal format.

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5530I Branched to label=*name***

**Explanation:** Successful branch to the named label.

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5531I RSM Extension has been found**

**Explanation:** IMS Connect Extensions extended RSM detected in response.

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5532I Len=*ln*, APAR=*ap*, flag=*flag*, content flag=*cont*, #message segments=*mc*, SAF request type=*reqtype***

**Explanation:** The contents of the RSM extension are formatted in three parts using the messages CEX5532I, CEX5533I, and CEX5534I. This message displays the first part. The DSECT describing the RSM is CEXERSM. Refer to this DSECT for additional information about the fields and field content included in this message.

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5533I SAF RC=*src*, SAF RSN=*srsn***

**Explanation:** The contents of the RSM extension are formatted in three parts using the messages CEX5532I, CEX5533I, and CEX5534I. This message displays the second part, which contains the SAF return code and reason code.

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5534I SAF Msg=*smsg***

**Explanation:** The contents of the RSM extension are formatted in three parts using the messages CEX5532I, CEX5533I, and CEX5534I. This message displays the third part, which contains the SAF message text. This message repeats for each text message in the RSM extension.

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5535I RSM Ending string found, len=*ln*, APAR level=*ap***

**Explanation:** Extended RSM end string found.

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5536I RSM Ending string is missing**

**Explanation:** Extended RSM end string not found.

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5537E Insufficient storage available**

**Explanation:** Insufficient storage available to complete execution.

**System action:** Client execution ends.

**User response:** Increase the region size.

**CEX5538W Error at line *line*: the HOST *host* is not a valid TCP/IP address**

**Explanation:** The Definition Maintenance utility was unable to connect to the IMS Connect system that was specified in the HOST parameter. The TCP/IP response indicated that the host name did not resolve to a valid IPv4 or IPv6 network address.

The likely explanation is one of the following:

- The host name or IP address is incorrect.
- The network connection is temporarily unavailable.
- IMS Connect Extensions is using the wrong TCP/IP stack.

**System action:** Processing continues.

**User response:**

1. Check that the host name is correct and resolves to a valid IP address. If necessary, correct the host name and retry.
2. Confirm that the host name is valid by using ISPF Option 6 to issue the PING command for the failing host name, as shown in the following example:

   ```
   PING host_name
   ```

   If the PING command is unsuccessful, then you can ignore this warning message as the network connection is temporarily unavailable. To ignore warning messages, insert the following statement at the start of the Definition Maintenance utility input:

   ```
   SET OPTION=(WARNING=IGNORE)
   ```

3. If the PING command is successful, IMS Connect Extensions might be using the wrong TCP/IP stack. For example, it might be using the default TCP/IP stack, and this stack is not in the correct path to resolve the host name.

   To eliminate the TCP/IP stack as the cause of the error message, add the following DD statement to the Definition Maintenance Utility job step:

   ```
   //SYSTCPD DD DISP=SHR,DSN=tcpip.data
   ```

   Replace *tcpip.data* with the name of the data set that contains the TCPDATA parameters to be used for this system. SYSTCPD explicitly identifies the data set that is to be used to obtain the TCP/IP parameters and bypasses the default system search order. The default search order for TCPDATA is documented in *z/OS Communications Server IP Configuration Guide*.

**CEX5539E Error at line *line*: overlap detected in custom offsets in IRM_OFFSET parameter**

**Explanation:** The IRM offsets specify fields that are 8 characters long. They must not overlap.

**System action:** Execution of testing client ends.

**User response:** Correct the IRM_OFFSET parameter and retry

**CEX5540E Error at line *line*: ARCHIVE_JCL *jcl* is invalid - *error_type***

**Explanation:** A control card error has been detected by the batch input utility. *error_type* specifies the error condition.

**System action:** Utility processing ends.

**User response:** Correct the ARCHIVE_JCL parameter and retry or use the SET command option WARNING=IGNORE to ignore this warning.

**CEX5541E Error at line *line*: ACTIVE_DS *dsn* is invalid - *error_type***

**Explanation:** The *dsn* specified for the ACTIVE_DS parameter is invalid. *error_type* specifies the error condition.

**System action:** Utility processing ends.

**User response:** Correct the ACTIVE_DS parameter and retry.

**CEX5542E Error at line *line*: ARCHIVE_DS *dsn* is invalid - *error_type***

**Explanation:** The *dsn* specified for the ARCHIVE_DS parameter is invalid. *error_type* specifies the error condition.

**System action:** Utility processing ends.

**User response:** Correct the ARCHIVE_DS parameter and retry.

**CEX5543I *value type* definition *mem***

**Explanation:** The command completed successfully.

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5544E Failed to ADD|DELETE *type* definition *mem*: member in use**

**Explanation:** The ADD or DELETE command failed to complete because the member name *mem* was already in use.

**System action:** Utility processing ends.

**User response:** Determine why the member is not available to the definition maintenance utility.

**CEX5545E IMS Connect configuration member *mbr* not found in *ddname***

**Explanation:** The IMS Connect configuration member was not found in the data set referenced by ddname.

**System action:** Utility processing ends.

**User response:** Ensure that the IMS Connect configuration member can be found in the named data set.

**CEX5546E HWS keyword not found in IMS Connect configuration member *mbr* in *ddname***

**Explanation:** The IMS Connect configuration member HWSCFG did not contain the HWS keyword.

**System action:** Utility processing ends.

**User response:** Ensure that the utility is referencing a valid HWSCFG member.

**CEX5547E No data in IMS Connect configuration member *mbr* in *ddname***

**Explanation:** The IMS Connect configuration member did not contain any statements.

**System action:** Utility processing ends.

**User response:** Ensure that the utility is referencing a valid HWSCFG member.

**CEX5548E IMS Connect configuration member *mbr* in *ddname* contains too many EXIT definitions**

**Explanation:** The IMS Connect configuration member contained more than 254 exit names in the EXIT statement

**System action:** Utility processing ends.

**User response:** Ensure that the utility is referencing a valid HWSCFG member.

**CEX5549E TCPIP keyword not found in IMS Connect configuration member *mbr* in *ddname***

**Explanation:** A required keyword (TCPIP or HOSTNAME) was not found in the IMS Connect configuration member.

**System action:** Utility processing ends.

**User response:** Ensure that the utility is referencing a valid HWSCFG member.

**CEX5550W Syntax error in IMS Connect configuration member *mbr* in *ddname***

**Explanation:** The IMS Connect configuration member contained invalid statements. Check previous messages for more information.

**System action:** Utility processing ends.

**User response:** Ensure that the utility is referencing a valid HWSCFG member.

**CEX5551W Duplicate HWSID *name* contained in IMS Connect configuration member *mbr* in *ddname***

**Explanation:** The HWSID named has already been encountered in a configuration member in a previous TAKEUP command.

**System action:** Processing continues. IMS Connect Extensions utility does not add the duplicate HWS definition.

**User response:** Check the TAKEUP commands were referencing the required HWS coinfiguration members.

**CEX5552W No definitions selected for TAKEUP**

**Explanation:** The TAKEUP command completed successfully, but no definitions were selected. This may be due to a previous TAKEUP command selecting the definitions named in the current IMS Connect configuration member.

**System action:** Processing continues.

**User response:** Check that the TAKEUP commands are referencing the correct HWS configuration members.

**CEX5553E** **Error at line** *line*: *list* **contains duplicate names**

**Explanation:** Lists of definitions must contains unique names.

**System action:** The utility stops.

**User response:** Correct the list parameter and retry.

**CEX5554E** *type* **definition** *mem* **contains unexpected data**

**Explanation:** Unexpected data was found in a record in the definition data set. The definition data set may be corrupted.

**System action:** The utility stops.

**User response:** Correct or delete the record in error and try running the utility again.

**CEX5555E** **CEX_PORT** *port* **is already assigned in HWS configuration file; the CEX console requires a unique port number assignment**

**Explanation:** IMS Connect Extensions console port already assigned in HWS configuration file; the IMS Connect Extensions console requires a unique port number.

**System action:** The utility stops.

**User response:** Assign a unique port number for the console port and run the utility again.

**CEX5556E** **Error at line** *line*: **APP override cannot be blank when definition type TXN is specified**

**Explanation:** Definition type TXN must have a valid APP override line.

**System action:** The utility stops.

**User response:** Correct the APP parameter and retry.

**CEX5557E** **Error at line** *line*: **client-server pairing not defined to z/OS Unicode Services**

**Explanation:** The client-server CCSID pairing is not defined to z/OS Unicode Services.

**System action:** Processing stopped.

**User response:** Verify that the client and server CCSID pairing has been defined in z/OS Unicode Services.

**CEX5558E** **Error at line** *line*: **client-server CCSID pairing not an ASCII-to-EBCDIC conversion**

**Explanation:** The client-server CCSID pairing is not a single byte conversion to EBCDIC.

**System action:** Processing stopped.

**User response:** Use only single-byte ASCII-like CCSIDs and single-byte EBCDIC CCSIDs.

**CEX5559E** **Error at line** *line*: **client-server CCSID pairing not a single byte character set (SBCS) conversion**

**Explanation:** CCSID pairing not a single byte character set conversion

**System action:** Processing continues.

**User response:** You can only use single-byte code sets for IMS Connect Extensions translation.

**CEX5560E** **Error at line** *line*: **z/OS Unicode Services error RC=***rc*, **RSN=***rsn*

**Explanation:** z/OS Unicode Services returned an unexpected return and reason code.

**System action:** Processing continues.

**User response:** Refer to *z/OS Support for Unicode: Unicode Services* to interpret the return and reason code information.

**CEX5561E** **Error at line** *line*: **z/OS Unicode Services interface error RC=***rc*,**RSN=***rsn*

**Explanation:** Internal error.

**System action:** Processing continues.

**User response:** Refer to Chapter 29, “Gathering diagnostic information,” on page 421, and then contact IBM.

**CEX5562E** **Error at line** *line*: **definition name** *name* **is invalid**

**Explanation:** The definition name is not in valid alphanumeric format or the generic form has been used inappropriately.

**System action:** The utility stops.

**User response:** Correct the definition name and try again.

**CEX5563I** **Duplicate** *type* **definition at line***line*: **definition** *name action*

**Explanation:** Duplicate definitions encountered. Existing definitions may be replaced, depending on the REPLACE option.

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5564E Error at line *line*: z/OS Unicode Services inactive, callable service routine not found**

**Explanation:** Message translation activation has failed for the user exit. The CEX Unicode Services interface could not find the Unicode Services callable service routine.

If stub routine CUNLCNV is statically linked then this is probably due to a linkedit error in SMP/E during product installation.

If the stub routine is dynamically located via the CVT then this is due to a z/OS environmental error.

**System action:** Processing stopped.

**User response:** If stub routine CUNLCNV is statically linked: verify that the Unicode Services library ACUNMOD (SYS1.ACUNMOD) has been defined to SMP/E and that no errors have occurred during product installation.

If the stub routine is dynamically located via the CVT: consult your z/OS system administrator to ensure that Unicode Services has been properly installed and configured.

**CEX5565E Refresh definition type *type* is invalid**

**Explanation:** The definition type specified in the refresh command is invalid.

**System action:** The utility stops.

**User response:** Correct the definition type and try again.

**CEX5566E Refresh name is required for definition type *type***

**Explanation:** A definition name is required for this definition type.

**System action:** The utility stops.

**User response:** Correct the definition name and try again.

**CEX5567E Refresh name is not permitted for definition type *type***

**Explanation:** A definition name is not permitted for this definition type.

**System action:** The utility stops.

**User response:** Remove the definition name and retry.

**CEX5568E Refresh definition name *name* is invalid for definition type *type***

**Explanation:** The definition name specified in the refresh command is invalid. *Name* must be a valid member name or, in the case of type TXN, a valid generic member name.

**System action:** The utility stops.

**User response:** Correct the definition name and try again.

**CEX5569E Invalid MODIFY command data. MODIFY command ignored.**

**Explanation:** The only supported command is SHUTDOWN.

**System action:** The MODIFY command is ignored.

**User response:** Reenter the MODIFY command.

**CEX5570I STOP/SHUTDOWN command detected. Shutting down.**

**Explanation:** IMS Connect Extensions detected a STOP (P) command or a MODIFY command specifying SHUTDOWN from the system console while waiting for the target IMS Connect system to become active.

**System action:** The job step is terminated normally.

**User response:** None. Informational message only.

**CEX5571W Error at line *line*: *archive_jcl* data set not cataloged**

**Explanation:** The ARCHIVE_JCL data set could not be found by the batch input utility. *line* is the line number in the SYSIN data set of the command in error. *archive_jcl* is the name of the archive JCL skeleton.

**System action:** If the WARNING=ABORT option is set, the batch input utility ends. This is the default.

If the WARNING=IGNORE option is set, the batch input utility continues to execute.

**User response:** Correct the ARCHIVE_JCL parameter and retry or use the SET command option WARNING=IGNORE to ignore this warning.

**CEX5572W Warnings issued and WARNING=IGNORE specified, processing completed**

**Explanation:** Warning messages have been issued and the option to ignore warnings has been specified.

**System action:** Return code 4 is set and processing is completed.

**User response:** None.

**CEX5573E Warnings issued and WARNING=ABORT option specified, processing terminated**

**Explanation:** Warning messages have been issued and the option to abort on warnings has been specified.

**System action:** Return code 12 is set and processing is terminated.

**User response:** If required, change the option to WARNING=IGNORE.

**CEX5574E Error at line *line*: *keyword* is invalid**

**Explanation:** The batch utility detected an invalid keyword value while processing an ADD command.

**System action:** Utility processing ends.

**User response:** Correct the invalid keyword value and rerun the utility.

**CEX5575I IMS Connect has closed the session,RSN=*rsn***

**Explanation:** The IVP program detected that IMS Connect has closed the session. The message contains a reason code indicating how the session closure was detected.

- RSN=1: The IVP received a 0 length record.
- RSN=2: The IVP received TCPIP ERROR number 54.

**System action:** The IVP utility has detected that IMS Connect has closed the session.

**User response:** None. Informational message only.

**CEX5576F CEXCSVFI and SYSIN DD statements are mutually exclusive**

**Explanation:** The definition extract utility (CEXBDSO0) detected both a CEXCSVFI DD statement and a SYSIN DD statement. This is not allowed.

**System action:** The utility stops.

**User response:** Remove whichever DD statement is not necessary for the desired function and rerun the job.

**CEX5577E CEXTECL1 could not connect task=*task_name* to service=*service*, RC= *rc*, RSN= *rsn***

**Explanation:** An IMS Connect task could not be connected to a required IMS Connect Extensions service.

**System action:** Processing continues, though some loss of function is likely.

**User response:** There is no action that a customer can take to correct this issue. Contact IBM Software Support.

**CEX5578I Attempting to dynamically add datastore=*datastore***

**Explanation:** IMS Connect Extensions detected that the datastore shown in the message has been dynamically added to IMS Connect. IMS Connect Extensions will try to dynamically add the datastore to the in-memory list of active datastores that are available for use with IMS Connect Extensions.

**System action:** Processing continues.

**User response:** Look for a corresponding completion message. CEX5579I is issued if the datastore was added successfully. Message CEX5580W is issued if the add failed.

**CEX5579I Dynamic addition for datastore=*datastore* was successful, cached datastore credentials were used|were not used**

**Explanation:** The datastore shown in the message was successfully added to the in-memory list of active datastores that are available for use with IMS Connect Extensions. If the datastore credentials are still cached from when the datastore was originally added to IMS Connect, they are used. If the cached datastore credentials are not available, the details supplied in the ADD host command, including the tmember name and optionally the super member name, are used.

**System action:** The datastore becomes available for routing and for viewing in the Status Monitor.

**User response:** If the new datastore has been defined in one or more routing rules, refresh the in-memory OTMA routing rules using the REFRESH host command or Refresh Commands dialog.

If cached datastore credentials were not used, the credentials shown in some IMS Connect Extensions displays and event records might be incomplete. Fully define the datastore in both IMS Connect and the IMS Connect Extensions repository to ensure that the correct credentials are shown after the next IMS Connect restart.

**CEX5580W Datastore=*datastore* was not added because *reason*, datastore credentials were cached|were not cached**

**Explanation:** IMS Connect Extensions tried to add *datastore* to the in-memory list of active datastores but the action failed. The following conditions can be reported:

- **datastore is already added**. A datastore of the same name already exists in the in-memory datastore table.
- **datastore is not defined in repository**. A definition for the datastore must already exist in the IMS Connect Extensions repository when the datastore is dynamically added in IMS Connect. If there is no

such datastore definition then a corresponding entry cannot be added to the IMS Connect Extensions in-memory datastore table.

- **datastore table is full**. There was not enough space reserved for dynamically added datastores in the Datastore table.
- **of repository read error | internal logic error**.

The remainder of the message indicates whether IMS Connect Extensions was able to cache the IMS Connect datastore credentials for later use by the ADD DS host command.

**System action:** Processing continues.

**User response:**

- If the datastore already exists and is correctly defined no action is required.
- If the datastore is not already defined in the IMS Connect Extensions repository, add the definition now using the ADD DS host command.
- If the datastore table is full, increase the value of the AUTOADD_DATASTORE control option and restart IMS Connect. Also consider defining new datastores in the IMS Connect configuration member so that they do not have to be dynamically added again.
- In the event of a repository read error or internal logic error, contact IBM Software Support.

**CEX5581E An attempt to create|delete a z/OS enclave failed. zIIP processing inactive. RC=** *rc*, **RSN=** *rsn*

**Explanation:** The message comes in two forms:

1. IMS Connect Extensions attempted to create a z/OS enclave for zIIP offload processing and the attempt failed. The message contains the return and reason codes from the create enclave macro.
2. IMS Connect Extensions attempted to delete a z/OS enclave that was created to be used for zIIP offload processing and the attempt failed. The message contains the return and reason codes from the create enclave macro.

**System action:** In the first case, processing continues but zIIP offload processing is not performed. In the second case, the z/OS enclave is not deleted and processing continues.

**User response:** Refer to *z/OS MVS Programming: Workload Management Services* for an explanation of the return and reason codes associated with the IWM4ECRE (enclave creation) or IWM4EDEL (enclave deletion) macros. If you can correct the problem, do so and restart IMS Connect. If you are unsure about what is causing the failure, contact IBM Software Support.

**CEX5582I z/OS enclave created for zIIP processing.**

**Explanation:** zIIP offload processing has been requested and the required z/OS enclave has been created to support zIIP processing.

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5583I Attempting to dynamically add port=***port*

**Explanation:** IMS Connect Extensions detected that the port number shown in the message has been dynamically added to IMS Connect. IMS Connect Extensions will try to dynamically add the port definition to the list of active ports in the IMS region.

**System action:** Processing continues.

**User response:** Look for a corresponding completion message. CEX5584I is issued if the port was added successfully. Message CEX5585W is issued if the add failed or was only partially successful.

**CEX5584I Dynamic addition for port=***port* **was successful**

**Explanation:** The port number shown in the message was successfully added to the list of active ports that are available for use with IMS Connect Extensions.

**System action:** Processing continues.

**User response:** Consider updating the IMS Connect Configuration member to make this port permanently available.

**CEX5585W Port=***port condition reason*

**Explanation:** IMS Connect Extensions tried to add the port number shown in the message to the list of active ports but the action failed or was only partially successful. The following conditions can be reported:

- port=*port* was not added because the port already exists.
- port=*port* was not added because the port table is full.
- port=*port* was not added because of an internal error.

**System action:** Processing continues.

**User response:**

- If the port already exists and is correctly defined no action is required.
- If the port table is full, define all dynamically added ports to the IMS Connect configuration member using the PORT statement or PORTID statement, then restart IMS Connect. This will make room for additional dynamic ports.
- If an internal error occurred, contact IBM Software Support.

**CEX5586E** **zIIP error:** *error_condition* **Routine=** *routine*, **RSN=** *rsn*, **offload processing is suspended**

**Explanation:** A routine or process needed to perform zIIP offload processing has failed. *error_condition* can be one of the following conditions:

- SRB abnormally terminated: The SRB used for zIIP offload has been scheduled and has abended.
- SRB failed to schedule: An attempt to schedule the SRB needed to perform zIIP offload processing failed.
- SRB not responding: The SRB used for zIIP offload has been scheduled but has not responded that it has completed processing.
- SRB process encountered an error: The SRB used for zIIP offload has been scheduled and reported it was not able to complete processing.

*routine* is the name of the routine or process in which the error occurred.

**System action:** If possible, zIIP offload processing is turned off and processing continues using task mode.

**User response:** If the information in the message indicates an error that can be fixed by the customer, do so, otherwise contact IBM Software Support.

**CEX5587I** **Comp Code=** *srb_ccode* **EPSW=***error_psw*

**Explanation:** One of the SRB routines used to provide zIIP offload ended abnormally. The SRB completion code and error PSW are shown in the message. An accompanying message CEX5588I shows the contents of the general registers at the time of the abend.

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5588I** **R***m* **- R***n* *xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx*

**Explanation:** This message follows message CEX5587I and displays the contents of the general registers at abend of the SRB.

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5589E** **The Functional Service libraries are not at the required service level. zIIP offload processing suspended.**

**Explanation:** The Functional Service libraries that are needed to support zIIP offload processing are not at the required service level.

**System action:** Processing continues but zIIP offload processing is not performed.

**User response:** Upgrade the Functional Service libraries to APAR level PM88073 or later.

**CEX5590E** **zIIP offload processing could not be activated. RC=** *rc*, **RSN=** *rsn*

**Explanation:** zIIP offload processing was requested but was not activated due to an error returned from offload activation.

**System action:** Processing continues but zIIP offload processing is not performed.

**User response:** This error cannot be fixed by the customer. The return and reason codes provided in the message are intended for service personnel. Contact IBM Software Support.

**CEX559II** **Job TCB time** *tcb-time* **Enclave zIIP time** *ziip-time* **with zIIP offload active|inactive**

**Explanation:** This message reports processor usage times. Times are in the format *hh:mm:ss.xxxxxx*

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5592E** **Security violation, no RACROUTE call RSN=***rsn*, **UID=***user_id*

**Explanation:**

IMS Connect Extensions security is active but no RACROUTE call was issued because the user ID, password, or password phrase associated with the security request was blank or invalid:

- RSN=04110010: the user ID was blank or invalid
- RSN=04110011: the password was blank or invalid
- RSN=04110014: the password phrase was invalid

**System action:** Processing continues.

**User response:** Correct the client application that sent the invalid user ID, password, or password phrase.

**CEX5593E** **Error at line** *line*: *listtype listname error*

**Explanation:** One of the following errors was detected in how a routing rule qualifier list was specified:

- The qualifier list **has no entries**. A qualifier list must contain at least one entry.
- The qualifier list **has a null entry** or it **has a bad entry** *entry*. A qualifier list must contain one transaction code per line. The list can contain fully qualified transaction names and generic transaction names. A generic transaction name consists of one or more characters followed by a trailing asterisk (*). Embedded asterisks are not allowed in generic transaction names.

*line* is the line number in the SYSIN data set for the command in error. *listtype* is the type of the qualifier list that is being defined, for example TRANLIST for a transaction name list. *listname* is its name. *entry* is the invalid entry.

**System action:** Utility processing ends.

**User response:** Correct the qualifier list definition and rerun the utility.

**CEX5594E Sdump failed, RC=***rc*

**Explanation:** An attempt by an FRR or ESTAE routine to take a dump of a failure was not successful. *rc* is the return code from the SDUMP request.

**System action:** Error processing continues depending upon the error condition.

**User response:** Correct the issue that caused the SDUMP request to fail.

**CEX5595I zIIP offload processing has been activated.**

**Explanation:** zIIP offload processing has been requested and was successfully activated.

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5596W No zIIP processor is currently available.**

**Explanation:** zIIP offload processing has been requested and was activated but no zIIP processors are available.

**System action:** Processing continues.

**User response:** Bring zIIP processors online.

**CEX5597E Could not access the plan control member; member is in use.**

**Explanation:** The plan control member is used to indicate the name of the active routing plan for an IMS Connect. A problem was encountered when trying to access the plan control member in the repository. The member that needs to be added or updated is in use by another user.

**System action:** Processing continues. The action taken by IMS Connect Extensions depends on which routine was trying to add or update the plan control member.

**User response:** Release the member and try the action again.

**CEX5598I The OTMA|ODBM routing plan name** ***plan*** **is active.**

**Explanation:** The OTMA or ODBM routing plan named *plan* has been successfully activated. *plan* specifies a group of routing rules that are intended to operate at the same time.

Routing rules of this type (OTMA or ODBM) that are not assigned to a routing plan are also in effect. Where there is a conflict between an unassigned routing rule and a routing rule that is explicitly assigned to the routing plan that is currently set, the rule that is assigned to *plan* takes precedence.

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5599I Routing plan status at IMS Connect start|after SET PLAN command**

**Explanation:** This message indicates the status of the routing plans. It can be issued at the following processing checkpoints:

- when IMS Connect starts up
- following a SET PLAN command

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX5600E Could not update the datastore|ODBM target|system member=** *member*, **RC=** *rc*, **RSN=** *rsn*

**Explanation:** A problem was encountered when trying to update the repository member needed for the UPDATE command. IMS Connect Extensions could not access the member in the repository.

**Note:** Internal reason codes are not documented here as they are used by IBM to debug internal errors. Only IBM can interpret these codes.

**System action:** The UPDATE command stops.

**User response:** Contact IBM Software Support. Have the RC and RSN codes from this message available.

**CEX5601E Could not add or update the plan control member; LMM error. RC=** *rc*, **RSN=** *rsn*

**Explanation:** The plan control member is used to indicate the name of the active routing plan for an IMS Connect. A problem was encountered when trying to add or update the plan control member to the repository. The status of the plan control member cannot be determined.

**Note:** Internal reason codes are not documented here as they are used by IBM to debug internal errors. Only IBM can interpret these codes.

**System action:** Processing continues. The action taken by IMS Connect Extensions depends on which routine was trying to add or update the plan control member.

**User response:** Contact IBM Software Support. Have the RC and RSN codes from this message available.

**CEX5602W The OTMA | ODBM routing plan name *plan* is not defined and has been deactivated.**

**Explanation:** The OTMA or ODBM routing plan name in the plan control member is not found in the repository. This situation could occur if the routing plan member is deleted after it is made active by a SET PLAN command. This message is issued at IMS Connect Extensions startup or as a result of a REFRESH command.

**System action:** Processing continues. No routing plan of this type (OTMA or ODBM) is currently active.

**User response:** Verify that all required routing plans are defined in the repository. If you want a routing plan to take effect, activate it in the system definition or by using the SET PLAN command and then use the REFRESH command to refresh the OTMA or ODBM routing rules.

**CEX5603I There is no active OTMA | ODBM routing plan.**

**Explanation:** On this IMS Connect, there is no active routing plan of this type (OTMA or ODBM). This message can be issued at the following processing checkpoints:

- when IMS Connect starts up
- following a SET PLAN command

**System action:** Processing continues. For routing rules of this type (OTMA or ODBM), only rules that are not assigned to a routing plan are in effect.

**User response:** None. Informational message only.

**CEX5604E The OTMA | ODBM routing plan *plan* is not valid | is not defined.**

**Explanation:** The routing plan specified on a SET PLAN command is in error. The name *plan* either includes invalid characters or it is not defined.

**System action:** The command fails.

**User response:** Correct the routing plan name and reissue the SET PLAN command.

**CEX5605E Could not read the OTMA | ODBM routing plan member *plan*; LMM error. RC= *rc*, RSN= *rsn***

**Explanation:** The plan control member is used to define a routing plan name. A problem was encountered after a SET PLAN command when trying to read the routing plan member.

**Note:** Internal reason codes are not documented here as they are used by IBM to debug internal errors. Only IBM can interpret these codes.

**System action:** The SET PLAN command stops.

**User response:** Contact IBM Software Support. Have the RC and RSN codes from this message available.

**CEX5606E Could not update the datastore | ODBM target | system member=*member*; member is in use.**

**Explanation:** The definition of a datastore, ODBM target, or system is contained in a member. A problem was encountered when an UPDATE command tried to update a datastore, ODBM target, or system member in the repository. The member is in use by another user.

**System action:** The UPDATE command stops.

**User response:** Release the member and try the action again.

**CEX5607E Could not add datastore member *member*, *error_condition***

**Explanation:** The ADD command adds an entry to the IMS Connect Extensions repository for a datastore that was dynamically defined to IMS Connect. A problem was encountered when trying to add the named datastore. *error_condition* can be one of the following:

- LMM access error (message text includes return code (RC= *rc*) and reason code (RSN= *rsn*))
- already defined to CEX
- not defined to IMS Connect
- datastore name is blank
- datastore name is nulls
- error reading datastore member (message text includes return code (RC= *rc*) and reason code (RSN= *rsn*))
- datastore is not defined to CEX
- datastore add process failed
- write of event record failed
- Tmember name is blank
- Tmember name is nulls
- XCF Group name is blank
- XCF Group name is nulls

**System action:** The ADD command stops.

**User response:** Contact IBM Software Support. Report the RC and RSN codes if they are included in the message.

**CEX5608E Could not update the datastore | ODBM target | system member** *member* **because the member is not defined**

**Explanation:** The UPDATE command was unable to update the definition. No member of that name and type (datastore, ODBM target, or system) was found in the repository.

**System action:** The UPDATE command stops.

**User response:** If the member name was specified incorrectly, correct it and reissue the UPDATE command.

**CEX5613I There are no eligible persistent sessions to drain**

**Explanation:** A DRAIN session command was received but there were no persistent sessions that were eligible to be drained. The criteria for determining eligible sessions are explained in "DRAIN" on page 252.

**System action:** The DRAIN command ends successfully.

**User response:** If you believe there are eligible sessions that should have been drained, contact IBM Software Support.

**CEX5614I The number of sessions drained is** *nnnnnnnnn*

**Explanation:** This message displays the number of eligible sessions that were drained as a result of the DRAIN SESSION command.

**User response:** None. Informational message only.

**CEX5615E The DRAIN TYPE=SESSION command failed due to an unexpected condition RC=** *rc*, **RSN=** *rsn*

**Explanation:** While processing the DRAIN TYPE=SESSION command an unexpected error was encountered.

**Note:** Internal reason codes are not documented here as they are used by IBM to debug internal errors. Only IBM can interpret these codes.

**System action:** Processing continues.

**User response:** Contact IBM Software Support. Have the RC and RSN codes from this message available.

**CEX5616I The pre-routing exit** *exit_status*

**Explanation:** The pre-routing exit is loaded. *exit_status* can be one of the following:

- was loaded and is active
- was loaded and is inactive
- requested that the exit be suspended

  This message is issued when the pre-routing exit was loaded but failed to initialize. The return code and reason code are appended.

**System action:** Processing continues.

**User response:** Look up the reason code and take appropriate action to fix the code in the pre-routing exit.

**CEX5617E Get for pre-routing exit failed, exit is suspended, RC=** *rc*, **RSN=** *rsn*

**Explanation:** The exit workarea that is needed to call the pre-routing exit could not be obtained. The pre-routing exit has been suspended and will not be called for any task, even if that task already has a workarea.

**System action:** Processing continues but the pre-routing exit is no longer called.

**User response:** If the storage could not be obtained due to the region size, increase the region size. The recommended region size is 0 MB.

**CEX5618E Drain session failed:** *session_identifier reason*

**Explanation:** A DRAIN session command failed because a matching session could not be found. *session_identifier* can be either **initial client id=***IclientID* or **session token=***token*.

*reason* can be one of the following:

- could not be found
- is not an OTMA session
- is not a persistent session
- session is in an IMS conversation
- session is for RTPIPE

**System action:** The specified session is not drained.

**User response:** Verify that the *session_identifier* used with the DRAIN command is correct. If the session is identified by client ID, you must use the initial client ID value, not the current client ID. If the session is identified by session token, the token must match the session token returned by the QUERY TYPE=SESSIONS command. If *session_identifier* was specified incorrectly, correct it and reissue the DRAIN command.

**CEX5619W Drain session failed:** ***session_identifier*** **is already drained**

**Explanation:** A DRAIN session command failed because the session was already in Drain status. *session_identifier* can be either **initial client id=***IclientID* or **session token=***token*.

The DRAIN command requests that the session be closed at the next input message. If the remote client does not send messages on a regular basis, the session might not drain for an extended period of time.

**System action:** Processing continues.

**User response:** It might be necessary to cancel the client session if the session does not drain within a required time period.

**CEX5620E The pre-routing exit** *exit_status*

**Explanation:** The pre-routing user exit is loaded during IMS Connect Extensions initialization. This message uses *exit_status* to report the following errors that can occur while the exit is loading or being initialized:

- failed to load, exit name=CEXRBUXT
- failed to initialize and returned RC= *rc*, RSN= *rsn*

**System action:** Processing continues. The pre-routing exit is not active.

**User response:** If possible, correct any problem in the pre-routing exit code and reload the exit. Refer to "Defining user exits" on page 204 for information about installing and maintaining user exits.

**CEX5621E ENF listener exit abended, code=***abend_code*

**Explanation:** The ENF listener exit used to monitor changes to user IDs has abended.

**System action:** Purging of cached ACEEs based on ENF 71 notifications is disabled.

**User response:** Contact IBM Software Support.

**CEX5622W ENF 71 events have been lost. All cached ACEEs have been purged.**

**Explanation:** A virtual storage shortage caused a problem identifying which ACEEs should be purged.

**System action:** All cached ACEEs are purged.

**User response:** Review the region size and increase it if necessary.

**CEX5623E Extract SETROPTS failed: SAFRC=***safrc***, RACRC=***racrc***, RACSC=***racsc***. Mixed-case support set from system definition.**

**Explanation:** A SAF call was issued to return the setting for mixed-case password support for this IMS Connect but this call failed. The R_admin function ADMN_XTR_SETR returned the reported SAF and RACF codes.

**System action:** The setting for mixed-case password support in the system definition is used.

**User response:** Contact IBM Software Support.

**CEX5624E MIXDCASE field was not provided by extract SETROPTS call. Mixed-case support set from system definition.**

**Explanation:** A SAF call was issued to return the setting for mixed-case password support for this IMS Connect. The SETROPTS returned by the R_admin function ADMN_XTR_SETR did not include field information for the MIXDCASE option.

**System action:** The setting for mixed-case password support in the system definition is used.

**User response:** Contact IBM Software Support.

**CEX6000E UNKNOWN MESSAGE CODE:** *msg*

**Explanation:** Module CEXRXENV or CEXRXADR tried to issue a message code that was not found in the message table.

**System action:** Initialization of the CEX REXX host environment or processing of the CEX REXX host command is terminated.

**User response:** Contact IBM Software Support.

**CEX6001E FAILED TO LOAD MODULE:** *module*

**Explanation:** *module* failed to load when CEXRXENV initialized the CEX REXX host environment.

**System action:** Initialization of the CEX REXX host environment stops.

**User response:** Contact IBM Software Support.

**CEX6002E BAD PARAMETER:** *parm*

**Explanation:** Module CEXRXENV was invoked with the unknown parameter *parm*. Valid parameters are INIT and TERM.

**System action:** Invocation of the CEX REXX host environment stops.

**User response:** Correct and rerun the REXX exec.

**CEX6003E STORAGE OBTAIN ERROR**

**Explanation:** A storage obtain error occurred when CEXRXENV initialized the CEX REXX host environment.

**System action:** Initialization of the CEX REXX host environment stops.

**User response:** Contact IBM Software Support.

**CEX6004E SERVICES INITIALIZATION ERROR, RC=** *rc***, RSN=** *rsn*

**Explanation:** Return code *rc* was received when CEXRXENV initialized the CEX REXX host environment.

**System action:** Initialization of the CEX REXX host environment stops.

**User response:** Contact IBM Software Support. Have the RC and RSN codes from this message available.

**CEX6005E SERVICES TERMINATION ERROR, RC=** *rc***, RSN=** *rsn*

**Explanation:** Return code *rc* was received when CEXRXENV terminated the CEX REXX host environment.

**System action:** Processing of the termination request is ended.

**User response:** Contact IBM Software Support. Have the RC and RSN codes from this message available.

**CEX6006E CEX REXX ENVIRONMENT NOT FOUND**

**Explanation:** CEXRXENV was called to end the REXX environment but no CEX REXX host environment exists.

**System action:** Processing stops.

**User response:** Correct and rerun the REXX exec.

**CEX6007E UNEXPECTED IRXSUBCM ERROR, RC=** *rc*

**Explanation:** Return code *rc* was returned by IRXSUBCM request.

**System action:** Processing stops.

**User response:** Contact IBM Software Support. Have the RC from this message available.

**CEX6008E CEX REXX environment already exists**

**Explanation:** CEXRXENV was called to initialize the REXX environment but a CEX REXX host environment already exists. This would happen if the `address LINK "CEXRXENV INIT"` statement is coded twice.

**System action:** The attempt to initialize the CEX REXX host environment is ended.

**User response:** Correct and rerun the REXX exec.

**CEX6009E Cache services initialization error, RC=** *rc***, RSN=** *rsn*

**Explanation:** Return code *rc* was received when CEXRXENV initialized the CEX REXX host environment.

**System action:** Initialization of the CEX REXX host environment stops.

**User response:** Contact IBM Software Support. Have the RC and RSN codes from this message available.

**CEX6010E Cache services connect error, RC=** *rc***, RSN=** *rsn*

**Explanation:** Return code *rc* was received when CEXRXENV initialized the CEX REXX host environment.

**System action:** Initialization of the CEX REXX host environment stops.

**User response:** Contact IBM Software Support. Have the RC and RSN codes from this message available.

**CEX6011E TCP/IP EZASMI INITAPI error, RC=** *rc***, RSN=** *rsn*

**Explanation:** Return code *rc* was received when CEXRXENV initialized the CEX REXX host environment.

**System action:** Initialization of the CEX REXX host environment stops.

**User response:** Contact IBM Software Support. Have the RC and RSN codes from this message available.

**CEX6012E TCP/IP EZASMI TERMAPI error, RC=** *rc***, RSN=** *rsn*

**Explanation:** Return code *rc* was received when CEXRXENV initialized the CEX REXX host environment.

**System action:** Initialization of the CEX REXX host environment stops.

**User response:** Contact IBM Software Support. Have the RC and RSN codes from this message available.

**CEX6013E TCP/IP EZASMI FREEADDRINFO error, RC=** *rc***, RSN=** *rsn*

**Explanation:** Return code *rc* was received when CEXRXENV initialized the CEX REXX host environment.

**System action:** Initialization of the CEX REXX host environment stops.

**User response:** Contact IBM Software Support. Have the RC and RSN codes from this message available.

**CEX6014E Cache services get next error, RC=** *rc*, **RSN=** *rsn*

**Explanation:** Return code *rc* was received when CEXRXENV initialized the CEX REXX host environment.

**System action:** Initialization of the CEX REXX host environment stops.

**User response:** Contact IBM Software Support. Have the RC and RSN codes from this message available.

**CEX6015E Message format services error, RC=** *rc*, **RSN=** *rsn*

**Explanation:** Return code *rc* was received from the $MSG macro.

**System action:** Processing of the CEX REXX host command stops.

**User response:** Contact IBM Software Support. Have the RC and RSN codes from this message available.

**CEX6016E Message stack services error, RC=** *rc*, **RSN=** *rsn*

**Explanation:** Return code *rc* was received from the $MSGK macro.

**System action:** Processing of the CEX REXX host command stops.

**User response:** Contact IBM Software Support. Have the RC and RSN codes from this message available.

**CEX6017E Command parsing error, RC=** *rc*, **RSN=** *rsn*

**Explanation:** Return code *rc* was received from the $PRS macro.

**System action:** Processing of the CEX REXX host command stops.

**User response:** Contact IBM Software Support. Have the RC and RSN codes from this message available.

**CEX6018E Command exceeds maximum length**

**Explanation:** The command string exceeded the maximum of 256 characters.

**System action:** Processing of the CEX REXX host command stops.

**User response:** Reduce the command string to 256 characters or fewer.

**CEX6019E Call to IKJEFTSR failed, RC=** *rc*, **RSN=** *rsn*

**Explanation:** IKJEFTSR is used to invoke the CEXPTGEN module to run authorized. The return code from IKJEFTSR was greater than 4.

**System action:**

**User response:** See “Return Codes from IKJEFTSR” and “Reason Codes from IKJEFTSR” in *TSO/E Programming Services*.

**CEX6020E No connection found with CONID=***conid*

**Explanation:** The connection that was specified in the CONID parameter was not found in the connection cache.

**System action:** Processing of the CEX REXX host command stops.

**User response:** Ensure that a previous CONNECT command is issued to establish the CONID before it is referenced by other host commands. If the CONNECT command failed, fix it and reissue the REXX exec.

**CEX6021E Either the DATASTORE or DSLIST option must be provided with the** *cmd_name* **command**

**Explanation:** The ROUTE and QUERY PENDING_RESPONSES host commands require either a DATASTORE or DSLIST option.

**System action:** Processing of the CEX REXX host command stops.

**User response:** Reissue the command using either the DATASTORE or DSLIST option.

**CEX6022E The DATASTORE and DSLIST options are mutually exclusive on the** *cmd_name* **command**

**Explanation:** The ROUTE and QUERY PENDING_RESPONSES host commands require either a DATASTORE or DSLIST option.

**System action:** Processing of the CEX REXX host command stops.

**User response:** Reissue the command using either the DATASTORE or DSLIST option.

**CEX6023E Either the CLIENTID or TOKEN option must be provided with the DRAIN command**

**Explanation:** Either the CLIENTID or TOKEN option is required to identify the session or sessions that are to be drained.

**System action:** Processing of the CEX REXX host command stops.

**User response:** Reissue the command using either the CLIENTID or TOKEN option.

**CEX6024E The CLIENTID and TOKEN options are mutually exclusive on the DRAIN command**

**Explanation:** Either the CLIENTID or TOKEN option is required to identify the session or sessions that are to be drained.

**System action:** Processing of the CEX REXX host command stops.

**User response:** Reissue the command using either the CLIENTID or TOKEN option.

**CEX6025E The TOKEN must be either an asterisk or a valid SVT token**

**Explanation:** The TOKEN option on the DRAIN command identifies sessions that are to be drained. `TOKEN=*` marks all eligible active sessions for closure.

**System action:** Processing of the CEX REXX host command stops.

**User response:** Reissue the command using either a valid SVT TOKEN or *.

**CEX6026E One and only one target option can be specified on the UPDATE command**

**Explanation:** The UPDATE command requires one option from the following list to be specified:

- DS
- ODT
- COLLECTION_LEVEL
- SES_MSG_LIMIT
- SES_MSG_THRESHOLD

**System action:** Processing of the CEX REXX host command stops.

**User response:** Reissue the command using one of the supported options.

**CEX6027E You can only change the KEEP_ACTIVE option when you activate the trace.**

**Explanation:** The **TRACE** command KEEP_ACTIVE option can only be set or cleared when tracing is activated.

**System action:** The **TRACE** command stops.

**User response:** If the KEEP_ACTIVE option must be changed, stop and then restart tracing.

**CEX6028I Dynamic deletion for port=*port* was successful**

**Explanation:** *port* was successfully deleted from the list of active ports available for use.

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX6029W Port=*port* was not deleted because *reason*.**

**Explanation:** IMS Connect Extensions tried to delete *port* from the list of active ports but the action failed for one of the following reasons:

- *port* was not found in the list of active ports
- internal storage error

**System action:** Processing continues.

**User response:** In the event of an internal storage error, contact IBM Software Support.

**CEX6030I Dynamic update for port=*port* was successful**

**Explanation:** *port* was successfully updated.

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX6031W Port=*port* was not updated because *port* was not found in the list of active ports.**

**Explanation:** A command was issued to update *port* but it was not found in the in-memory list of ports.

**System action:** Processing continues.

**User response:** Correct the port number and try again.

**CEX6032I Dynamic deletion for datastore=*datastore* was successful.**

**Explanation:** *datastore* was successfully deleted from the list of active datastores available for use.

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX6033W Datastore=*datastore* was not deleted because *reason*.**

**Explanation:** IMS Connect Extensions tried to delete *datastore* from the list of active datastores but the action failed for one of the following reasons:

- *datastore* was not found in the list of active datastores
- internal storage error

**System action:** Processing continues.

**User response:** In the event of an internal storage error, contact IBM Software Support.

**CEX6034I Dynamic update for datastore=*datastore* was successful.**

**Explanation:** *datastore* was successfully updated.

**System action:** Processing continues.

**User response:** None. Informational message only.

**CEX6035W Datastore=*datastore* was not updated because *datastore* was not found in the list of active datastores.**

**Explanation:** A command was issued to update *datastore* but it was not found in the in-memory list of datastores.

**System action:** Processing continues.

**User response:** Correct the datastore name and try again.

## FUN-prefixed messages

This topic describes messages with the **FUNnnnnx** message identifier format.

**FUN1003I Processing *event* at *time***

**Explanation:** These are startup and shutdown information messages. *event* identifies 'started' or 'ended'.

**System action:** The job continues.

**User response:** None. Informational message only.

**FUN1004I Message file DD *ddname reason* - records *action***

**Explanation:** The output message file is in error or cannot be found and the records have been suppressed or lost. *ddname* identifies the output file, *reason* identifies 'not found' or 'in error' and *action* the action taken with the records.

**System action:** The job continues.

**User response:** None. Informational message only.

**FUN1005W Message data with destination *dest* has been lost**

**Explanation:** Output message data for the specified destination *dest* has been lost.

**System action:** The job continues.

**User response:** None required. This is a warning message only.

**FUN1006S Internal logic error *rsn*. INFO=*info/info2***

**Explanation:** The functional support environment has detected an internal error and is about to quiesce its functionality.

**System action:** The functional support environment quiesces.

**User response:** Gather the following documentary evidence at the time the problem occurs: copy of the job output, the PRINT data set and a portion of the JES syslog. Contact IBM Software Support.

**FUN1007E Processing error *rsn*. INFO=*info/info2***

**Explanation:** The functional support environment has detected an internal processing error and is about to quiesce its functionality.

**System action:** The functional support environment quiesces.

**User response:** Gather the following documentary evidence at the time the problem occurs: copy of the job output, the PRINT data set and a portion of the JES syslog. Contact IBM Software Support.

**FUN1008E JCL parameter *parm* is missing or invalid**

**Explanation:** There is a missing parameter for this job.

**System action:** The job will not run.

**User response:** Refer to the instructions for running the job and then resubmit the job.

**FUN1009E Unable to *function* module *name*. INFO=*info***

**Explanation:** The functional support environment could not find the specified module. *function* identifies the function being performed and *name* the module name. *info* identifies the reason code.

**System action:** The functional support environment quiesces.

**User response:** Gather the following documentary evidence at the time the problem occurs: copy of the job output, the PRINT data set and a portion of the JES syslog. Contact IBM Software Support.

**FUN1010E Insufficient storage available. INFO=*info/info2***

**Explanation:** The functional support environment was unable to acquire sufficient storage. *info* identifies the amount of storage requested.

**System action:** The functional support environment quiesces.

**User response:** Increase the region size parameter for the job and restart.

**FUN1011I** **Max trace option cards (*maxl*) exceeded - continuing with tracing suppressed**

**Explanation:** Number of trace cards exceeded *maxl.* Tracing suppressed.

**System action:** Processing continues.

**User response:** None. Informational message only.

**FUN1012I** **Trace options invalid - continuing with tracing suppressed. INFO=*rsn***

**Explanation:** Invalid trace options.

**System action:** Processing continues.

**User response:** None. Informational message only.

**FUN1013I** **Trace options ignored for *id* as object not found**

**Explanation:** Trace options ignored.

**System action:** Processing continues.

**User response:** None. Informational message only.

**FUN1014I** **Trace buffer dump *poslit* at *stck***

**Explanation:** Trace buffer dump.

**System action:** Processing continues.

**User response:** None. Informational message only.

**FUN1015I** **Trace destination not defined - continuing with tracing suppressed**

**Explanation:** Trace destination not defined.

**System action:** Processing continues.

**User response:** None. Informational message only.

**FUN1016I** **Trace service initialization failed - continuing with tracing suppressed**

**Explanation:** Trace service initialization failed.

**System action:** Processing continues.

**User response:** None. Informational message only.

**FUN2051E** **Error on VSAM file *dsname*, VSAM request is *func*, return code = *rc*, reason code = *rsn***

**Explanation:** The functional support environment has detected an error with the specified VSAM file *dsname. func* identifies the VSAM macro that failed, for example OPEN, GET, PUT, ERASE. *rc* and *rsn* identify the VSAM Macro return and reason codes.

**System action:** The functional support environment stops processing.

**User response:** Ensure the file is a valid VSAM file and resubmit the job. If it fails repeatedly, refer to the DFSMS/MVS Macro Instructions for Data Sets (SC26-4913) for an explanation of the VSAM Macro Return and Reason Codes. If you cannot correct the problem, contact IBM Software Support.

**FUN2052E** **Unable to open VSAM file *ddname* - DD statement missing**

**Explanation:** The functional support environment cannot open a VSAM file because the DD statement is missing. *ddname* identifies the VSAM file.

**System action:** The functional support environment stops processing.

**User response:** Ensure the file is defined and resubmit the job.

**FUN2053E** ***dsname* is not a VSAM KSDS**

**Explanation:** The functional support environment has identified that the specified file *dsname* is not a VSAM file.

**System action:** The functional support environment stops processing.

**User response:** Ensure the file is defined correctly as a VSAM file and resubmit the job.

**FUN2054E** **Unable to locate Catalog Search Interface routine *csiname***

**Explanation:** The functional support environment cannot locate the specified Catalog Search Interface *csiname.*

**System action:** The functional support environment stops processing.

**User response:** Investigate why the named module is not available. If necessary add the library containing the module to the STEPLIB and resubmit the job.

**FUN2055E** **VSAM file *dsname* data component has shareoptions *dataopt1 dataopt2*, index has *idxopt1 idxopt2***

**Explanation:** The functional support environment has identified that the shareoptions for the data component and the index component are not the same. *dsname* identifies the VSAM file, *dataopt1* and *dataopt2* the shareoptions for the data component and *idxopt1* and *idxopt2* the shareoptions for the index component.

**System action:** The functional support environment stops processing.

**User response:** The shareoptions must be the same for the data and index components. Ensure the file is defined correctly and resubmit the job.

**FUN2056E VSAM file *dsname* defined with shareoptions *opt1 opt2* instead of (4 3)**

**Explanation:** The functional support environment has identified that the VSAM file has been specified with incorrect shareoptions. *dsname* identifies the VSAM file and *opt1* and *opt2* the shareoptions. The shareoptions must be defined (4 3 ).

**System action:** The functional support environment stops processing.

**User response:** Ensure the file is defined with the correct shareoptions and resubmit the job.

**FUN2060E VSAM file *dsname* defined with incorrect key *length offset* instead of (64 0)**

**Explanation:** The functional support environment has identified an incorrect key for the VSAM file. *length* and *offset* identify the invalid key values.

**System action:** The functional support environment stops processing.

**User response:** Ensure the file is defined with a key of (64 0) and resubmit the job.

**FUN2100F Failed to format a message. The Message ID is *messageid*. Reason: *reason***

**Explanation:** The functional support environment failed to format a message.

**System action:** Processing continues.

**User response:** Gather the following documentary evidence at the time the problem occurs: copy of the job output, the PRINT data set and a portion of the JES syslog. Contact IBM Software Support.

**FUN2210F Syntax error at end of input: string or comment terminator missing**

**Explanation:** The Parameter Parser has reached the end of the input parameters while trying to locate the end of a string or a comment.

**System action:** The Parameter Parser stops processing.

**User response:** Add the missing delimiter and resubmit the job.

**FUN2211I Syntax error at line *record*: the input *command* is not a known command**

**Explanation:** The Parameter Parser has detected an unexpected command. *command* identifies the command and *record* identifies the record number containing the unexpected command.

**System action:** The Parameter Parser continues.

**User response:** None. Informational message only.

**FUN2212F Syntax error at end of input: record continuation on last record**

**Explanation:** The Parameter Parser encountered a continuation character on the last record of the input parameters.

**System action:** The Parameter Parser stops processing.

**User response:** Remove the incorrect continuation and resubmit the job.

**FUN2213I Syntax error at line *record*: input does not match known keywords or positional parameters. Input string:'*input*'**

**Explanation:** The Parameter Parser has detected an unexpected parameter. *input* identifies the unknown parameter or keyword, and *record* identifies the record number where the error was found.

**System action:** The Parameter Parser stops processing.

**User response:** Edit the input and resubmit the job.

**FUN2214F Syntax error: the command *command* is required**

**Explanation:** The Parameter Parser has detected a missing mandatory command. *command* identifies the input containing the missing command.

**System action:** The Parameter Parser stops processing.

**User response:** Add the mandatory command and resubmit the job.

**FUN2215F Syntax error: required keyword/parameter is not present. Option:*option*.**

**Explanation:** The Parameter Parser has detected a missing mandatory option. *option* identifies the input containing the missing option.

**System action:** The Parameter Parser stops processing.

**User response:** Add the mandatory option and resubmit the job.

**FUN2216F Syntax error at line *record*: the command *command* may not be repeated**

**Explanation:** The Parameter Parser has detected multiple commands *command* where only a single command is allowed. *record* identifies the record number where the second or subsequent command was found.

**System action:** The Parameter Parser stops processing.

**User response:** Remove the multiple entries and resubmit the job.

**FUN2217F Syntax error at line *record*: the keyword/parameter *option* may not be repeated**

**Explanation:** The Parameter Parser has detected multiple keywords or parameters *option* where only a single keyword or parameter is allowed. *record* identifies the record number where the second or subsequent keyword or parameter was found

**System action:** The Parameter Parser stops processing.

**User response:** Remove the multiple entries and resubmit the job.

**FUN2218F Syntax error at line *record*: invalid length for *option*. Value=*input*.**

**Explanation:** The Parameter Parser has detected input values that do not match expected length parameters. *option* identifies the option where the invalid length was found, *input* identifies the input string containing the invalid value, and *record* identifies the record number where the error was found. *length* identifies the length specified, and *min* and *max* respectively identify the minimum and maximum allowable lengths of the string.

**System action:** The Parameter Parser stops processing.

**User response:** Correct the value to conform to the expected length and resubmit the job.

**FUN2219F Syntax error at line *record*: Invalid datatype for *option*. Value=*input*.**

**Explanation:** The Parameter Parser has detected an input parameter *option* with a specified value that has an invalid data type.

**System action:** The Parameter Parser stops processing.

**User response:** Correct the value to conform to the expected data type and resubmit the job.

**FUN2220F Syntax error at line *record*: input value does not match the allowed syntax for *option*. Input:*input*.**

**Explanation:** The Parameter Parser has detected parameter *option* that has invalid syntax. *input* indicates the string in error and *record* identifies the record where the error was detected.

**System action:** The Parameter Parser stops processing.

**User response:** Edit the input and resubmit the job.

**FUN2221F Syntax error at line *record*: invalid value for *option*. Value=*input*.**

**Explanation:** The Parameter Parser has detected parameter *option* that has been specified with an invalid value. *input* indicates the string containing the invalid value and *record* indicates the record where the error was detected.

**System action:** The Parameter Parser stops processing.

**User response:** Correct the parameter to specify a valid value and resubmit the job.

**FUN2222F Syntax error at line *record*: number of values for parameter *option* exceed the list limit**

**Explanation:** The Parameter Parser has detected a larger number of items in a list than expected. *option* identifies the input string containing the items, and *record* identifies the record where the error was detected.

**System action:** The Parameter Parser stops processing.

**User response:** Edit the list of items and resubmit the job.

**FUN2223F Syntax error at line *record*: invalid value for *option***

**Explanation:** The Parameter Parser has detected an invalid parameter. *option* identifies the input string containing the error, and *record* identifies the record where the error was detected.

**System action:** The Parameter Parser stops processing.

**User response:** Edit the input and resubmit the job.

**FUN2224F Syntax error at line *record*: unmatched parenthesis. Input:*input*.**

**Explanation:** The Parameter Parser has detected a missing parenthesis. *input* identifies the input string containing the missing parenthesis, and *record* identifies the record where the error was detected.

**System action:** The Parameter Parser stops processing.

**User response:** Add the missing parenthesis and resubmit the job.

**FUN2225F Syntax error at line *record*: unexpected data command expected. Input:*input*.**

**Explanation:** The Parameter Parser has detected a missing command. *input* identifies the input string containing the missing command, and *record* identifies the record where the error was detected.

**System action:** The Parameter Parser stops processing.

**User response:** Add the missing command and resubmit the job.

**FUN2226F** **Syntax error at line** *record***: command** *command* **is out of sequence**

**Explanation:** The Parameter Parser has detected an out of sequence command. *command* identifies the input string containing the sequence error, and *record* identifies the record where the error was detected.

**System action:** The Parameter Parser stops processing.

**User response:** Put the command in sequence and resubmit the job.

**FUN2227F** **Syntax error at line** *record***: invalid delimiter for** *option***. Value=***val***.**

**Explanation:** The Parameter Parser has detected an invalid delimiter. *option* identifies the input string containing the error, and *record* identifies the record where the error was detected.

**System action:** The Parameter Parser stops processing.

**User response:** Put the command in sequence and resubmit the job.

**FUN2300F** **Load of resource class** *classname* **failed, SAF return code=***safrc/racfrc/racfrs*

**Explanation:** A request to reload the security rules with name *classname* was rejected by the security system. *safrc* identifies the SAF return code. *racfrc/racfrsn* is the return/reason code from RACF or the installation security exit.

**System action:** Processing continues with the existing security setting.

**User response:** Refer to your installation security product documentation to check the return and reason codes.

**FUN2301F** **RACROUTE** *function* **failed, SAF return code=***safrc/racfrc/racfrs*

**Explanation:** A security SAF request *function* was rejected by the security system. *safrc* identifies the SAF return code. *racfrc* / *racfrsn* is the return/reason code from RACF or the installation security exit.

**System action:** The requested security function fails.

**User response:** Refer to your installation security product documentation to check the return and reason codes.

**FUN2500I** **Journal Manager initialized with** *numact* **active and** *numover* **overflow journals**

**Explanation:** Journal Manager initialization has completed and the Journal manager is ready to start recording events. *numact* identifies the number of Active data sets and *numover* the number of Overflow data sets.

**System action:** The Journal Manager continues.

**User response:** None. Informational message only.

**FUN2501I** **Journal full option is** *option***, archiving is** *option*

**Explanation:** Journal Manager initialization has completed and the Journal Manager is ready to start recording events. The Journal full option and Archiving options are identified.

**System action:** The Journal Manager continues.

**User response:** None. Informational message only.

**FUN2502I** **Archive JCL Skeleton PDS is** *pds***, member is** *member*

**Explanation:** The Journal Manager detected Archiving is active and identifies the skeleton JCL that will be used. *pds* identifies the data set name and *member* the member name containing the skeleton.

**System action:** The Archive utility continues.

**User response:** None. Informational message only.

**FUN2503E** *dsn member* **is in use by another address space**

**Explanation:** The same Definitions data set and members are being used by two different systems. *dsn* identifies the Definitions data set name and *member* the member name.

**System action:** The Journal Manager cannot run.

**User response:** Ensure the System Definition name matches the HWS ID parameter in the IMS Connect configuration. Ensure there is only one IMS Connect system with that name and then resubmit the job.

**FUN2504E** **Journal Manager shut down due to errors**

**Explanation:** The Journal Manager has shut down with errors.

**System action:** The Journal Manager stops processing.

**User response:** Previous error messages explain the reason for the error.

**FUN2505E Active Journal prefix *dsn* is in use by a Journal Manager in another address space**

**Explanation:** Two systems are using the same data set name prefix for the Active Journal. To prevent overwriting of data, the Journal Manager does not allow this to occur and stops processing. *dsn* identifies the data set name.

**System action:** The Journal Manager stops processing.

**User response:** Change the Active Journal data set name definition using the ISPF panels and then resubmit the job.

**FUN2506E Journal Manager was unable to determine journal status due to errors processing checkpoint record**

**Explanation:** The Journal Manager was unable to determine the status of the active journal due to a problem reading the internal checkpoint record in the repository.

**System action:** The Journal Manager stops processing.

**User response:**

Use the **RESET** command from the Active Journal Data Set panel to correct the problem. Refer to “Resizing or redefining active journals” on page 70. Note that some records might be lost.

If the problem persists contact IBM Software Support.

**FUN2507E Journal Manager shutting down - no Active journals available and journal full option is 'END'**

**Explanation:** The Journal Manager is unable to write records to the Active Journals. You have specified END as the Journal full option so the Journal manager stops processing.

**System action:** The Journal Manager stops processing.

**User response:** Check the status of the Active Journals and why they are not available. You can also select to change the Journal full option using the ISPF panels and then resubmit the job.

**FUN2508E Journal Manager detected an error in the Active journal definition - *reason***

**Explanation:** The Journal Manager has detected an error in the Active Journal definition. *reason* identifies the type of error.

**System action:** The Journal Manager stops processing.

**User response:** If there is an error with the record length then you need to delete and redefine the Definitions data set. Other problems can be addressed using the ISPF panels. Make the necessary changes and then resubmit the job.

**FUN2514I Journal Archive complete, nn active journal(s) processed**

**Explanation:** The Archive Manager has completed and archived the the specified number of Active Journal data sets.

**System action:** The Archive Manager continues.

**User response:** None. Informational message only.

**FUN2515I Journal Manager now writing event records to active journal *dsn***

**Explanation:** The Journal Manager has started to use the Active Journal data set. *dsn* identifies the data set name.

**System action:** The Journal Manager continues.

**User response:** None. Informational message only.

**FUN2516W Active journal *dsn* is unusable**

**Explanation:** The Journal Manager is unable to use the Active Journal data set *dsn*.

**System action:** The Journal Manager continues.

**User response:** Previous error messages explain the reason for the error.

**FUN2517I Writing Archive *dsn***

**Explanation:** The Archive utility is writing records to the Archive Journal data set. *dsn* identifies the data set name.

**System action:** Processing continues.

**User response:** None. Informational message only.

**FUN2518I Reading Journal *dsn***

**Explanation:** The Archive utility is reading data from the Journal data set. *dsn* identifies the data set name.

**System action:** Processing continues.

**User response:** None. Informational message only.

**FUN2519I Closing Archive *dsn***

**Explanation:** The Archive utility is closing the Archive Journal data set. *dsn* identifies the data set name.

**System action:** Processing continues.

**User response:** None. Informational message only.

**FUN2520E FIRST and/or JFIRST not allowed for tape DSN skeleton**

**Explanation:** Unsupported symbols (&FIRST or &JFIRST) detected for tape archiving using a skeleton from version 1 release 1 of IMS Connect Extensions.

**System action:** The archiving job fails.

**User response:** Use &DATE and &TIME or update your JCL skeletons to the latest supported format.

**FUN2521F Journal archive failed, refer to previous error messages**

**Explanation:** A failure occurred. Refer to previous messages for the error.

**System action:** Utility processing ends.

**User response:** Address the cause of the failure and retry.

**FUN2522W No journals archived**

**Explanation:** The Archive utility ran successfully but did not find any journals to archive. This can happen if a second archive job is submitted before the previous archive job is able to complete. The Archive Manager utility will archive any active journals that are full at the time the utility starts running. If the first job finds two full journals to archive and then completes, the second job might start up and find no journals to archive.

If archive jobs overlap in this way it might indicate that the jobs are slow to initiate or execute.

**System action:** Processing continues.

**User response:** If this message appears regularly, consider increasing the number of initiators or assigning a dedicated initiator for archive jobs.

**FUN2530F Allocation failed for** ***dsn***

**Explanation:** Allocation for a data set has failed. *dsn* identifies the data set name.

**System action:** Processing stops.

**User response:** Previous error messages explain the reason for the error and help diagnosis. If the message relates to an active journal data set, refer to "Resizing or redefining active journals" on page 70. If the message relates to a non-SMS GDG archive journal data set, ensure that the pattern DSCB name is the same name as the GDG base name.

**FUN2531F Open failed for** ***dsn***

**Explanation:** Opening of a data set has failed. *dsn* identifies the data set name.

**System action:** Processing stops.

**User response:** Previous error messages explain the reason for the error and help diagnosis.

**FUN2532F Open failed for DDname** ***ddn***

**Explanation:** Opening of ddname has failed. *ddn* identifies the ddname.

**System action:** Processing stops.

**User response:** Previous error messages explain the reason for the error and help diagnosis.

**FUN2533W Deallocation failed for** ***dsn***

**Explanation:** An attempt to free a data set has failed. *dsn* identifies the data set name.

**System action:** Processing continues and the data set remains allocated.

**User response:** Previous error messages explain the reason for the error and help diagnosis.

**FUN2534I Added Active Journal DSN** ***dsn***

**Explanation:** The number of Active Journal data sets has been changed in the Active Journal definition. As a result the Journal Manager has added a data set to the Active Journals. *dsn* identifies the data set name.

**System action:** The Journal Manager continues.

**User response:** None. Informational message only.

**FUN2535I Added Overflow DSN** ***dsn***

**Explanation:** The number of Overflow data sets has been changed in the Active Journal definition. As a result the Journal Manager has added an overflow data set. *dsn* identifies the data set name.

**System action:** The Journal Manager continues.

**User response:** None. Informational message only.

**FUN2536I Removed Active Journal DSN** ***dsn***

**Explanation:** The number of Active Journal data sets has been changed in the Active Journal definition. As a result the Journal Manager has removed a data set from the Active Journals. *dsn* identifies the data set name.

**System action:** The Journal Manager continues.

**User response:** None. Informational message only.

**FUN2537I Removed Overflow DSN** ***dsn***

**Explanation:** The number of Overflow data sets has been changed in the Active Journal definition. As a result the Journal Manager has removed an overflow data set. *dsn* identifies the data set name.

**System action:** The Journal Manager continues.

**User response:** None. Informational message only.

**FUN2538I New Journal dataset *dsn* allocated**

**Explanation:** The Journal Manager required a journal data set and one was not found so a new data set was created.

**System action:** The Journal Manager continues.

**User response:** None. Informational message only.

**FUN2539I The Journal manager cannot access the checkpoint record**

| **Explanation:** The archive manager is unable to access the checkpoint record in the repository. The checkpoint record is being held by another job.

**System action:** This message is issued every minute until the checkpoint record is freed.

| **User response:** Check if an archive job is already active. Ensure that the archive job is not being delayed in some way such as waiting for a tape mount.

**FUN2542W No Active Journal data sets available**

**Explanation:** The Active Journal data sets are unavailable - they may be full.

**System action:** The Journal Manager continues. If an Overflow data set has been specified it will be used. If no Overflow data set was specified, the Journal full option will be used.

**User response:** Refer to "Resizing or redefining active journals" on page 70.

**FUN2543W No Overflow data sets available**

**Explanation:** The Overflow data sets are unavailable - they may be full.

**System action:** The Journal Manager continues. The Journal full option will be used.

**User response:** Refer to "Resizing or redefining active journals" on page 70.

**FUN2560I Archive job *jobname*, *jobnumber* submitted using *rule_level* rules**

**Explanation:** The Journal Manager has submitted a job to archive a full active journal data set. *jobname* and *jobnumber* identify the job. The job can use one of the following for *rule_level*:

**V110** Archive skeleton uses version 1 release 1 rules. That is, the archive skeleton does not contain an ARCOUT DD with a DSN.

**V120** Archive skeleton uses version 1 release 2 rules. That is, the archive skeleton contains an ARCOUT DD with a DSN or DSN template.

**System action:** The Archive Manager continues.

**User response:** None. Informational message only.

**FUN2563F Archive job submission failed, archive JCL member *member* not found**

**Explanation:** The job to archive the active journal data set has failed. The skeleton member *member* name was not found.

**System action:** The Archive Manager failed.

**User response:** Verify the Archive JCL member data set and member name.

**FUN2564F Journal records are being discarded due to unavailable Active journals**

**Explanation:** The Active Journals are not available and the Journal full option specifies Discard. Records will be discarded until the Active Journals become available.

**System action:** The Journal Manager continues.

**User response:** None. Informational message only.

**FUN2565F Journal records are no longer being discarded**

**Explanation:** The Active Journals have become available and the records are no longer being discarded.

**System action:** The Journal Manager continues.

**User response:** None. Informational message only.

**FUN2566W Job card detected in both Active Journal Data Set definition and Archive JCL skeleton. Definition JCL ignored. Job names=*jobname1*, *jobname2***

**Explanation:** The Journal Archive Manager detected a valid job card from the Active Journal Data Set panel in the ISPF dialog (panel ID CEXJDSEP) and in the Archive JCL Skeleton member.

**System action:** The archive manager job is submitted using the job card from the Archive JCL Skeleton. All of the JCL cards from the Active Journal Data Set panel are ignored.

**User response:** Correct the ambiguity between the two job cards.

**FUN2567W No job card detected in either the Active Journal Data Set definition or the Archive JCL skeleton. Job names=*jobname1*, *jobname2***

**Explanation:** The Journal Archive Manager did not find a valid job card in the Active Journal Data Set panel in the ISPF dialog (panel ID CEXJDSEP) or in the Archive JCL Skeleton member.

**System action:** Processing continues. The archive manager job is submitted using the non-blank definition JCL cards and the Archive JCL Skeleton, but the submitted job will likely fail.

**User response:** Use one of the available methods to provide a valid job card.

---

**FUN2568E** **Invalid job ID returned from Internal Reader submission. Job ID=***jobID*

**Explanation:** The Internal Reader returned something other than the three possible job types after the submission of the archive manager JCL. The three allowable job types are JOB, STC, and TSU. This is most likely due to JES2/JES3 not recognizing the submitted JCL as a job.

**System action:** Processing continues.

**User response:** Determine the cause of the invalid job type and correct the problem.

---

**FUN2570I** *operation* **ARC records prior to** *hh***:***mm***:***ss* **on day** *day* **in month** *month* **of year** *year*

**Explanation:** Shows the date and time threshold used to determine if an archive data set is a candidate for deletion.

**System action:** The utility continues.

**User response:** None. Informational message only.

---

**FUN2571I** **DSN=***data_set_name***, first=** *yyyy-mm-dd hh:mm:ss.sss***, last=***yyyy-mm-dd hh:mm:ss.sss***,** *dsn_full_status***,** *dsn_catalog_status*

**Explanation:** Provides information about an archive data set, which was selected for processing.

*data_set_name*
The name of the archive data set.

**first** *yyyy-mm-dd hh:mm:ss.sss*
The timestamp for the first (oldest) record in the data set.

**last** *yyyy-mm-dd hh:mm:ss.sss*
The timestamp for the last (newest) record in the data set.

*dsn_full_status*
One of the following:

**F** Full. Archive data set has reached threshold. No further data will be added.

**R** Ready. Data can still be added.

**U** Unknown or transient state.

*dsn_catalog_status*
One of the following:

**Cataloged**
The data set is cataloged.

**Deleted**
The data set is not cataloged.

**Catlg GDG**
The data set is cataloged GDG.

**System action:** The utility continues.

**User response:** None. Informational message only.

---

**FUN2572I** *data_set_name***,** *literal_1***,** *literal_2*

**Explanation:** Information about the operations that the archive cleanup utility performed.

*data_set_name*
The name of the data set.

*literal_1* Specifies what happened to the archive entry in the repository.

*literal_2* Specifies what happed to the archive data set.

**System action:** The utility continues.

**User response:** None. Informational message only.

---

**FUN2573I** **Control card:** *content*

**Explanation:** The card image of the input control card read from the ARCCNTL DD.

**System action:** The utility continues.

**User response:** None. Informational message only.

---

**FUN2574I** **Archive cleanup retention period is 0, retention period defaulted to 7 days.**

**Explanation:** IMS Connect Extensions found the archive cleanup retention period was invalid. This is possible if you have applied APAR PK17019 or later, which adds this field to your existing definitions.

**System action:** Processing continues.

**User response:** Specify an archive cleanup retention period on the archive data set panel under system definitions.

---

**FUN2575E** *Variable*

**Explanation:** An archive cleanup control card contains an invalid parameter.

**System action:** Utility processing ends.

**User response:** Correct the invalid control card.

---

**FUN2580I** **Journal refresh has changed** *literal_1 literal_2*

**Explanation:** Highlights changes detected during the processing of the JOURNAL REFRESH command.

**System action:** The Journal Manager continues.

**User response:** None. Informational message only.

**FUN2581F** **DSNAME string must have a valid terminator – only a comma or blank are acceptable**

**Explanation:** The Journal Manager has found an invalid character used to terminate the data set name string in the skeleton archive JCL member.

**System action:** The Journal Manager continues.

**User response:** Ensure that either a comma or a blank character is used to terminate the data set name string.

**FUN2582F** **Archive JCL skeleton contains blank card**

**Explanation:** The Journal Manager found a blank card in the skeleton archive JCL member.

**System action:** The Journal Manager continues.

**User response:** Remove the blank card from the skeleton archive JCL member.

**FUN2583I** *jcl*

**Explanation:** Echoes JCL statements for an archive job.

**System action:** Processing continues.

**User response:** None. Informational message only.

**FUN2594W** *nn* **blocks of journal data skipped**

**Explanation:** Some data records have not been archived and have been discarded (the Journal full option is set to Discard). *nn* identifies the number of blocks of data discarded.

**System action:** The Journal Manager continues.

**User response:** Refer to “Resizing or redefining active journals” on page 70.

**FUN2595I** **Processing started for HWSID=***hwsid*

**Explanation:** CLEAN command processing has started for the IMS Connect system named in the message.

**System action:** Processing continues.

**User response:** None. Informational message only.

**FUN2596I** **Processing ended for HWSID=***hwsid* **Return Code:** *rc* **Reason Code:** *rsn*

**Explanation:** CLEAN command processing has ended for the IMS Connect system named in the message.

**User response:** None. Informational message only.

**FUN2597I** **Candidate selection ended, highest RC received=***rc*

**Explanation:** Processing has ended for all of the IMS Connect systems listed in the NAME specification in the CLEAN command.

**User response:** None. Informational message only.

**FUN2598I** *message*

**Explanation:** This message is associated with dynamic allocation failures.

**System action:** Processing continues.

**User response:** Previous error messages explain the reason for the error and help diagnosis. If the message relates to an active journal data set, refer to “Resizing or redefining active journals” on page 70.

**FUN2599W** *module* **- RC:** *return_code***, RSN:** *reason_code***, INFO:** *error_description*

**Explanation:** The archive cleanup utility has encountered a problem. The utility provides the following diagnostic information:

*module* The module that encountered the problem.

*return_code*
Error return code.

*reason_code*
Error reason code.

*error_description*
Optional descriptive information.

**System action:** Processing continues.

**User response:** Previous error messages explain the reason for the error and help diagnosis. Refer to the appropriate IBM manual.

**FUN2600E** **The Journal Manager was unable to determine the archive JCL skeleton - no data set name specified**

**Explanation:** The Journal Manager cannot determine the data set name for the JCL skeleton.

**System action:** The Journal Manager stops.

**User response:** Check that the Active Journal definition specifies the skeleton JCL data set and member name and then resubmit the job.

**FUN2601E** **The Journal Manager was unable to determine the archive JCL skeleton - no member name specified**

**Explanation:** The Journal Manager cannot determine the member name for the JCL skeleton.

**System action:** The Journal Manager stops.

**User response:** Check that the Active Journal definition specifies the skeleton JCL data set and member name and then resubmit the job.

**FUN2602E The Journal Manager was unable to determine the archive settings**

**Explanation:** The Journal Manager cannot determine any settings and could not run.

**System action:** The Journal Manager fails.

**User response:** Verify the Archive Journal definition exists and then resubmit the job.

**FUN2603E The Journal Manager was unable to determine the active journal settings**

**Explanation:** The Journal Manager cannot determine any settings for the Active Journal and could not run.

**System action:** The Journal Manager fails.

**User response:** Verify the Active Journal definition exists and then resubmit the job.

**FUN2604I Using PET support for TDB=***tdb***, task=***task***, type=***type*

**Explanation:** The TDB service has been enabled to support zIIP offload if it has been requested.

**System action:** Processing continues.

**User response:** None. Informational message only.

**FUN2701F Open failed for ddname** *ddn*

**Explanation:** Open failed for utility data set: *ddn*.

**System action:** Utility pauses.

**User response:** Add the missing ddname to the utility JCL and resubmit the job.

**FUN2702E FUNTDB ERROR** *tdb-name***, T=***failure-type***, R=***return-code***, H=***hndl-address***, P=***pflag-bits*

**Explanation:** *failure-type* can be any of the following:

APET: Allocate PET

DPET: Deallocate PET

NHDL: PET not in handle

PWTR: Pause in PUT WAIT

PRDR: Pause in GET WAIT

RWTR: Release of WRITER

RRDR: Release of READER

SERR: HNDL state error

**System action:** Processing continues.

**User response:** Contact IBM Software Support. Provide the system dump and the text of this message.

**FUN2708I Jobname=** *jobname***, Comp Code=** *task_ccode***, EPSW=***error_psw*

**Explanation:** A task has ended abnormally. The task completion code and error PSW are shown in the message. An accompanying message FUN2709I shows the contents of the general registers at the time of the abend.

**System action:** Processing continues. If possible a System Dump is taken.

**User response:** Contact IBM Software Support. Provide the system dump and the text of this message and the accompanying message FUN2709I.

**FUN2709I R***m* **- R***n xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx*

**Explanation:** This message follows message FUN2708I and displays the contents of the general registers at the time of the task abend.

**System action:** Processing continues.

**User response:** None. Informational message only.

**FUN2710I Sdump failed, RC=***rc*

**Explanation:** An attempt to take a dump of a failure was not successful. *rc* is the return code from the SDUMP request.

**System action:** Error processing continues depending upon the error condition.

**User response:** Correct the issue that caused the SDUMP request to fail.

**FUN2711I D37 abend, offset=***offset***,** *xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx*

**Explanation:** A series of these messages is issued during ESTAE processing in the Journal Manager if a D37 abend occurs. Each line contains an offset into the diagnostic information associated with the abend.

**System action:** Processing continues after the diagnostic messages are issued.

**User response:** Contact IBM Software Support. Provide the system dump and the text of this message.

**FUN2712W Journal data set** *dsn* **is corrupted**

**Explanation:** The journal data set is corrupted. Records may have been lost. This could occur if z/OS is unable to close the data set automatically, for example if the FORCE command was used to end a job or step or in the event of a hardware failure.

**System action:** The journal data set is closed and marked as ready so that it can be used by the Journal Manager. Archive processing continues.

**User response:** No action is required. This message is issued to report the possibility of some records being omitted from the journal at the time of the failure.

**FUN2713W Storage could not be obtained for the Journal Manager trace RC=*rc*, RSN=*rsn***

**Explanation:** The request to obtain storage for the Journal Manager trace facility failed.

**System action:** Processing continues with the trace being inactive.

**User response:** Increase the IMS Connect region size. If that does not resolve the issue: Contact IBM Software Support.

**FUN2714W Journal data set *dsn* was not closed by the Journal Manager, potential record loss**

**Explanation:** This condition is usually the result of a system reset. Buffered event records may have been lost.

**System action:** Processing continues.

**User response:** No action is required. This message is issued to report the possibility of some records being omitted from the journal.

**FUN2715E Journal data set *dsn* may be unusable. Error description:** *error*

**Explanation:** The journal data set is corrupted. Event records may have been lost.

**System action:** The journal data set is closed and marked as unusable. Processing continues.

**User response:** Investigate why the data set could not be closed automatically by z/OS , for example due to a hardware failure.

# Chapter 29. Gathering diagnostic information

Before you report a problem with IMS Connect Extensions to IBM Software Support, you need to gather the appropriate diagnostic information.

## Procedure

Provide the following information for all IMS Connect Extensions problems:

- A clear description of the problem and the steps that are required to re-create the problem.
- Version, release, modification number, and details of applied maintenance (such as PTF numbers) for IMS and IMS Connect Extensions.
- Extracts from the following IMS Connect Extensions server SYSOUT output file ddnames:

  **CEXDIAG**
  IMS Connect Extensions diagnostic information

  **CEXPRINT**
  IMS Connect Extensions messages

  **JESMSGLG**
  MVS, VSAM, and RACF
- A copy of the IMS Connect Extensions repository.
- A copy of the IMS Connect Extensions active journal data set.
- A dump of the IMS Connect region from the time the problem occurred.
- The effect of the problem on your workflow, including details of any workaround that you use.

Provide additional information based on the type of problem that you experienced:

**For online abends, provide the following information**

- A description of the task that you were doing before the abend occurred
- A screen shot of the ISPF panel that you were using when the abend occurred, or the name of the panel that you were using, with input field values.

  **Tip:** Use the ISPF command **PANELID** to display the panel identifier.

**For errors in REXX execs, provide the following information**

- To ensure that all messages are written to SYSTPRT, include the `OPTION MSGLVL=VERBOSE` statement in the exec after initializing the CEX host command environment:

```
parse arg PARM

address LINK "CEXRXENV INIT"

address CEX "OPTION MSGLVL=VERBOSE"

⋮
```

**For errors in batch processing, provide the following information**

- Print output
- Contents of any data sets that were used during the processing

**For errors in the IMS Connect Extensions Operations Console, provide the following information**

- The version numbers of the following plug-ins (use **Help** > **About IBM Explorer for z/OS** > **Installation Details**):
  - IMS Connect Extensions for z/OS Plug-in for Eclipse
  - IMS Connect Extensions for z/OS Common Plug-in for Eclipse
  - Common Services Library Framework Plug-in for Eclipse
  - Common Services Library Framework Common Plug-in for Eclipse
  - IBM z/OS Explorer Core UI Plug-in
- A screen shot showing the problem or error message.
- A description of the task that you were doing before the problem occurred.

# Part 8. Reference

These topics provide reference information for IMS Connect Extensions, including details of exits and event records and the control options that can be specified in the CEXCTLIN input data set.

**Topics:**

# Chapter 30. Client services exit

The client services exit CEXSVC01 provides services to client applications.

CEXSVC01 provides the following services to clients:

- Verify that the client user ID and password are authentic
- Change password
- Return client identity and session related information (who-am-I)
- Submit user data for logging to the journal

The client application passes a formatted message in ASCII or EBCDIC to IMS Connect with `IRM_ID=*CEXSVC*` in the IMS request message (IRM) header. The exit, CEXSVC01, processes the message and returns a response to the client. The message is not passed to IMS. The message structure is mapped by the CEXSVCCB macro, which is in the *cexpre*.USERMAC library.

**Note:** The socket is always closed after issuing a service exit request, even if the socket is persistent.

## Input message format

Here is the general format of an input message.

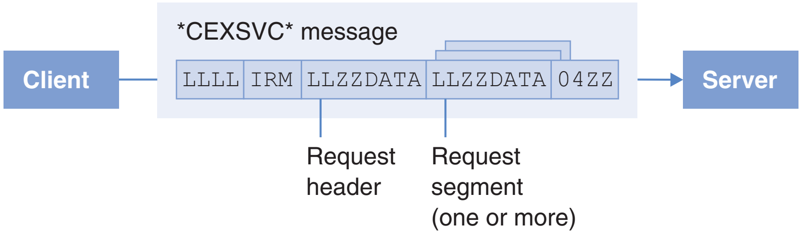


*Figure 120. Input message format*

Figure 120 shows the following:

**LLLL**
Fullword that specifies the total length of all data in the message.

**IRM**
The IMS request message (IRM). Use `IRM_ID=*CEXSVC*` in this segment. The IRM layout is defined by the HWSIMSCB macro. See the section on IMS Connect message structures in *IMS Communications and Connections*.

**Service request header**
Specifies the type of service requested and the number of requests that follow. Each request is one segment, and all segments must request the same type of service.

**Service request segment (one or more)**
Specifies the length of the segment and the segment number and additional information depending on the type of service:

**Verify user**
The user ID and password.

**Password change**
The user ID, old password, and new password.

**Who-am-I**
The field queried. The client can request one of the following per segment:

- Client ID
- The IMS Connect system identifier of the server
- IP version (IPv6 or IPv4)
- Client IP address
- Port number
- Socket type
- Event key

**Log user data**
The client ID, the RACF user ID and password, and the event record. The user ID and password fields are required because users must be authorized to use this service.

**04ZZ (optional)**
The end of message marker.

## Output message format

Here is the general format of an output message.

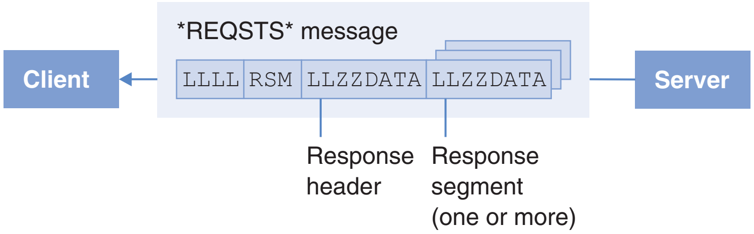


*Figure 121. Output message format*

Figure 121 shows the following:

**LLLL** Fullword that specifies the total length of all data in the message.

**RSM** The request status message (RSM). The RSM has the following form:

```
|LLZZ|*REQSTS*|return code|reason code
```

The RSM layout is defined by the HWSIMSCB macro. See the section on IMS Connect message structures in *IMS Communications and Connections*.

**Service response header**
Specifies the type of service requested and the number of responses that follow. Each response is one segment, and all segments respond to the same type of service.

**Service response segment (one or more)**
Specifies the length of the segment and the segment number and additional information depending on the type of service:

**Verify user**
The user ID and password.

**Password change**
The user ID, SAF return code, and RACF (or the installation security exit) return and reason codes.

**Who-am-I**
The field queried. The response can contain one of the following per segment:

- Client ID
- The IMS Connect system identifier of the server
- IP version (IPv6 or IPv4)
- Client IP address
- Port number
- Socket type
- Event key

**Log user data**
The event writer return and reason codes.

## Service header

This topic shows the format of the input message service header and output message service response header.

Table 11 shows the format of the input message service header.

*Table 11. Service header segment format*

| Offset | Type | Length | Field name | Description |
|---|---|---|---|---|
| 0 | SIGNED | 2 | SVC1_HDR_LEN | This segment's length. Enter 32. |
| 2 | N/A | 2 | ZZ | low values |
| 4 | CHARACTER | 4 | SVC1_HDR_ID | Control block ID. You must use: `SVC1` |
| 8 | SIGNED | 2 | SVC1_HDR_SR | The service request type:<br>**240** Verify user<br>**241** Password change<br>**242** Who-am-I<br>**243** Log user data |
| 10 | SIGNED | 2 | SVC1_HDR_NSEG | Number of segments in the message |
| 12 | N/A | 20 | | low values |

The following table shows the format of the output message service header.

*Table 12. Service response header segment format*

| Offset | Type | Length | Field name | Description |
|---|---|---|---|---|
| 0 | SIGNED | 2 | SVC1_HDR_LEN | This segment's length (always 32) |
| 2 | N/A | 2 | ZZ | low values |
| 4 | CHARACTER | 4 | SVC1_HDR_ID | Control block ID |
| 8 | SIGNED | 2 | SVC1s_HDR_SR | The service request type:<br>**240** Verify user<br>**241** Password change<br>**242** Who-am-I<br>**243** Log user data |
| 10 | SIGNED | 2 | SVC1_HDR_NSEG | Number of segments in the message |

Table 12. Service response header segment format (continued)

| Offset | Type | Length | Field name | Description |
|---|---|---|---|---|
| 12 | SIGNED | 4 | SVC1_HDR_RC | Return code, if the return code is non-zero, one or more of the requests may have failed. Examine the return code for individual messages. |
| 16 | BITSTRING | 4 | SVC1_HDR_RSN | Reason code |
| 20 | CHARACTER | 4 | SVC1_HDR_INFO | For some return codes that are not zero, provides the offset to the segment containing the error. |
| 24 | N/A | 8 | | low values |

When the user is not authorized for the service request and extended RSM support is specified for the Service Exit, the service header is followed by an extended RSM containing the message list returned by the RACROUTE request. The extended RSM is mapped by the CEXERSM macro, which is in the *cexpre*.USERMAC library.

## Verify user segment

If you set a service request type of 240 you need to enter one or more segments, each representing a request for user authentication.

The following table shows the format of such a segment.

**Note:** If the passwords you are using are encrypted, then the security preprocessing exit can be used to decrypt them.

Table 13. Verify user request format

| Offset | Type | Length | Field name | Description |
|---|---|---|---|---|
| 0 | SIGNED | 2 | SVC1_VUSI_LEN | This segment's length. Enter 24. |
| 2 | N/A | 2 | ZZ | low values |
| 4 | SIGNED | 2 | SVC1_VUSI_SEGN | Segment number |
| 6 | N/A | 2 | ZZ | low values |
| 8 | CHARACTER | 8 | SVC1_VUSI_UID | User ID. If shorter than 8 characters, pad with spaces. |
| 16 | CHARACTER | 8 | SVC1_VUSI_PW | Password. If shorter than 8 characters, pad with spaces. |

The following table shows the format of the “verify user” formatted segments that are returned for a service request type of 240.

Table 14. Verify user response format

| Offset | Type | Length | Field name | Description |
|---|---|---|---|---|
| 0 | SIGNED | 2 | SVC1_VUSO_LEN | This segment's length (always 28). |
| 2 | N/A | 2 | ZZ | low values |
| 4 | SIGNED | 2 | SVC1_VUSO_SEGN | Segment number. |
| 6 | SIGNED | 2 | SVC1_VUSO_#MSG | Number of messages in list. 1 |
| 8 | CHARACTER | 8 | SVC1_VUSO_UID | User ID |
| 16 | SIGNED | 4 | SVC1_VUSO_SAFRC | SAF return code |

Table 14. Verify user response format (continued)

<table>
<tr><th>Offset</th><th>Type</th><th>Length</th><th>Field name</th><th>Description</th></tr>
<tr><td>20</td><td>SIGNED</td><td>4</td><td>SVC1_VUSO_RACFRC</td><td>RACF return code</td></tr>
<tr><td>24</td><td>SIGNED</td><td>4</td><td>SVC1_VUSO_RACFRSN</td><td>RACF reason code</td></tr>
<tr><td colspan="5">1 When extended RSM support is requested for the Service Exit, SVC1_VUSO_#MSG contains the number of messages returned by the SAF RACROUTE request and these messages are returned in the following list. For each message, there is a length field followed by the message text. The length field contains the total length of the message including the length field. This sequence is repeated for the number of times indicated in SVC1_VUSO_#MSG.</td></tr>
<tr><td>28</td><td></td><td>0</td><td>SVC1_VUSO_MSGLIST</td><td>Message list</td></tr>
<tr><td>28</td><td>SIGNED</td><td>2</td><td>SVC1_VUSO_MSGLL</td><td>Message length</td></tr>
<tr><td>30</td><td>CHARACTER</td><td><i>MSGLL</i></td><td>SVC1_VUSO_MSG</td><td>Message text</td></tr>
</table>

## Change password segment

If you set a service request type of 241, you need to enter one or more segments, each representing a request to change a user password.

The following table shows the format of such a segment.

**Note:** If the passwords you are requesting are encrypted, then the security preprocessing exit can be used to decrypt them. See “Security preprocessing exit” on page 177 for details.

Table 15. Password change request format

| Offset | Type | Length | Field name | Description |
|---|---|---|---|---|
| 0 | SIGNED | 2 | SVC1_CPWI_LEN | This segment's length. Enter 32. |
| 2 | N/A | 2 | ZZ | low values |
| 4 | SIGNED | 2 | SVC1_CPWI_SEGN | Segment number |
| 6 | N/A | 2 | ZZ | low values |
| 8 | CHARACTER | 8 | SVC1_CPWI_UID | User ID. If shorter than 8 characters, pad with spaces. |
| 16 | CHARACTER | 8 | SVC1_CPWI_OPW | Old password. If shorter than 8 characters, pad with spaces. |
| 24 | CHARACTER | 8 | SVC1_CPWI_NPW | New password. If shorter than 8 characters, pad with spaces. |

The following table shows the format of the “password change” formatted segments that are returned for a service request type of 241.

Table 16. Password change response format

| Offset | Type | Length | Field name | Description |
|---|---|---|---|---|
| 0 | SIGNED | 2 | SVC1_CPWO_LEN | This segment's length (always 28) |
| 2 | N/A | 2 | ZZ | low values |
| 4 | SIGNED | 2 | SVC1_CPWO_SEGN | Segment number |
| 6 | SIGNED | 2 | SVC1_CPWO_#MSG | Number of messages in list 1 |
| 8 | CHARACTER | 8 | SVC1_CPWO_UID | User ID |
| 16 | SIGNED | 4 | SVC1_CPWO_SAFRC | SAF return code |

*Table 16. Password change response format (continued)*

<table>
<tr><th>Offset</th><th>Type</th><th>Length</th><th>Field name</th><th>Description</th></tr>
<tr><td>20</td><td>SIGNED</td><td>4</td><td>SVC1_CPWO_RACFRC</td><td>RACF return code</td></tr>
<tr><td>24</td><td>SIGNED</td><td>4</td><td>SVC1_CPWO_RACFRSN</td><td>RACF reason code</td></tr>
<tr><td colspan="5">1 When extended RSM support is requested for the Service Exit, SVC1_CPWO_#MSG contains the number of messages returned by the SAF RACROUTE request and these messages are returned in the following list. For each message, there is a length field followed by the message text. The length field contains the total length of the message including the length field. This sequence is repeated for the number of times indicated in SVC1_CPWO_#MSG.</td></tr>
<tr><td>28</td><td></td><td>0</td><td>SVC1_CPWO_MSGLIST</td><td>Message list</td></tr>
<tr><td>28</td><td>SIGNED</td><td>2</td><td>SVC1_CPWO_MSGLL</td><td>Message length</td></tr>
<tr><td>30</td><td>CHARACTER</td><td><i>MSGLL</i></td><td>SVC1_CPWO_MSG</td><td>Message text</td></tr>
</table>

## Who-am-I segment

If you set a service request type of 242, you need to enter one or more segments, each representing a request for information about your client.

The following table shows the format of such a segment.

*Table 17. Who-am-I request format*

| Offset | Type | Length | Field name | Description |
|---|---|---|---|---|
| 0 | SIGNED | 2 | SVC1_WHOI_LEN | This segment's length. Enter 16. |
| 2 | N/A | 2 | ZZ | low values |
| 4 | SIGNED | 2 | SVC1_WHOI_SEGN | Segment number |
| 6 | N/A | 2 | ZZ | low values |
| 8 | CHARACTER | 8 | SVC1_WHOI_NAME | Queried field name, which is one of the following values:<br>**CLIENTID** Client ID<br>**HWSID** Server ID (pad with spaces)<br>**IPVER** IP protocol version (pad with spaces)<br>**IPADDR** Client IP address (pad with spaces)<br>**PORTNO** Client outgoing port number (pad with spaces)<br>**SOCKTYPE** Socket type (pad with spaces)<br>**EVENTKEY** Event key (pad with spaces) |

The following table shows the format of the "who-am-I" formatted segments that are returned for a service request type of 242.

*Table 18. Who-am-I response format*

| Offset | Type | Length | Field name | Description |
|---|---|---|---|---|
| 0 | SIGNED | 2 | SVC1_WHOO_LEN | This segment's length (maximum length is 32 bytes) |
| 2 | N/A | 2 | ZZ | low values |

*Table 18. Who-am-I response format (continued)*

| Offset | Type | Length | Field name | Description |
|---|---|---|---|---|
| 4 | SIGNED | 2 | SVC1_WHOO_SEGN | Segment number |
| 6 | N/A | 2 | ZZ | low values |
| 8 | CHARACTER | 8 | SVC1_WHOO_NAME | The name of the field that was queried |
| **One of the following, depending on the queried field:** | | | | |
| 16 | CHARACTER | 8 | SVC1_WHOO_CLIENTID | The client ID |
| 16 | CHARACTER | 8 | SVC1_WHOO_HWSID | The server ID |
| 16 | CHARACTER | 2 | SVC1_WHOO_IPVER | The IP protocol version, where:<br>**V4** IPv4<br>**V6** IPv6 |
| 16 | BITSTRING | 4 or 16 | SVC1_WHOO_IPADDR | The IP address. 16 bytes for an IPv6 address. 4 bytes for an IPv4 addresses |
| 16 | UNSIGNED | 2 | SVC1_WHOO_PORTNO | The port from which the client is connecting |
| 16 | CHARACTER | 1 | SVC1_WHOO_SOCKTYPE | The type of socket connection:<br>**P** Persistent socket<br>**N** Non-persistent or transaction socket |
| 16 | BITSTRING | 8 | SVC1_WHOO_EVENTKEY | The event key. |

## Log user data segment

If you set a service request type of 243 you need to enter one or more segments, each representing a request to log a block of user data.

The following table shows the format of such a segment.

*Table 19. Log user data request format*

| Offset | Type | Length | Field name | Description |
|---|---|---|---|---|
| 0 | SIGNED | 2 | SVC1_LOGI_LEN | This segment's length. |
| 2 | N/A | 2 | ZZ | low values |
| 4 | SIGNED | 2 | SVC1_LOGI_SEGN | Segment number |
| 6 | N/A | 2 | ZZ | low values |
| 8 | CHARACTER | 1 | SVC1_LOGI_TYPE | User data type:<br>**C** Text<br>**X** Binary (default) |
| 9 | CHARACTER | 1 | SVC1_LOGI_CODE | User log code (optional, user-defined). |
| 10 | SIGNED | 2 | SVC1_LOGI_CCSID | User data CCSID (optional, user-defined). |
| 12 | CHARACTER | 8 | SVC1_LOGI_FMTID | User data format identifier (optional, user-defined). |
| 20 | CHARACTER | *n* | SVC1_LOGI_DATA | User data (maximum 960 bytes). |

The following table shows the format of the “log user data” formatted segments that are returned for a service request type of 243.

*Table 20. Log user data response format*

| Offset | Type | Length | Field name | Description |
|---|---|---|---|---|
| 0 | SIGNED | 2 | SVC1_LOGO_LEN | This segment's length (always 16). |

*Table 20. Log user data response format (continued)*

| Offset | Type | Length | Field name | Description |
|---|---|---|---|---|
| 2 | N/A | 2 | ZZ | low values |
| 4 | SIGNED | 2 | SVC1_LOGO_SEGN | Segment number. |
| 6 | N/A | 2 | ZZ | low values |
| 8 | SIGNED | 4 | SVC1_LOGO_RC | Log writer return code. |
| 12 | SIGNED | 4 | SVC1_LOGO_RSN | Log writer reason code. |

## Reason and return codes

The service exit provides return and reason codes in the RSM, response header, and individual service segment. The RSM and response header contain the same information. The return and reason codes in the service header indicate the overall success of a multi-request message.

A return code of 0 indicates that all requests have been processed successfully. Any other return code, indicates that one or more requests have failed. In the latter case, you need to examine each request segment to determine the presence and cause of any errors.

The service segment will contain the return code for the specific request. For example, if three requests are made and one fails, the request that fails will have a return code of 8 or 12, and the remaining requests will have a return code of 0.

*Table 21. Reason and return codes*

| Return code | Reason code | Description |
|---|---|---|
| 0 | 0 | Successfully completed. |
| 8 | 041E0008 | Unsuccessful user verification. The SAF return code and RACF return and reason codes are posted in the response segment. |
| 8 | 041E0009 | Unsuccessful password change request. The SAF return code and RACF return and reason codes are posted in the response segment. |
| 8 | 041E000B | Unsuccessful verify user request. The SAF return code and RACF return and reason codes are posted in the response segment. |
| 12 | 041E0001 | Length specified for message does not match total length of all segments. |
| 12 | 041E0002 | A message field format is invalid. Check the segment lengths, for negative values or values that are out of range. |
| 12 | 041E0003 | Header segment length field (SVC1_HDR_LEN) does not match actual length. |
| 12 | 041E0004 | Control block ID (SVC1_HDR_ID) not recognized. Use SVC1. |
| 12 | 041E0005 | The number of segments specified in the header (SVC1_HDR_NSEG) does not match the actual number of segments. |
| 12 | 041E0006 | Service request (SVC1_HDR_SR) not recognized. Use a known request value. |
| 12 | 041E0007 | The length specified for a service request segment does not match the actual segment length. |
| 12 | 041E0008 | The number of the service request segment does not match its actual order. |

Table 21. Reason and return codes (continued)

| Return code | Reason code | Description |
|---|---|---|
| 12 | 041E000A | A who-am-I request was made with an unrecognized field name (SVC1_WHOI_NAME). |
| 12 | 041E000C | Client not authorized. The SAF return code and RACF return and reason codes are posted in the response segment. |
| 12 | 041E000D | User data type error. Value must be C=text or X=binary. Default is binary. |
| 12 | 041E000E | Event logging failed. The event writer return and reason codes are posted in the response segment. |
| 16 | 041E0101 | IMS Connect Extensions not active. |
| 16 | 041E0102 | Internal error in EXIT function. |
| 16 | 041E0103 | Internal error when XMIT was entered. |
| 16 | 041E0104 | Internal error when EXER entered. |

# Chapter 31. Event records recorded by IMS Connect Extensions

Event records are collected continuously as messages are processed by IMS Connect. An event record consists of an event number and data associated with the event.

Event numbers can be X'00' - X'FF' (decimal 0 - 255) and the associated data varies depending on the event number. The event number is appended to the record prefix.

## Connect status event records

The following table lists all single event records. The Collection level determines whether the event record is written to the Journal.

*Table 22. Connect status event records*

| Code (Hex) | Code (Dec) | Event description | Collection level |
|---|---|---|---|
| 00 | 00 | Control record | 0 |
| 01 | 01 | IMS Connect region initialization | 0 |
| 02 | 02 | IMS Connect region has completed termination | 0 |
| 03 | 03 | A support task (TCB) has been created | 1 |
| 04 | 04 | A support task (TCB) is terminating | 1 |
| 05 | 05 | Begin INIT API | 4 |
| 06 | 06 | End INIT API | 4 |
| 07 | 07 | Begin Bind Socket | 4 |
| 08 | 08 | End Bind Socket | 4 |
| 09 | 09 | Listen on socket | 4 |
| 0A | 10 | Begin Accept Socket | 4 |
| 0B | 11 | End Accept Socket | 3 |
| 0E | 14 | Begin Message Exit INIT | 1 |
| 10 | 16 | IMS datastore becomes available | 0 |
| 11 | 17 | IMS datastore becomes unavailable | 0 |
| 12 | 18 | An IMS TMEMBER joins the XCF group | 0 |
| 13 | 19 | An IMS TMEMBER leaves the XCF group | 0 |
| 14 | 20 | Begin SCI Registration | 4 |
| 15 | 21 | End SCI Registration | 1 |
| 16 | 22 | Begin SCI De-registration | 4 |
| 17 | 23 | End SCI De-registration | 1 |
| 18 | 24 | Recorder trace DCB has been opened | 1 |
| 19 | 25 | Recorder trace DCB pre-close | 1 |
| 1A | 26 | User message exit return from INIT | 1 |
| 1B | 27 | User message exit return from TERM | 1 |

*Table 22. Connect status event records (continued)*

| Code (Hex) | Code (Dec) | Event description | Collection level |
|---|---|---|---|
| 1C | 28 | Begin Secure Environment Open | 1 |
| 1D | 29 | End Secure Environment Open | 1 |
| 20 | 32 | Begin Secure Environment Close | 1 |
| 21 | 33 | End Secure Environment Close | 1 |
| 22 | 34 | Begin Local Port Setup | 3 |
| 23 | 35 | End Local Port Setup | 3 |
| 24 | 36 | Begin RRS Connect | 1 |
| 25 | 37 | End RRS Connect | 1 |
| 26 | 38 | List In-doubt Context | 1 |
| 27 | 39 | Begin RRS Disconnect | 1 |
| 28 | 40 | End RRS Disconnect | 1 |
| 29 | 41 | Begin ODBM Registration | 1 |
| 2A | 42 | End ODBM Registration | 1 |
| 2B | 43 | Begin ODBM De-registration | 1 |
| 2C | 44 | End ODBM De-registration | 1 |
| 2D | 45 | Datastore Status Update | 1 |
| 2E | 46 | Return From Port Exit INIT Call | 1 |
| 2F | 47 | Return From Port Exit TERM Call | 1 |
| 30 | 48 | Begin ODBM Routing Exit INIT | 1 |
| 31 | 49 | End ODBM Routing Exit INIT | 1 |
| 32 | 50 | Begin ODBM Routing Exit TERM | 1 |
| 33 | 51 | End ODBM Routing Exit TERM | 1 |
| 34 | 52 | XML Adapter INIT Call Begin | 1 |
| 35 | 53 | XML Adapter INIT Call End | 1 |
| 36 | 54 | XML Adapter TERM Call Begin | 1 |
| 37 | 55 | XML Adapter TERM Call End | 1 |
| 38 | 56 | OM Registration | 1 |
| 39 | 57 | OM Deregistration | 1 |
| A7 | 167 | Internal Command Event | 0 |

## Workload-related event records

The following tables list all event records relating to message-related events, Open Database events, and IMS-to-IMS, and IMS-to-CICS® TCP/IP communications.

*Table 23. Workload-related event records*

| Code (Hex) | Code (Dec) | Event description | Collection level |
|---|---|---|---|
| 0C | 12 | Begin Close Socket | 4 |
| 0D | 13 | End Close Socket | 3 |
| 3C | 60 | Prepare for socket read | 2 |

Table 23. Workload-related event records (continued)

| Code (Hex) | Code (Dec) | Event description | Collection level |
|---|---|---|---|
| 3D | 61 | User message exit entered for READ, XMIT, EXER, or RXML | 2 |
| 3E | 62 | User message exit return for READ, XMIT, EXER, or RXML | 1 |
| 3F | 63 | Begin SAF security request | 3 |
| 40 | 64 | End SAF security request | 3 |
| 41 | 65 | Message sent to OTMA | 2 |
| 42 | 66 | Message received from OTMA | 2 |
| 43 | 67 | Message sent to SCI | 2 |
| 44 | 68 | Message received from SCI | 2 |
| 45 | 69 | OTMA Timeout | 1 |
| 46 | 70 | Deallocate Request | 3 |
| 47 | 71 | Session Error. This event is called when an unrecoverable error has been encountered and the session is being aborted | 1 |
| 48 | 72 | Trigger event. This is the end-of-frame event recorded by IMS Connect when a multi-event process has completed | 2 |
| 49 | 73 | Read socket | 3 |
| 4A | 74 | Write socket | 3 |
| 4B | 75 | Local Client Connect | 3 |
| 4C | 76 | Local Message Send | 3 |
| 4D | 77 | Local Message Receive | 3 |
| 4E | 78 | Local Message Send-then-Receive | 3 |
| 4F | 79 | Local Disconnect | 3 |
| 50 | 80 | Begin Create Context | 4 |
| 51 | 81 | End Create Context | 3 |
| 52 | 82 | Begin RRS Prepare | 4 |
| 53 | 83 | End RRS Prepare | 3 |
| 54 | 84 | Begin RRS Commit/Abort | 4 |
| 55 | 85 | End RRS Commit/Abort | 3 |
| 56 | 86 | Begin Secure Environment Select | 3 |
| 57 | 87 | End Secure Environment Select | 3 |
| 58 | 88 | Hold compensation queue | 1 |
| 59 | 89 | Port Edit Exit Called | 2 |
| 5A | 90 | Port Edit Exit Returned | 1 |
| 5B | 91 | DRDA Request | 2 |
| 5C | 92 | DRDA Reply | 2 |
| 5D | 93 | An Allocate PSB Command is Received | 2 |
| 5E | 94 | An Allocate PSB Command is Sent | 2 |
| 5F | 95 | A Deallocate PSB Command is Received | 2 |
| 60 | 96 | A Deallocate PSB Command is Sent | 2 |
| 61 | 97 | ODBM Routing Exit Entered | 3 |

*Table 23. Workload-related event records (continued)*

| Code (Hex) | Code (Dec) | Event description | Collection level |
|---|---|---|---|
| 62 | 98 | ODBM Routing Exit Returned | 3 |
| 63 | 99 | ODBM Security Exit Entered | 3 |
| 64 | 100 | ODBM Security Exit Returned | 3 |
| 65 | 101 | RRS Parent UR Token Creation Begins | 3 |
| 66 | 102 | RRS Parent UR Token Creation Ends | 3 |
| 69 | 105 | Message is Sent to ODBM | 3 |
| 6A | 106 | Message is Received from ODBM | 3 |
| 6B | 107 | RRS Delegate Commit Agent UR Begins | 3 |
| 6C | 108 | RRS Delegate Commit Agent UR Ends | 3 |
| 6D | 109 | XML Adapter CALL Begin | 1 |
| 6E | 110 | XML Adapter CALL End | 1 |
| 6F | 111 | XML Converter CALL Begin | 1 |
| 70 | 112 | XML Converter CALL End | 1 |
| 71 | 113 | Connected to Remote IMS Connect | 3 |
| 72 | 114 | Disconnected from Remote IMS Connect | 3 |
| 73 | 115 | Start Remote Thread for Partner ICON | 3 |
| 74 | 116 | Message Received for Remote ALTPCB | 3 |
| 75 | 117 | ALTPCB Message Sent to Remote Partner | 3 |
| 76 | 118 | ALTPCB Message Received from Remote Partner | 3 |
| 77 | 119 | Message Sent to OTMA for ALTPCB | 3 |
| 78 | 120 | MSC Message Received from MSC | 3 |
| 79 | 121 | MSC Message Sent to Partner IMS Connect | 3 |
| 7A | 122 | MSC Message Received from Partner IMS Connect | 3 |
| 7B | 123 | MSC Message Sent to MSC | 3 |
| 7C | 124 | Connection to Partner IMS Connect Time-out | 3 |
| 7D | 125 | Start of a session | 3 |
| 7E | 126 | Trigger end of a session | 3 |
| 80 | 128 | IMS Connect Extensions Automatic Trigger | 1 |
| 81 | 129 | IMS Connect Extensions Exception Event | 1 |
| 82 | 130 | Event Collection Status | 1 |
| AD | 173 | IMS Connect Extensions ACEE Cache Delete Ageing Cycle Event | 1 |
| AE | 174 | Pre-Routing Exit Audit Event | 1 |

*Table 24. Extended event records for event number 255*

| Code (Hex) | Code (Dec) | Extended event number and event description | Collection level |
|---|---|---|---|
| FF | 255 | **256** Socket connected on RMTCICS<br>**257** Socket disconnected from RMTCICS<br>**258** IMS Connect refreshed a cached RACF user ID after receiving a type 71 Event Notification Facility (ENF) notification<br>**259** IMS Connect sent a health status report to Work Load Manager (WLM)<br>**2050** Communication thread started for a RMTCICS connection<br>**2051** ISC message received from IMS<br>**2052** ISC message sent to IMS<br>**2053** ISC message received on RMTCICS socket connection<br>**2054** ISC message sent on RMTCICS socket connection<br>**2055** ISC message received on CICSPORT socket connection<br>**2056** ISC message sent on CICSPORT socket connection | 1 |

## Recorder trace records

Recorder trace data produced by IMS Connect is converted to event record format by the IMS Connect Extensions Recorder trace utilities.

*Table 25. Recorder trace event records produced by IMS Connect*

| Code (Hex) | Code (Dec) | Event description | Collection level |
|---|---|---|---|
| A0 | 160 | HWS Recorder Base Section Trace | Not applicable |
| A1 | 161 | HWS Recorder IPB Section Trace | Not applicable |
| A2 | 162 | HWS Recorder OPB Section Trace | Not applicable |

## IMS Connect trace records

IMS Connect trace event records are produced when the Tracing Level has been set for the system.

*Table 26. IMS Connect trace event records*

| Code (Hex) | Code (Dec) | Event description | Collection level |
|---|---|---|---|
| A3 | 163 | Event Collection OTMA Trace | 1 |
| A4 | 164 | Event Collection IRM Trace | 1 |
| A5 | 165 | Event Recording RSM Message Trace | 1 |

*Table 26. IMS Connect trace event records (continued)*

| Code (Hex) | Code (Dec) | Event description | Collection level |
|---|---|---|---|
| A6 | 166 | Event Recording EXIT Output Message Trace | 1 |
| A9 | 169 | RXML Trace for XML Adapter | 1 |
| AA | 170 | ODBM trace record | 1 |
| AC | 172 | IMS Connect Extensions Command and Response **1** | 1 |

**1** Event record 172 (X'AC') is a special type of event record that is used to represent a set of events resulting from IMS Connect Extensions command being issued. A record is produced for each significant action that is initiated by this command or is issued by another process in response to or on behalf of this command.

For example, this event record is recorded when the ROUTE command for DRAIN or RESUME is issued. It includes the ROUTE command as well as details of actions taken on behalf of the ROUTE command such as AUTORESUME processing. If the ROUTE DRAIN or ROUTE RESTORE command targets a routing list, one “request” record and one “response” record is created for the routing list, as well as one “both” record for each datastore in the routing list. The response record for the routing list will contain the highest return code and reason code for all of the datastore records.

## User data logging record

The Event Recording User Data Logging record is produced when user data is submitted to the journal for logging.

*Table 27. User data logging event record*

| Code (Hex) | Code (Dec) | Event description | Collection level |
|---|---|---|---|
| AB | 171 | Event Recording User Data Logging | 1 |

## Mapping event records

The CEXCR*nn* members in the *cexpre*.USERMAC library map event records. The *nn* suffix represents the event record number. You can use the member CEXCERE to generate mappings for one or more event records.

### Syntax

```
►►──label──CEXCERE──REQ=──┬─DSECT─┬──┬──────────────────────┬──────────────────────────►
                          └─AREA──┘  │         ┌─CERE────┐ │
                                     └─PFX=────┴─prefix──┴─┘

►──┬──────────────────────────┬──┬───────────────────────┬──────────────────────────►◄
   │     ┌─ALL────────────┐   │  │           ┌─NO──┐     │
   └─ID=─┴─event_number───┴───┘  └─HEXRID=───┴─YES─┴─────┘
```

## Parameters

**REQ**
The type of mapping:

**DSECT**
Assembler mapping.

**AREA**
Assembler storage area generation.

**PFX**
The prefix for generated event mapping. The name consists of the prefix and record ID. The default prefix is CERE.

**ID**
Generates the event mapping for an event with a specific record ID (in decimal). The default is ALL, which means all event mappings are generated.

**HEXRID**
Determines whether event record IDs are generated in decimal or hexadecimal.

**NO**
Generates event record IDs in decimal. This is the default.

**YES**
Generates event records IDs in hexadecimal.

# Chapter 32. Control input data set options

The control input data set enables configuration options to be provided. These options take effect when IMS Connect Extensions restarts. The input data set is specified through an optional CEXCTLIN DD statement in the IMS Connect startup job.

The active CEXCTLIN records used when restarting IMS Connect Extensions are echoed to CEXPRINT.

**Note:** If an error occurs while processing any CEXCTLIN parameter, messages are written to CEXPRINT and the job log, and IMS Connect Extensions initialization terminates.

### Notes on syntax

A statement starting with an asterisk (*) is treated as a comment. For example:

```
* This is a comment
CEXTRACE BUFSIZE=1
CEXROUTE IMSCMDROUTE=NO
ODBMDEFAULTS DEFAULTACTION=BLANKALIAS
```

Only the first instance of a control option in this file is processed. If an option such as CEXROUTE is issued more than once with different keywords, the second and subsequent statements are ignored.

If a control card is listed on more than one line, a continuation character must be placed in column 72.

### Sample control options file

The SCEXSAMP library includes a sample member named CEXCTL01. You can use this file to explore the effect of customizing different configuration options.

1. Copy this member and make any changes in your copy.
2. Specify the location of the member in the CEXCTLIN DD statement in the IMS Connect startup job.

   Changes will take effect when IMS Connect restarts.

## Parameters

The following keywords are supported in the CEXCTLIN DD statement.

### AUTOADD_DATASTORE COUNT=(*n*|20)

This option sets the number of entries in the datastore table reserved for adding datastores that are dynamically defined in IMS Connect. The maximum number of reserved entries is 99. The default number is 20. A value of 0 indicates that AUTOADD is not supported.

Reserve enough space to add the maximum number of datastores needed for the life of the IMS Connect region. When all of the reserved entries are used it is necessary to increase the AUTOADD_DATASTORE value and restart IMS Connect to provide for additional entries. In this event, consider defining new datastores in the IMS Connect configuration member so that they do not have to be dynamically added again.

## CEXROUTE

CEXROUTE recognizes the following parameters:

```
►►──CEXROUTE──┬───────────────────────────┬──────────────────────────────────────────────────►
              │               ┌─NO──┐     │
              └─IMSCMDROUTE=──┴─YES─┴─────┘

►──┬─────────────────────────────────────────────┬──┬───────────────────────────────────┬────►
   │                ┌─GLOBALFLOODSEVERE───┐      │  │               ┌─ORIGINAL─┐        │
   └─INELIGIBLEIF=──┴─GLOBALFLOODWARNING──┴──────┘  └─RBR_FAILURE=──┴─REJECT───┴────────┘

►──┬─────────────────────────────┬──┬──────────────────────────────┬──────────────────────►◄
   │              ┌─PASS───┐     │  │               ┌─PASS──┐      │
   └─RBR_NODEST=──┴─destid─┴─────┘  └─RBR_NOALIAS=──┴─alias─┴──────┘
```

**IMSCMDROUTE**

When CEXROUTE IMSCMDROUTE=YES is specified, IMS Connect Extensions will allow routing of IMS commands using the same rules that are specified for IMS transactions.

Rules-based routing is the recommended method for routing IMS commands.

**Notes:**

1. This keyword only *requests* that IMS Connect Extensions attempt to route IMS commands. The routing rules setup will determine whether IMS commands are actually routed. Specifically, an OTMA routing rule in which the message type Send-Receive Transactions is enabled must be in effect.
2. If you are attempting to route commands using transaction routing, routing of the command is conditional on the specification of the $NOTRAN$ transaction profile. If this profile is missing, IMS commands will not be candidates for transaction routing.

**INELIGIBLEIF**

The INELIGIBLEIF keyword is used to control how IMS Connect Extensions treats datastores that are experiencing a Global Flood Warning condition. The acceptable values are:

**GLOBALFLOODWARNING**

If the Global Flood Warning condition occurs (Degraded state) then the candidate datastore will become ineligible for routing as if the status were Unavailable.

**GLOBALFLOODSEVERE**

If a Global Flood Severe condition occurs (Unavailable state) then the candidate datastore will become ineligible for routing. This is the default value.

**RBR_FAILURE**

The RBR_FAILURE keyword specifies how an input message is to be handled when rules-based routing is being used and IMS Connect Extensions determines that there is no valid destination. The acceptable values are:

**ORIGINAL**

The input message is passed to IMS Connect with the destination set to the original destination. If the original destination is a generic datastore name, message HWSS0742W is issued.

**REJECT**

The input message is rejected with an RSM indicating that no destination is available.

**Note:** If INELIGIBLEIF=GLOBALFLOODWARNING is specified, and routing fails because the target datastore is in Global Flood Warning condition (Unavailable state), then the input message is always rejected with an RSM regardless of how this keyword is specified.

**RBR_NODEST**

Controls how OTMA rules-based routing functions if the inbound DestID is not found in the routing rules during the initial pass. If the option is PASS, then normal processing is performed.

If RBR_NODEST=*destid* has been specified, the inbound DestID is changed to the value specified and the rules are scanned again to see if a rule matching the DestID specified is present. If a matching rule is not found, processing continues as if no rule is matched.

**RBR_NOALIAS**

Controls how ODBM rules-based routing functions if the inbound alias is not found in the routing rules during the initial pass. If the option is PASS, normal processing is performed.

If RBR_NOALIAS=*alias* has been specified, the inbound alias is changed to the value specified and the rules are scanned again to see if a rule matching the alias specified is present. If a matching rule is not found, processing continues as if no rule is matched. Note that the routing rule for *alias* must not refer to a PSB name list.

## CEXTRACE BUFSIZE=*n*

The size of the IMS Connect Extensions trace buffer, in 1024 byte increments. The LRECL of the journal must be large enough to allow for the maximum trace record that can be written.

| If the IMS Connect SOAP Gateway exit routine (HWSSOAP1) is active, a higher
| BUFSIZE value will be needed, as XML data tends to take up a lot of space in the
| trace record.

The default is 1, or a trace buffer size of 1024 bytes. The maximum value of 20 represents a trace buffer size of 20480 bytes of user data.

**Important:** The maximum value represents a significant amount of region storage. Use the following formula to calculate trace buffer usage:

```
((4 + #ports) * (cextrace_value * 1024))
```

where:

- *#ports* is the total number of ports, including OTMA ports and DRDA ports.
- *cextrace_value* is the value specified by the CEXTRACE BUFSIZE control option.

## EVENTLOGGING WRITE=BUFFER|IMMEDIATE

This setting specifies whether to buffer event records before they are written to the journal. Buffering can improve performance by reducing the number of times records have to be written to the Journal Manager.

WRITE=IMMEDIATE causes event records to be written one record at a time. This setting minimizes the number of event records that might be lost if the IMS Connect region is canceled.

WRITE=BUFFER causes the event manager to wait until the buffer is full before writing the event records to the journal. This technique is designed to improve performance but can result in some number of event records not being logged if the IMS Connect region is canceled.

## ICON_CONTROL PORT=*nnnnn*

The command shell connects to an available IMS Connect message port to submit IMS Type-1 commands to a datastore via message exit HWSJAVA0. A dedicated port is needed if you use a TCP/IP routing device such as sysplex distributor, to prevent commands being routed to a different IMS Connect system.

Use the ICON_CONTROL PORT control option to specify a dedicated IMS Connect port which is to be used for routing IMS Type-1 commands that are issued from IMS Connect Extensions. *nnnnn* must be between 1 and 65535 and must be an OTMA port that is defined in HWSCFG. Ensure that the port is not subject to TCP/IP routing.

If an undefined port is specified a warning message CEX5130W is displayed at startup and PORT is set to the default value of 0. This means that the first port of the correct type that is defined in HWSCFG will be used.

**Note:** If all of the datastores for an IMS system are stopped then there will be no pathway for executing IMS Type-1 commands.

## MESSAGE

The MESSAGE control option controls which CEX and FUN messages are written to SYSLOG. The ADD_WTO suboption writes additional messages to SYSLOG. The DROP_CEXPRINT suboption stops specified messages being written to CEXPRINT.

```
►►──MESSAGE──┬────────────────────┬──┬───────────────────────────┬──────────────►◄
             └─ADD_WTO=(msglist)──┘  └─,DROP_CEXPRINT=(msglist)──┘
```

**ADD_WTO**

The ADD_WTO suboption accepts a list of one or more message numbers and adds the WTO destination to each message in the list. You can use this, for example, to use automation to intercept or count specific messages in SYSLOG.

Messages are truncated at 126 characters. If a message is split over multiple lines in CEXPRINT only the first line is written to the WTO destination.

**DROP_CEXPRINT**
: The DROP_CEXPRINT suboption accepts a list of one or more message numbers and removes the CEXPRINT destination from each message in the list.

*msglist* is a list of four-digit message identification numbers separated by spaces or commas. For example, the following command will write the messages FUN2515I and CEX5403E to SYSLOG and stop CEX5407I being written to CEXPRINT:

```
MESSAGE ADD_WTO=(2515,5403),DROP_CEXPRINT=(5407)
```

The message CEX5151I is displayed in CEXPRINT to show when the message handler exit becomes active or inactive. Messages issued while the message handler exit is inactive cannot be suppressed or redirected.

## ODBMDEFAULTS DEFAULTACTION=INPUTALIAS|BLANKALIAS

Specifies the default behavior for ODBM routing.

**INPUTALIAS**
: The input alias is copied as the output alias and the ODBM name is set to blanks. This is also the behavior of the HWSROUT0 exit supplied with IMS Connect.

**BLANKALIAS**
: The output alias and the ODBM name are both set to blanks.

## SECURITY

The SECURITY control option recognizes the following parameters:

```
              ┌─CACHED_USER_MAX=10000─┐  ┌─PWCASE=SYSTEMDEFINITION─┐
►►──SECURITY──┼───────────────────────┼──┼─────────────────────────┼──────────────►
              └─CACHED_USER_MAX=n─────┘  ├─PWCASE=AUTOMATIC────────┤
                                         ├─PWCASE=UPPER────────────┤
                                         └─PWCASE=MIXED────────────┘

   ┌─VALIDATE_TRUSTED=NO──┐
►──┼──────────────────────┼──────────────────────────────────────────────────────►◄
   └─VALIDATE_TRUSTED=YES─┘
```

**CACHED_USER_MAX=*n***
: The CACHED_USER_MAX keyword specifies the maximum number of cached users when IMS Connect Extensions security is activated and ACEE caching has been requested. The number specified is converted to the initial size of the ACEE cache pool. The ACEE cache pool is allowed to expand beyond the initial value if needed to accommodate additional user IDs in the ACEE cache. The number can be in the range 1000 - 2000000. The default value is 10000.

**PWCASE=SYSTEMDEFINITION|AUTOMATIC|UPPER|MIXED**
: The PWCASE control option determines how mixed-case passwords are handled by IMS Connect Extensions on this system. You can use this to specify that passwords are to be translated to uppercase, or left as mixed case, or else the setting from the external security manager is to be used.

  **SYSTEMDEFINITION**
  : The setting from the Activate Mixed Case Passwords option in the system definition is used. This is the default.

**AUTOMATIC**
The setting for mixed-case password support in the external security manager is used. To obtain this, a SAF call is issued to return the MIXDCASE setting from the SETROPTS (RACF options) data.

**UPPER**
Passwords are to be translated to uppercase.

**MIXED**
Passwords are to be treated as mixed case.

**VALIDATE_TRUSTED**
In the system definition, if **Activate Security** and **Activate Validation** are selected and **Validation type**=2, IMS Connect Extensions will check the user ID associated with an incoming OTMA transaction and reject the message if RACF returns an invalid status.

The VALIDATE_TRUSTED keyword determines whether IP validation occurs when the message has the OMUSR_TRSTUSR (trusted user) flag set. When VALIDATE_TRUSTED=NO is specified, messages from trusted users are not validated. When VALIDATE_TRUSTED=YES is specified and IP validation (type 2) is in effect, messages are validated even if the OMUSR_TRSTUSR flag is set.

## SERVICE_EXIT CHECK_MSG_HDR=N|Y

This controls the behavior of the IMS Connect Extensions Service exit (CEXSVC01) when inspecting the inbound message header. The default is N, which means that the message header must be encoded in EBCDIC. A value of Y indicates that the service exit is to accept either ASCII or EBCDIC encoding in the message header.

## ZIIP_OFFLOAD OPTION=N|Y|A

Determines if IMS Connect Extensions is to offload event collection processing to available ZIIP processors.

**N** Do not perform ZIIP offload processing. This is the default.

**Y** Enable ZIIP offload processing even if there are no ZIIP processors available on the LPAR.

**A** Enable ZIIP offload processing only if there is at least one ZIIP processor available on the LPAR.

# Chapter 33. 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).

```
►►──required_item──────────────────────────────────────────────►◄
```

- Optional items appear below the main path.

```
►►──required_item──┬───────────────┬───────────────────────────►◄
                   └─optional_item─┘
```

  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.

```
                   ┌─optional_item─┐
►►──required_item──┴───────────────┴───────────────────────────►◄
```

- 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.

```
►►──required_item──┬─required_choice1─┬────────────────────────►◄
                   └─required_choice2─┘
```

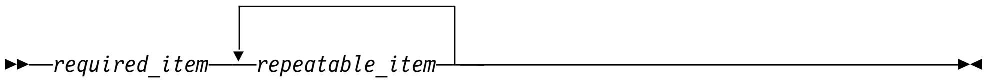
  If choosing one of the items is optional, the entire stack appears below the main path.

```
►►──required_item──┬──────────────────┬────────────────────────►◄
                   ├─optional_choice1─┤
                   └─optional_choice2─┘
```

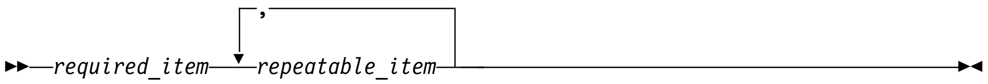
  If one of the items is the default, it appears above the main path, and the remaining choices are shown below.

```
                   ┌─default_choice──┐
►►──required_item──┼─────────────────┼─────────────────────────►◄
                   ├─optional_choice─┤
                   └─optional_choice─┘
```

- 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.

- 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).

# Part 9. Appendixes

# Notices

This information was developed for products and services offered in the U.S.A.

This material may 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 give you any license to these patents. You can send license inquiries, in writing, to:

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

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

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

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:** INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This 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 Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

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

Licensees of this program who 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 Corporation
J64A/G4
555 Bailey Avenue
San Jose, CA 95141-1003
U.S.A.

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 information 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.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise 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.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

## Programming interface information

This publication primarily documents information that is NOT intended to be used as Programming Interfaces of IMS Connect Extensions.

This publication also documents intended Programming Interfaces that allow the customer to write programs to obtain the services of IMS Connect Extensions. This information is identified where it occurs by an introductory statement to a topic or section.

## Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered marks 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.

Windows is a trademark of Microsoft Corporation in the United States, other countries, or both.

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

## Terms and conditions for product documentation

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

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

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

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

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

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

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

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

## Privacy policy considerations

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

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

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

For more information about the use of various technologies, including cookies, for these purposes, see IBM’s Privacy Policy at http://www.ibm.com/privacy and IBM’s Online Privacy Statement at http://www.ibm.com/privacy/details the section entitled “Cookies, Web Beacons and Other Technologies” and the “IBM Software Products and Software-as-a-Service Privacy Statement” at http://www.ibm.com/software/info/product-privacy.

# Index

Product Number:  5655-S56

SC19-4364-01