IBM IMS Connect Extensions for z/OS
Version 3 Release 1

User's Guide

IBM
Contents

About this information ........................................ ix

Part 1. Getting started ........................................ 1

Chapter 1. Overview .......................................... 3
What’s new in IMS Connect Extensions ....................... 4
Changes for Version 3 Release 1 ............................. 4
Changes for Version 2 Release 4 ............................. 5
Features and benefits of IMS Connect Extensions ........... 8
IMS Connect Extensions components and architecture .... 11
IMS Connect Extensions components ....................... 11
Architecture .................................................. 14
IMS Connect Extensions utilities ............................ 19
Related IBM tools .......................................... 20
IMS Connect Extensions terminology ....................... 21
Application management solutions ........................... 22
Service updates and support information ................. 23
IMS Connect Extensions documentation and updates .... 23
Accessibility features ......................................... 24

Chapter 2. Installing and customizing IMS Connect Extensions ........................................ 25
Customization overview ....................................... 25
   Hardware and software prerequisites ................... 27
   IMS Connect Extensions product libraries .............. 27
Planning your customization .................................. 29
Reviewing control options ................................... 30
Configuring the ODBMROUT exit member .................... 31
Configuring the Open Database security exit .............. 31
Creating an initial configuration for IMS Connect Extensions ........................................ 32
APF-authorizing load libraries and the MODIFY command ........................................ 35
Modifying and submitting the IMS Connect startup job ........................................ 35
Verify the installation of IMS Connect and IMS Connect Extensions ........................................ 37
   Viewing the IMS Connect system in IMS Connect Extensions ........................................ 37
   Running IVP jobs to generate sample OTMA workload ........................................ 38
   Running IVP jobs to generate sample Open Database workload ........................................ 40
Advanced ISPF dialog configuration ......................... 42
   CEXOREXX ................................................. 42
   Configuring libraries to be invoked dynamically .... 42
   Configuring libraries to be invoked statically ....... 42
   Overriding the data set low level qualifiers ........... 43
Upgrading to IMS Connect Extensions V3.1 ............... 44
Installing the IMS Connect Extensions Operations Console ........................................ 44

Chapter 3. User interface options ......................... 47
IMS Connect Extensions ISPF dialog ....................... 47
   ISPF environment considerations ....................... 47
   Online Help ............................................. 48
Operations Console for z/OS Explorer ..................... 49
   Eclipse environment considerations .................. 50
   Online Help ............................................. 51

Part 2. IMS Connect event collection, reporting, and forwarding ........................................ 53

Chapter 4. Event collection ................................ 55
Active journals and archive journals ........................ 55
Types of event record ........................................ 56
Sample event flows in IMS Connect ......................... 58
Configuring event collection and journals ................... 61
   Estimating the size of active journals ......... 61
   Naming archive journal data sets ............... 62
Defining and managing active and archive IMS Connect Extensions journals .......................... 63
   Creating an archive JCL skeleton .................. 63
   Configuring active journals ....................... 68
   Creating an archive journal template ........... 69
Resizing or redefining active journals ....................... 70

Chapter 5. Reporting IMS Connect activity ........... 73
Reporting and analysis with IMS Problem Investigator .... 73
   Browse events recorded in the IMS Connect Extensions journal .......................... 73
   Merge IMS and IMS Connect log files ............. 78
Using IMS Problem Investigator in a Open Database environment ........................................ 80
Reporting and analysis with IMS Performance Analyzer ........................................ 82
   IMS Connect transaction transit reports produced by IMS Performance Analyzer .... 83
   IMS Connect transit extract produced by IMS Performance Analyzer ................... 85
   IMS Connect resource usage reports produced by IMS Performance Analyzer .......... 85
   IMS Connect trace reports produced by IMS Performance Analyzer ................. 89
   Reporting on OTMA workload routing in IMS Connect using IMS Performance Analyzer .......... 90

Chapter 6. Tracing message flows ....................... 93
Using the IMS Connect Extensions trace .................. 93
Controlling the IMS Connect Recorder Trace facility .... 96
Chapter 13. OTMA operations management and tuning .............................. 273
Suspending OTMA workload routing to IMS data stores using drain ........ 273
Dynamically adding IMS data stores as targets for OTMA routing ............ 274
Rebalancing sessions across IMS Connect systems ............................ 276
Managing IMS Connect transaction options ..................................... 277

Chapter 14. ODBM workload routing in IMS Connect .............................. 279
Principles of ODBM rules-based routing ....................................... 279
Developing ODBM routing rules for IMS Connect ............................ 281
Routing ODBM workload to one or more IMS data stores .................. 282
Routing ODBM workload to a list of fallback IMS data stores .......... 287
Routing ODBM workload according to PSB name ....................... 288
Setting the capacity of an ODBM target .................................. 290
Assigning ODBM routing rules to a plan .................................. 293
Activating ODBM routing rules in IMS Connect ............................ 294
Reviewing the active ODBM routing rules in IMS Connect .................. 297
Controlling ODBM routing behavior via CEXCTLIN control options ........ 299

Chapter 15. Enhancing IMS Connect security .................................... 301
Caching user credentials to improve performance .......................... 301
Validating an incoming OTMA transaction or DRDA request ............ 302
Creating workload rules for specific IP addresses ...................... 305
Security preprocessing exit .................................................. 308

Chapter 16. Customizing IMS Connect message translation .................. 311

Part 5. Administering IMS Connect Extensions definitions .................. 313

Chapter 17. Administering IMS Connect Extensions definitions with the ISPF dialog .................................................. 315
Definition list panels: common features ..................................... 318
Definition edit panels: primary commands .................................. 319
Defining IMS Connect systems .............................................. 320

Chapter 18. Definition utilities ...................................................... 375
Definition takeup utility ....................................................... 375
TAKEUP command ......................................................... 377
SET command .............................................................. 378
Definition maintenance utility .................................................. 379
ADD command ............................................................. 380
DELETE command ......................................................... 413
SET command ............................................................. 414
Definition extract utility ...................................................... 415
TRANSFORM command ..................................................... 416
EXTRACT command ......................................................... 419
SET command ............................................................. 420
Definition import/export utility ................................................ 421
EXPORT command ........................................................ 422
IMPORT command ........................................................ 424
VERIFY command .......................................................... 425
SET command .............................................................. 426

Part 6. Application programming interfaces to IMS Connect

Chapter 19. IMS Connect Extensions host command environment for REXX 429
ADD command ............................................................. 429
CLEAR command .......................................................... 430
CONNECT command ......................................................... 431
DELAY command ............................................................ 433
DRAIN command ........................................................... 434
OPTION command .......................................................... 435
QUERY commands .......................................................... 436
QUERY TYPE=ACEE_CACHE command .................................... 436
QUERY TYPE=JOURNAL command ....................................... 438
QUERY TYPE=PENDING_RESPONSES command ...................... 438
QUERY TYPE=PLAN command .......................................... 440
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

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

IBM® IMS™ Connect Extensions for z/OS® (referred to as IMS Connect Extensions) provides essential features for advanced control of IMS Connect. In this section, you will learn about the features and benefits of IMS Connect Extensions and how to install and configure the product.

Topics:

- Chapter 1, “Overview,” on page 3
- Chapter 2, “Installing and customizing IMS Connect Extensions,” on page 25
- Chapter 3, “User interface options,” on page 47
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 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 use this information to debug clients and new applications. Use the IMS Connect Extensions feed to expose near real-time IMS Connect transaction summary data to analytics platforms. This includes information regarding the message’s origin, its destination, and the overall response time.

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 from the IMS Connect Extensions Operations Console for z/OS Explorer. Use the IMS Connect Extensions host command environment for REXX to automate operations management tasks as desired.

Enhanced workload management
Dynamic management of TCP/IP transactions, allowing you to define routing rules to automatically balance and distribute workloads and reroute messages when IMS system failures occur. Temporarily suspend routing of messages to an IMS data store to reduce the likelihood of transactions failing or being rejected when IMS is shut down. Ability to assign selected OTMA routing rules to a routing plan and dynamically switch between routing plans. Automatically response to changes in the IMS environment such as dynamically added IMS data stores and flood conditions. 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.

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. Use IP address rules to treat messages coming from an IP address as a trusted user or to assign a specific user ID to the request.
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.

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 3 Release 1
This topic lists the technical changes that were introduced in or incorporated into IMS Connect Extensions Version 3 Release 1.

Expose near real-time IMS Connect transaction summary data to analytics engines
Analyze IMS Connect transaction summary data in analytics platforms such as Splunk using the IMS Connect Extensions feed. This information can be formatted as JSON Lines, written to SMF, or by writing it to data sets for use in IBM IMS Performance Analyzer for z/OS, IBM IMS Problem Investigator for z/OS, and IBM Transaction Analysis Workbench for z/OS. For more information on the IMS Connect Extensions feed, see Chapter 7, “Forwarding a live feed of IMS Connect events,” on page 99.

Enhancements to the IMS Connect Extensions status monitor to display socket and journal usage
Users can now view socket and IMS Connect Extensions journal usage in the IMS Connect Extensions ISPF dialog and Operations Console for z/OS Explorer. For more information on ISPF changes, see “Using the Operations dialog” on page 120. For more information on the Operations Console, see “Displaying statistics on IMS Connect activity” on page 188.

Query journal, socket, and status monitor resource statuses and statistics in REXX
Users can now query the status of the IMS Connect Extensions journal, IMS Connect socket usage, and statistics supplied by the status monitor using the IMS Connect Extensions host command environment for REXX. Several new REXX samples that demonstrate just some of the ways in which these queries can be used for enhanced automation of your systems can be found in the SCEXSAMP library. For more information, see “QUERY commands” on page 436. For new REXX samples that demonstrate how to use these features, see “REXX automation samples for the IMS Connect Extensions host command environment” on page 223.

IP address rules
IP address rules allow you to specify what preconditioning is to take place on workloads coming from specific IP addresses. Use IP address rules to treat messages coming from an IP address as a trusted user or to assign a specific user ID to the request. See “Creating workload rules for specific IP addresses” on page 305.
Changes to CONNECT host command for REXX

The CONNECT host command for REXX has been modified in the following ways:

- The APPLID parameter is no longer required.
- PASSTIK=YES is now the default authenticator.

See “CONNECT command” on page 431.

Changes to TRACE host command for REXX

The TRACE host command for REXX has been modified to allow you to more easily modify an active trace by making the default values apply only when a trace is not currently active. See “TRACE command” on page 456.

Renamed sample job CEXCMDS for access control configuration

Sample job CEXCMDS that implements IMS Connect Extensions resource profiles has been renamed to CEXRACF. See “Example: typical access control configuration” on page 240.

Support for PassTickets in IMS Open Database (APAR PH01608)

Support in IMS Connect Extensions security to allow IMS Connect DRDA clients to pass in a PassTicket instead of a password for user authentication in IMS Connect for accessing IMS DB through ODBM.

Wildcard support for the DESTID in OTMA routing rules (APAR PI79483)

Users can now define OTMA routing rules that can service a set of transactions by specifying wildcards (*) that can match multiple IRM_IMSDestID field values. This reduces the number of rules required to support a range of IRM_IMSDestIDs that need to be handled in the same way. For more information, see “Defining OTMA routing rules” on page 356 and “OTMA routing rule (RTG) ADD command” on page 405.

Installation requirements for product library SFUNLINK

Product library SFUNLINK previously supplied with IMS Connect Extensions V2.4 is now supplied in IBM Common Services Library for z/OS (5655-CSL). Refer to the IMS Connect Extensions program directory for installation instructions.

IMS Connect Extensions Operations Console for IBM Explorer for z/OS (z/OS Explorer)

The IMS Connect Extensions Operations Console is now available for direct download via the Internet. For the most up to date information, visit the Install Eclipse Tools page on the IBM Developer website.

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, PI51458, PI59534, PI43276, PI26433.

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. See “MESSAGE option” on page 566.
• **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. See “SECURITY option” on page 568.

**Changes to the IMS Connect Extensions Operations Console**

• CSV export has been rewritten to conform to RFC 4180. “Exporting data to a CSV file” on page 185.
• Performance improvements in handling of large editors.
• Accessibility improvements.
• Functional Support Library renamed to Common Services Library feature.

**Miscellaneous updates**

• Purge ACEE cache entries based on ENF 71 notification of changes to a user’s security profile. See “Caching user credentials to improve performance” on page 301.
• Automatic detection of mixed-case passwords setting when an IMS Connect system starts up.
• New WARNRC0 option on Archive Manager utility sets the return code to zero after a non-critical warning.

**SC19-4364-00**

IMS Connect Extensions host command environment for REXX supersedes the batch command utility

In IMS Connect Extensions 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 IMS Connect Extensions V2.4, the new IMS Connect Extensions 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.

• 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 IMS Connect Extensions 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 data store 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** command activates or deactivates the OTMA routing plan or ODBM routing plan.
• **UPDATE** command allows certain settings, such as data store 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. See Chapter 10, “Automating IMS Connect operations with REXX,” on page 221.

Routing enhancements

- Ability to group OTMA routing rules and ODBM routing rules into routing plans, and to switch dynamically from one routing plan to another. See “Assigning OTMA routing rules to a plan” on page 257.

- The pre-routing user exit for rules-based routing provides a customizable service that can be used to prevent messages being routed to specific data stores based on the availability or health of a specific transaction on the IMS system. See “Pre-routing user exit” on page 269.

- Session drain allows safe closure of specific sessions without interrupting in-flight messages. See “Active Sessions” on page 130.

- 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. See “Using the Operations dialog” on page 120.

- Ability to assign data stores and Open Database targets a relative processing capacity of zero, making them ineligible as routing candidates. See “Setting the capacity of an IMS data store for OTMA workloads” on page 254.

Improvements to the IMS Connect Extensions Operations Console

- Enhancements to Status Monitor:
  - **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 including Start Time, OTMA Routing Plan, ODBM Routing Plan, Event Collection Level, Message Limit, and Message Limit Threshold.

  See “Managing IMS Connect systems” on page 194.

  - **Datastores** tab
    - New **Update Capacity Weight** context option.
    - New fields including: Member, TMember, and XCF Group.

  See “Managing IMS data stores” on page 204.

- Enhancements to Sessions:
  - New **Drain Eligible Sessions** context option allows safe closure of specific sessions without interrupting in-flight messages.
  - New fields including: Socket Type, In IMS Conversation, Commit Mode, Synch Level, Resume TPIPE, Alternate Transaction Used, Alternate Transaction Length, Alternate Transaction Offset, and Alternate Transaction Code.

  See “Managing active sessions” on page 211.
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.

---

**Features and benefits of IMS Connect Extensions**

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 Operations Console for z/OS Explorer 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 IMS data stores across all of your systems for OTMA flood conditions.

Combined with IBM IMS Problem Investigator for z/OS or IBM Transaction Analysis Workbench for z/OS, IMS Connect Extensions enables rapid problem determination, including the ability to view important state event information such as IMS data store 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 data stores and reroute all messages to an alternate IMS data store when a given IMS data store 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.

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 the Operations Console for z/OS Explorer. 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 IMS data store 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:
- Suspend routing of messages to an IMS data store, 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.
- Set session message limits to specify the maximum number of input messages for a persistent session before the session is automatically closed. In an IMS sysplex distributor environment, if you design your applications to automatically reconnect when the session is closed, new persistent sessions will be routed to the IMS Connect with the lowest session totals. This feature is particularly useful when an IMS Connect system is being brought back online to quickly rebalance your persistent sessions across IMS Connect systems.
- Develop, set, and dynamically adjust routing plans for OTMA and ODBM workloads and adjust distribution of work according to the
capacity of each IMS data store or ODBM target. These can be used to respond to differing periods of demand or planned maintenance outages.

- Respond to changing data store conditions such as processing status and number of active connections.
- Control access to IMS Connect instances via a system authorization facility (SAF) security class.
- Refresh IMS Connect SAF security profiles dynamically.
- Control access to an OTMA transaction or DRDA request based on the client IP address and IMS Connect port number.
- Add, delete, or reload some IMS Connect message exits without interrupting IMS Connect.
- Dynamically add data store capacity and adjust workload balancing profiles.

**Automate IMS Connect operations with REXX**

Use the IMS Connect Extensions host command environment for REXX to automate IMS Connect operational tasks and to integrate IMS functions from other REXX command interfaces such as the IMS REXX SPOC API and the SDSF REXX programming interface. Use the host command environment for REXX to automatically respond to scenarios such as an IMS Connect system approaching MAXSOC, or when an IMS Connect system is being brought back online and you want to use session limits to quickly rebalance workload across IMS Connect systems.

**Expose near-real time IMS Connect transaction summary data to analytics engines**

Monitor IMS Connect, track trends, and to visualize different workloads in your topology using the IMS Connect Extensions feed and the features of your analytics platform.

**Record and report IMS Connect activity with IBM IMS Performance Analyzer for z/OS**

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 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 data stores and ports.
- Throughput information for different programming models; for example, conversational, non-conversational, and Send Only.
- Resource availability.

**Enhance security for IMS Connect**

IMS Connect Extensions enhances the security features of IMS Connect with the following security options:

- Verify the supplied authenticator (password, PassTicket, or phrase) for the user ID associated with the incoming message.
- Check if the user ID associated with the incoming message is authorized to access IMS Connect and/or validate the source IP address.
- Cache ACEE security control block for improved performance.
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.

**IMS Connect Extensions components**

This topic describes the main components of IMS Connect Extensions.

![Diagram of IMS Connect Extensions components](image)

**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 for z/OS Explorer

An Eclipse-based PC application for z/OS Explorer 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.

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 IMS Connect Extensions journals which can then be used by other tools such as IBM IMS Problem Investigator for z/OS and IBM IMS Performance Analyzer for z/OS for in-depth analysis and performance reporting. 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, IMS data stores, ODBMs, aliases, and ports, including data stores and ports that are added dynamically in IMS. Status monitor statistics are available through the ISPF Operations dialog or Operations Console for z/OS Explorer.

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 for z/OS Explorer, or the IMS Connect Extensions publisher API. Through the ISPF Operations dialog and Operations Console you can view details of active sessions and stop unresponsive sessions.

Active IMS Connect Extensions journals, archive IMS Connect Extensions 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 may be kept as sequential data sets or as data sets in a generation.
data group (GDG), 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, IBM Transaction Analysis Workbench, and IMS Performance Analyzer to provide debugging, performance analysis, and auditing purposes.

**IMS Connect Extensions feed**

The IMS Connect Extensions feed exposes near real-time IMS Connect transaction summary data to analytics engines. This includes information regarding the message’s origin, its destination, and the overall response time. This information can be formatted as JSON Lines, written to SMF, or by writing it to data sets for use in IMS Performance Analyzer, IMS Problem Investigator, and IBM Transaction Analysis Workbench. This method avoids the need to post-process IMS Connect Extensions journals in IMS Performance Analyzer to generate IMS Connect transaction indexes (containing CA20 event records). The IMS Connect Extensions feed is supplied data by the publisher API. The publisher API also supports the IMS Connect monitoring function of OMEGAMON® for IMS on z/OS.

**Batch utilities**

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

**IMS Connect Extensions host command environment for REXX**

The IMS Connect Extensions 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.

**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 5, 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.

As shown in Figure 6, 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.

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.

**IMS Connect Extensions 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.

![Diagram of IMS Connect Extensions event collection architecture: OTMA processing](image)

*Figure 7: IMS Connect Extensions event collection architecture: OTMA processing*

The following figure shows the equivalent architecture for ODBM processing.
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.

Use IBM IMS Performance Analyzer for z/OS and IBM Problem Investigator for z/OS to report on events collected by IMS Connect Extensions.

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.

Figure 8. IMS Connect Extensions event collection architecture: ODBM processing

Related concepts:
Chapter 4, “Event collection,” on page 55
Chapter 5, “Reporting IMS Connect activity,” on page 73
As shown in Figure 10, 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.

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.

Figure 9. IMS Connect Extensions repository and IMS Connect configuration

Figure 10. IMS Connect Extensions repositories
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 tasks:
“Creating an initial configuration for IMS Connect Extensions” on page 32
Use the Definitions Setup dialog to take up definitions from the IMS Connect configuration member into an IMS Connect Extensions repository. This provides you with an initial configuration and a basis from which you can start customizing IMS Connect Extensions.

**IMS Connect Extensions utilities**

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

**IMS Connect Extensions host command environment for REXX**
The IMS Connect Extensions 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 suspend or resume routing to an IMS data store, query the status of the IMS Connect Extensions journal and perform a switch, control the IMS Connect Extensions trace, retrieve information on IMS Connect socket usage, TCP/IP ports, IMS Connect, IMS data stores, user exits, ODBM targets, IMS aliases, MSC link statistics, and remote IMS Connect system statistics, refresh and update IMS Connect Extensions repository definitions, and run IMS Connect commands or IMS type-1 commands on a specified target system or IMS data store.

- 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 for z/OS Explorer 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 the IMS Connect Recorder Trace facility 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.

IBM Transaction Analysis Workbench for z/OS
Transaction Analysis Workbench for z/OS is a tool for analyzing problems with the performance or behavior of transactions on z/OS. You can analyze transaction logs in an ISPF-based log browser or in off-host analytics dashboards. Analyze records from multiple z/OS subsystems, including IBM CICS®, IBM Db2®, IBM IMS, IBM MQ, SMF, IBM WebSphere® Application Server for z/OS, and IBM z/OS Connect.
IBM 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:
Chapter 5, “Reporting IMS Connect activity,” on page 73
Use IBM IMS Performance Analyzer for z/OS and IBM Problem Investigator for z/OS to report on events collected by IMS Connect Extensions.

**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, Operations Console for z/OS Explorer, 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 feed**

A component of IMS Connect Extensions that forwards IMS Connect transaction summary data to analytics platforms. This information can be formatted as JSON Lines, written to SMF, or by writing it to data sets for use in IMS Performance Analyzer, IMS Problem Investigator, and IBM Transaction Analysis Workbench.

**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 IMS data store for a transaction. IMS Connect Extensions can generate a list of candidate data stores based on the transaction type, transaction code, and data store 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 IMS data stores, and to balance DRDA requests across multiple databases. IMS Connect Extensions can select a data store 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:


**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:


You can also access documentation for many IMS Tools from IBM Knowledge Center:


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


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:


**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:

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.

Note: All steps are required unless noted otherwise.

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
<th>Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Determine the type of customization that is appropriate for your environment and gathered information.</td>
<td>“Planning your customization” on page 29</td>
</tr>
<tr>
<td>2</td>
<td>Review the default control option settings to see whether they best suit your requirements.</td>
<td>“Reviewing control options” on page 30</td>
</tr>
<tr>
<td>3</td>
<td>Start the ISPF interface and configure the definitions repository and load libraries. 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.</td>
<td>“Creating an initial configuration for IMS Connect Extensions” on page 32</td>
</tr>
<tr>
<td>4</td>
<td>APF-authorize the IMS Connect Extensions load libraries and the MVS MODIFY command.</td>
<td>“APF-authorizing load libraries and the MODIFY command” on page 35</td>
</tr>
<tr>
<td>5</td>
<td>Modify the IMS Connect startup job to include IMS Connect Extensions libraries and data sets.</td>
<td>“Modifying and submitting the IMS Connect startup job” on page 35</td>
</tr>
<tr>
<td>6</td>
<td>Verify that IMS Connect Extensions is installed in IMS Connect and is accessible from the Operations dialog.</td>
<td>“Viewing the IMS Connect system in IMS Connect Extensions” on page 37</td>
</tr>
<tr>
<td>7</td>
<td>If you are upgrading from an earlier version of IMS Connect Extensions, see if you need to perform any additional steps. (Optional)</td>
<td>“Upgrading to IMS Connect Extensions V3.1” on page 44</td>
</tr>
<tr>
<td>8</td>
<td>Install Operations Console plug-in on distributed workstations.</td>
<td>“Installing the IMS Connect Extensions Operations Console” on page 44</td>
</tr>
</tbody>
</table>
## Configuring routing and security

This table provides links to information on customizing IMS Connect Extensions to route OTMA or Open Database (ODBM) 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.

<table>
<thead>
<tr>
<th>Feature/workload</th>
<th>Optional or workload-dependent customization</th>
<th>Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>OTMA routing</td>
<td>Verify that communications are configured to allow monitoring and routing of IMS request messages through TCP/IP connections.</td>
<td>“Running IVP jobs to generate sample OTMA workload” on page 38</td>
</tr>
<tr>
<td>OTMA routing</td>
<td>Configure OTMA routing rules and IMS data stores; customize routing of IRM messages.</td>
<td>Chapter 12, “OTMA workload routing in IMS Connect,” on page 245</td>
</tr>
<tr>
<td>Required for Open Database routing</td>
<td>Configure the IMS Connect BPE environment to use the CEXROUT0 exit.</td>
<td>“Configuring the ODBMROUT exit member” on page 31</td>
</tr>
<tr>
<td>Open Database (ODBM); security (optional)</td>
<td>Configure the IMS Connect BPE environment to use the CEXAUTH0 exit to perform user authentication for connections to IMS DB.</td>
<td>“Configuring the Open Database security exit” on page 31</td>
</tr>
<tr>
<td>Open Database (ODBM) routing</td>
<td>Verify that communications are configured to allow monitoring and routing of DRDA requests through TCP/IP connections.</td>
<td>“Running IVP jobs to generate sample Open Database workload” on page 40</td>
</tr>
<tr>
<td>Open Database (ODBM) routing</td>
<td>Configure ODBM routing rules and ODBM targets; customize routing of DRDA requests.</td>
<td>Chapter 14, “ODBM workload routing in IMS Connect,” on page 279</td>
</tr>
<tr>
<td>IMS Connect security</td>
<td>Review and configure IMS Connect Extensions enhancements to IMS Connect security.</td>
<td>Chapter 15, “Enhancing IMS Connect security,” on page 301</td>
</tr>
<tr>
<td>IP address rules (optional)</td>
<td>Review and configure IMS Connect Extensions workload rules for specific IP addresses.</td>
<td>“Creating workload rules for specific IP addresses” on page 305</td>
</tr>
</tbody>
</table>

## 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.

<table>
<thead>
<tr>
<th>Feature/workload</th>
<th>Optional or workload-dependent customization</th>
<th>Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event collection</td>
<td>Customize event collection and journaling. For example, control what events are recorded and create the archive JCL member.</td>
<td>“Configuring event collection and journals” on page 61</td>
</tr>
<tr>
<td>All</td>
<td>Automate IMS Connect operations tasks.</td>
<td>Chapter 10, “Automating IMS Connect operations with REXX,” on page 221</td>
</tr>
<tr>
<td>OTMA</td>
<td>Change how IMS Connect translates messages received from clients between character sets.</td>
<td>Chapter 16, “Customizing IMS Connect message translation,” on page 311</td>
</tr>
<tr>
<td>All; security</td>
<td>Customize access controls to IMS Connect z/OS and WTOR commands and IMS Connect Extensions commands.</td>
<td>“Controlling access to commands in your external security manager” on page 238</td>
</tr>
<tr>
<td>All</td>
<td>Customize the ISPF dialog.</td>
<td>“Advanced ISPF dialog configuration” on page 42</td>
</tr>
</tbody>
</table>
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 products:

- 5650-zOS z/OS, V2.2 or later
- 5655-G44 SMP/E for z/OS, V3.6 or later
- 5655-CSL IBM Common Services Library for z/OS, V1.1 or later with APAR PH04718 applied.

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.

IMS Connect Extensions Operations Console for z/OS Explorer is an Eclipse plug-in. It is designed to operate on Microsoft Windows platforms supported by z/OS Explorer Version 3.1 or later. For IMS Connect Extensions Operations Console to operate, IMS Connect Extensions 2.4.0 or later must be installed on all IMS Connect systems that you want to connect to.

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.
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**  
IBM Common Services Library for z/OS (5655-CSL) load module library for the IMS Connect Extensions server components and ISPF client. The Common Services Library contains common functions 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.

**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.

---

*Figure 11. Product libraries used by each IMS Connect Extensions component*
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.

<table>
<thead>
<tr>
<th>Item</th>
<th>Enter your value</th>
<th>Installation parameter/ DD statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMS Connect Extensions load libraries (SMP/E-controlled)</td>
<td>(Example: cxpre.SCEXLINK where cxpre is the high-level qualifier for the IMS Connect Extensions product library)</td>
<td></td>
</tr>
<tr>
<td>IBM Common Services Library for z/OS (5655-CSL) load libraries (SMP/E-controlled)</td>
<td>(Example: cslpre.SFUNLINK where cslpre is the high-level qualifier for the IBM Common Services Library product library)</td>
<td></td>
</tr>
<tr>
<td>Name of the target definitions repository you will be creating</td>
<td></td>
<td>your.repos in //CEXREPOS DD</td>
</tr>
<tr>
<td>Name of the control options data set you will be creating (optional)</td>
<td></td>
<td>your.options.ds in //CEXCTLIN DD</td>
</tr>
</tbody>
</table>

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.

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

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

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

**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 26, “Control input data set options,” on page 561.
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. 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.

CEXROUT0 replaces the user-written IMS Connect DB Routing user exit routine (HWSROUT0).

Procedure

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=ODBMROUT,EXITS=(CEXROUT0),ABLIM=8,COMP=HWS)

Results

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

Related concepts:

“Developing ODBM routing rules for IMS Connect” on page 281

To route ODBM workloads in IMS Connect, you'll need to create one or more ODBM routing rules that depend on several supporting definitions in the IMS Connect Extensions repository. The definitions you build will depend on the routing scenario you would like to create.

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

CEAUTH0 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 302

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 an IMS Connect Extensions repository. This provides you with an initial configuration and 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.
- 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 'cexpr.SCEXEXEC(CEXOREXX)' 'cexpr ENU'
   where cexpr is the high-level qualifier for the IMS Connect Extensions product libraries.
   For example:
   EX 'CEX.V3R1M0.SCEXEXEC(CEXOREXX)' 'CEX.V3R1M0 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 42 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 29. 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 Common Services Library using the information prepared in “Product libraries” on page 29. Review the other profile settings such as the time zone and warning override.

4. Select option 1.5 Definitions Setup.
Complete the fields in the dialog using the information you prepared in “IMS Connect startup job” on page 30 and the host name and console listener port in “Client connections” on page 29.

Note: If journal archiving is required, the Archive JCL member must already exist.

5. Enter the EDIT command to generate JCL to perform the setup. 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. Enter the **SUB** command to submit the job.

7. Look in the CEXPRINT output from step 5 on page 33 for this message:

| CEX5501I Updates completed successfully |

This message confirms that IMS Connect Extensions has configured definitions for the IMS Connect system, its IMS data stores, 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.

8. From the IMS Connect Extensions primary menu, select option 1 **Definitions** and then option 1 **System Definitions**. The IMS Connect system you specified is displayed.

**Results**

The setup job has created an example configuration based on your current environment and on the settings you specified in step 5 on page 33. This configuration includes providing access to advanced features such as commands and monitoring.

**What to do next**

To create additional definitions, you can use the following methods:
- Run the TAKEUP step again, this time on a different configuration member. You can target the same IMS Connect Extensions repository, or use a different repository as desired.
- Use the IMS Connect Extensions ISPF dialog. See Chapter 17, “Administering IMS Connect Extensions definitions with the ISPF dialog,” on page 315.

**Tip:** From the System Definition panel, you can use the following command to create a new IMS Connect repository definition modeled on an existing definition:

```
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.

**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.
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.
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 (SCEXLINK) and IBM Common Services Library for z/OS load library (SFUNLINK) in front of the IMS Connect load library (1). Replace cexpre and csilpre with the high-level qualifiers for each product library. 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 Common Services Library 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.
Verify the installation of IMS Connect and IMS Connect Extensions

This topic explains several ways to verify your installation of IMS Connect Extensions with IMS Connect.

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 Operations dialog, select the following options:
   a. In the View field, select option 2 Systems.
   b. In the Exclude inactive systems field, select option 2 No.
3. Confirm that your IMS Connect system is active. To do this, check the value in the Status column is being reported as ACTIVE.

Tip: If your system is not being reported as active, see “Connection failure between a client and an IMS Connect Extensions console listener” on page 483.

Figure 15. An active IMS Connect system displayed in the Operations panel in the IMS Connect Extensions ISPF dialog

Related tasks:
Use the primary menu of the IMS Connect Extensions ISPF dialog to access the Operations dialog.

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 IMS data store 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 SYNCELEVEL None transaction.
  - CEXIVP2 performs a Commit Mode 1 SYNCELEVEL Confirm transaction.
  - CEXIVP3 performs a Commit Mode 1 SYNCELEVEL None transaction using persistent sockets.
  - CEXIVP4 performs a Commit Mode 1 SYNCELEVEL 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 security is enabled in the IMS Connect system definition in the IMS Connect Extensions repository. See “Defining IMS Connect systems” on page 320.
  - 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 CEXRXCI1 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 data stores 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

- 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 “Activating the client services exit” on page 597 for details.
- Use IBM IMS Problem Investigator for z/OS to browse the events recorded in the IMS Connect Extensions journal. See “Reporting and analysis with IMS Problem Investigator” on page 73.
- Use IBM IMS Performance Analyzer for z/OS to analyze IMS Connect performance. See “Reporting and analysis with IMS Performance Analyzer” on page 82.
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: OTMA workloads” on page 591.

Related reference:
“REXX automation samples for the IMS Connect Extensions host command environment” on page 223

The SCEXSAMP library includes several sample REXX execs that use the IMS Connect Extensions host command environment. As well as performing useful operations tasks, you can use these samples as a guide to developing your own REXX execs that improve automation across your IMS environment.

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 ODBMROUTE 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 security is enabled in the IMS Connect system definition in the IMS Connect Extensions repository. See “Defining IMS Connect systems” on page 320.
- 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 #Request (Write Socket)
000016 #Reply: (Length: +54)
000017
000018 'Header' FDSSLen: +54 FDDMID: D0 FormatID: 41 FRQSCRR: 0001
000019 'Objects'
000020
```

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**

- Use IBM IMS Problem Investigator for z/OS to browse the events recorded in the IMS Connect Extensions journal. See "Reporting and analysis with IMS Problem Investigator" on page 73.
- Use IBM IMS Performance Analyzer for z/OS to analyze IMS Connect performance. See "Reporting and analysis with IMS Performance Analyzer" on page 82.
- 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 594.

Related reference:
The SCEXSAMP library includes several sample REXX execs that use the IMS Connect Extensions host command environment. As well as performing useful operations tasks, you can use these samples as a guide to developing your own REXX execs that improve automation across your IMS environment.

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.V3R1M0. 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.V3R1M0.SCEXEXEC(CEXOREXX) 'CEX.V3R1M0 ENU'

   If the qualifier for your IMS Connect Extensions installation data sets is not CEX.V3R1M0, 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 \texttt{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 \texttt{prefix.SCEXPENU} in the ISPPLIB concatenation.
   - Include the skeleton library \texttt{prefix.SCEXSENU} in the ISPSLIB concatenation.
   - Include the link/load module library \texttt{prefix.SCEXLINK} in the ISPLLIB concatenation.
   - Include the table library \texttt{prefix.SCEXTENU} in the ISPTLIB concatenation.
   - Include the message library \texttt{prefix.SCEXMENU} in the ISPMLIB concatenation.

3. On the TSO command processor panel, enter the following command:
   \texttt{%CEXOREXX NODYNAM lang}

4. To add IMS Connect Extensions to an ISPF menu, set \&ZSEL to:
   \texttt{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 29. 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:

\texttt{CMD(EX 'prefix.SCEXEXEC(CEXOREXX) ' prefix.lang (EXEC, LNK, MSG, PNL, TBL, SKL) ')}

IMS Connect Extensions will then use the following libraries:

- \texttt{prefix.EXEC}
  - REXX EXECs
- \texttt{prefix.LNK}
  - Executable load modules
- \texttt{prefix.MSG}
  - ISPF messages
- \texttt{prefix.PNL}
  - ISPF panels
- \texttt{prefix.TBL}
  - ISPF input tables
- \texttt{prefix.SKL}
  - ISPF skeletons
Upgrading to IMS Connect Extensions V3.1

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

Upgrading from IMS Connect Extensions V2.4

Note the following changes that may affect existing user's of IMS Connect Extensions V2.4:

Installation requirements for product library SFUNLINK

Product library SFUNLINK previously supplied with IMS Connect Extensions V2.4 is now supplied in IBM Common Services Library for z/OS (5655-CSL). Refer to the IMS Connect Extensions program directory for installation instructions.

Changes to CONNECT host command for REXX

The CONNECT host command for REXX has been modified in the following ways:

- The APPLID parameter is no longer required.
- PASSTIK=YES is now the default authenticator.

See “CONNECT command” on page 431.

Changes to TRACE host command for REXX

The TRACE host command for REXX has been modified to allow you to more easily modify an active trace by making the default values apply only when a trace is not currently active. See “TRACE command” on page 456.

Renamed sample job CEXCMDS for access control configuration

Sample job CEXCMDS that implements IMS Connect Extensions resource profiles has been renamed to CEXRACF. See “Example: typical access control configuration” on page 240.

IMS Connect Extensions Operations Console for IBM Explorer for z/OS (z/OS Explorer)

The IMS Connect Extensions Operations Console is now available for direct download via the Internet. For the most up to date information, visit the Install Eclipse Tools page on the IBM Developer website.

Installing the IMS Connect Extensions Operations Console

You can install the IMS Connect Extensions Operations Console for z/OS Explorer by downloading it from the IBM Developer website.

Before you begin

The IMS Connect Extensions Operations Console is an optional plug-in for IBM Explorer for z/OS (z/OS Explorer) that provides you with additional capabilities beyond the IMS Connect Extensions ISPF-based interface. Use the Operations Console to view real-time IMS Connect information, stop and start IMS data stores, suspend and resume routing to an IMS data store, and issue IMS Connect WTOR command or IMS Connect z/OS command and IMS type-1 commands. For more information on the capabilities of the Operations Console, see “Getting started with Operations Console” on page 175.
Procedure

To install and start the IMS Connect Extensions Operations Console, complete the following steps:

1. Download and install IBM Explorer for z/OS (z/OS Explorer) and the IMS Connect Extensions Operations Console on your PC. For the most up to date information, visit the Install Eclipse Tools page on the IBM Developer website.

2. Start the Operations Console. For more information, see “Getting started with Operations Console” on page 175.
Chapter 3. User interface options

IMS Connect Extensions provides several interface options for centralized management and control of all your IMS Connect systems.

IMS Connect Extensions ISPF dialog

The IMS Connect Extensions ISPF dialog 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.

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.

Related concepts:

- Chapter 8, “ISPF Operations dialog,” on page 119

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.

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.

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 for z/OS Explorer

IMS Connect Extensions Operations Console is a plug-in for z/OS Explorer 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.

Related concepts:
Chapter 9, “IMS Connect Extensions Operations Console for z/OS Explorer,” on page 175
IBM IMS Connect Extensions for z/OS Operations Console is a plug-in for z/OS Explorer that provides you with additional capabilities beyond the IBM IMS Connect Extensions for z/OS ISPF-based interface. Use the Operations Console to view real-time IMS Connect information, stop and start IMS data stores, suspend and resume routing, and issue IMS Connect WTOR command or IMS Connect z/OS command and IMS type-1 commands.
Eclipse environment considerations

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

The Operations Console can be configured in several ways. A typical configuration consists of the following components:

- 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 for displaying messages
- At the right: in-product help

![Figure 16. The IMS Connect Extensions Operations Console for z/OS Explorer](image)

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. IMS Connect event collection, reporting, and forwarding

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

Topics:
- Chapter 4, “Event collection,” on page 55
- Chapter 5, “Reporting IMS Connect activity,” on page 73
- Chapter 6, “Tracing message flows,” on page 93
- Chapter 7, “Forwarding a live feed of IMS Connect events,” on page 99
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. For more information on how you can use the collected event data to report on IMS Connect performance, resource usage, and transit event tracing, see “IMS Connect Extensions event collection” in the IBM IMS Performance Analyzer for z/OS Report Reference.

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'AO' - X'FF' is used to identify IMS Connect event records created by IMS Connect Extensions.

Related concepts:

- "IMS Connect Extensions 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.

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 there are outstanding event records to be archived.

The journal is switched prior to becoming full. You can force a switch manually using any of the following interfaces:

• The IMS Connect Extensions ISPF dialog using either of the following panels:
  – The Operations panel. See Using the Operations dialog on page 120.
  – The Commands panel. See Issuing IMS Connect Extensions commands from the Commands panel on page 145.

• The IMS Connect Extensions Operations Console for z/OS Explorer. See “Managing IMS Connect systems” on page 194.

• The IMS Connect Extensions host command environment for REXX. See Chapter 19, “IMS Connect Extensions host command environment for REXX,” on page 429.

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

**Note:** If all of the active data sets are full, event records might be discarded or overwritten depending on the setting of **journal full option** in the Active Journal Data Set definition. See “Defining active journal data set processing options” on page 329.

It is 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.

**Related reference:**
“Defining active journal data set processing options” on page 329

The Active Journal Data Set panel contains processing options for the active IMS Connect Extensions journal. To access this ISPF panel, select option 1.1 **System Definitions** from the IMS Connect Extensions primary menu and then select **Active Data Set**.

“Defining the archive journal data set template” on page 332

The Archive Journal Data Set Template panel contains options for the archive IMS Connect Extensions journals. To access this ISPF panel, select option 1.1 **System Definitions** from the IMS Connect Extensions primary menu and then select **Archive Data Set**.

---

**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.

**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.

**RECORDER trace event**

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

**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:**
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.

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 in IMS Connect

The following sample event flows demonstrate a typical sequence of events that can be captured in IMS Connect with IMS Connect Extensions event collection.

Sample OTMA event flow: Sync Level None

3C Prepare Read Socket  
49 Read Socket  
3D Message Exit called for READ  
3E Message Exit return for READ  
41 Message sent to OTMA  
42 Message received from OTMA  
3D Message Exit called for XMIT  
3E Message Exit return for XMIT  
4A Write Socket  
0C Begin Close Socket  
0D End Close Socket  
48 Trigger event CLOS

Figure 17. Sample event flow: Sync Level None

Sample OTMA event flow: Sync Level Confirm

3C Prepare Read Socket  
49 Read Socket  
3D Message Exit called for READ  
3E Message Exit return for READ  
41 Message sent to OTMA  
42 Message received from OTMA  
3D Message Exit called for XMIT  
3E Message Exit return for XMIT  
4A Write Socket  
49 Read Socket  
3D Message Exit called for READ  
3E Message Exit return for READ  
41 Message 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  
0C Begin Close Socket  
0D End Close Socket  
48 Trigger event CLOS

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

3C Prepare READ Socket  
49 READ Socket  
5B ORDA 1041 EXCSAT-Exchange Server Attributes  
49 READ Socket  
49 READ Socket  
5B ORDA 1060 ACSEC-Access Security  
5C ORDA 1443 EXSACRD-Server Attributes Reply Data  
4A WRITE Socket  
49 READ Socket  
49 READ Socket  
5B ORDA 106E SECHK-Security Check  
63 ODBM Security Exit called  
6A ODBM Security Exit returned  
5C ORDA 1219 SECHKRE-Security Check Reply Message  
4A WRITE Socket  
49 READ Socket  
49 READ Socket  
5B ORDA 2001 ACCDB-Access RDB  
5D ODBM begin Allocate PSB (APSB) Program=AUTPSB11  
61 ODBM 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  
5C ORDA 2201 ACCDGR-Access RDB Reply Message  
4A WRITE Socket  
4B Trigger Event for ODBMMSG  
3C Prepare READ Socket  
49 READ Socket  
5B ORDA 2000 OPQRY-Open Query  
49 READ Socket  
49 READ Socket  
5B ORDA CC03 DLIFUN-DL/I function  
49 READ Socket  
49 READ Socket  
5B ORDA CC01 INIAI-AIB data  
49 READ Socket  
49 READ Socket  
5B ORDA CC04 RTRFVL-Field client wants to retrieve data  
49 READ Socket  
49 READ Socket  
5B ORDA CC06 SSLIST-List of segment search argument  
AA ODBM Trace: Message sent to ODBM  
69 Message sent to ODBM  
AA ODBM Trace: Message received from ODBM  
6A Message received from ODBM  
SC ORDA 2205 OPQRYRM-Open Query Complete  
4A WRITE Socket  
4B Trigger Event for ODBMMSG  
...  
3C Prepare READ Socket  
49 READ Socket  
5B ORDA CB01 DEALLOCDB-Deallocation PSB  
5F ODBM begin Deallocation 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 Deallocation PSB (DPSB)  
5C ORDA CA01 DEALLOCDBRM-Name of deallocated PSB  
4A WRITE Socket  
4B Trigger Event for ODBMMSG  
3C Prepare READ Socket  
47 Session Error  
0C Begin CLOSE Socket  
0D End CLOSE Socket  

Figure 19. Sample ODBM event flow
Sample MSC event flow

7D ICON to ICON start of session
7B MSC message received from MSC Msgtype=REQUEST <= Notification of transaction coming 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=REQUEST <= Transaction passed to remote IMS Connect via TCP/IP

Figure 20. Sample MSC event flow: local IMS Connect system connected by a MSC TCP/IP-type link

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 IMS Connect via TCP/IP
7E ICON to ICON end of session

Figure 21. Sample MSC event flow: remote IMS Connect system connected by a MSC TCP/IP-type link

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 local IMS
7B MSC message sent to MSC Msgtype=REQRESP
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

Related concepts:
“Browse events recorded in the IMS Connect Extensions journal” on page 73
Use IMS Problem Investigator to browse and analyze the events in an IMS Connect Extensions archive journal data set.
Chapter 24, “Events recorded by IMS Connect Extensions,” on page 551
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.
Configuring event collection and journals

Before you can collect and report on IMS Connect events, you must configure your systems.

To configure your systems for event collection, complete the following steps:

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
<th>Reference and instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Determine a journal archiving scheme that is appropriate for your environment and gather required information.</td>
<td>Collection level&lt;br&gt;&quot;Estimating the size of active journals&quot;&lt;br&gt;&quot;Defining active journal data set processing options&quot; on page 329&lt;br&gt;&quot;Naming archive journal data sets&quot; on page 62</td>
</tr>
<tr>
<td></td>
<td>• Event collection level&lt;br&gt;• Number of active journals in the pool and their size&lt;br&gt;• A base name for active journal data sets&lt;br&gt;• When to take an archive copy of the active journal and how long to retain it&lt;br&gt;• A base name for archive journal data sets</td>
<td>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.</td>
</tr>
<tr>
<td>2</td>
<td>Create an archive JCL skeleton based on one of the samples.</td>
<td>&quot;Creating an archive JCL skeleton&quot; on page 63</td>
</tr>
<tr>
<td>3</td>
<td>In the system definition:</td>
<td>&quot;Defining IMS Connect systems&quot; on page 320</td>
</tr>
<tr>
<td></td>
<td>• <strong>Activate Event Collection</strong> and set the collection level.&lt;br&gt;• Enter / (slash) next to <strong>Active Data Set.</strong></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>In the Active Journal Data Set panel:</td>
<td>&quot;Defining active journal data set processing options&quot; on page 329</td>
</tr>
<tr>
<td></td>
<td>• Specify the base name and attributes of Active Journal data sets.&lt;br&gt;• Select <strong>Archive manager</strong> to activate journal archiving.&lt;br&gt;• Specify the name of the archive JCL skeleton member.</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>In the Archive Journal Data Set Template panel:</td>
<td>&quot;Defining the archive journal data set template&quot; on page 332</td>
</tr>
<tr>
<td></td>
<td>• Specify a pattern for naming Archive Journal data sets and other Archive Journal properties.</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>To implement the new event collection and journal settings, either refresh the system definition or restart the IMS Connect system.</td>
<td>&quot;Refreshing in-memory definitions from the IMS Connect Extensions repository&quot; on page 150</td>
</tr>
</tbody>
</table>

**Estimating the size of active journals**

This topic provides some assistance when determining IMS Connect Extensions 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)**

<table>
<thead>
<tr>
<th>Collection level</th>
<th>Tracks/day</th>
<th>Cyls/day</th>
<th>MB/day</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>&lt; 1</td>
<td>&lt; 1</td>
</tr>
<tr>
<td>1</td>
<td>5184</td>
<td>346</td>
<td>277</td>
</tr>
<tr>
<td>2</td>
<td>13738</td>
<td>916</td>
<td>734</td>
</tr>
<tr>
<td>3</td>
<td>22291</td>
<td>1486</td>
<td>1190</td>
</tr>
<tr>
<td>4</td>
<td>24106</td>
<td>1607</td>
<td>1287</td>
</tr>
</tbody>
</table>

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 &DATE

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Result format</th>
<th>Example result</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;DATE</td>
<td>The date the archive journal was created.</td>
<td>DYYMMDD</td>
<td>D160115</td>
</tr>
<tr>
<td>&amp;FIRST</td>
<td>The date and time of the earliest event in the archive journal.</td>
<td>DYYMMDD.THHMMSS</td>
<td>D160115.T153025</td>
</tr>
<tr>
<td>&amp;ID</td>
<td>The HWS (system) ID of the IMS Connect system.</td>
<td>1 - 8 characters</td>
<td>HWS1</td>
</tr>
<tr>
<td>&amp;JDATE</td>
<td>The Julian date when the archive journal was created.</td>
<td>JYYDDD</td>
<td>J160105</td>
</tr>
<tr>
<td>&amp;JFIRST</td>
<td>The date and time of the earliest event in the archive journal. The date is in Julian format.</td>
<td>JYYDDD.THHMMSS</td>
<td>J160105.T153025</td>
</tr>
<tr>
<td>&amp;TIME</td>
<td>The time the archive journal was created.</td>
<td>THHMMSS</td>
<td>T153025</td>
</tr>
</tbody>
</table>

**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 IMS Connect Extensions journals

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

Related reference:

“Archive Manager utility” on page 587

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.

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 via the following interfaces:

• The IMS Connect Extensions ISPF dialog. See "Using the Operations dialog” on page 120.
• The IMS Connect Extensions Operations Console for z/OS Explorer. See “Managing IMS Connect systems” on page 194.
• The IMS Connect Extensions host command environment for REXX. See “SWITCH command” on page 456.

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 panel. 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.

```
//STEP01 EXEC PGM=FUNEXEC,PARM=CEXJARC,REGION=0M
//STEPLIB DD DISP=SHR,DSN=funpre.SFUNLINK
// DD DISP=SHR,DSN=cexpre.CSEXLINK
```

**Figure 23. How the IMS Connect Extensions Archive Manager uses the archive skeleton JCL**

**Related reference:**
"Defining active journal data set processing options" on page 329

The Active Journal Data Set panel contains processing options for the active IMS Connect Extensions journal. To access this ISPF panel, select option 1.1 **System Definitions** from the IMS Connect Extensions primary menu and then select **Active Data Set**.

**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.
Enter the high-level qualifier of the IMS Connect Extensions Common Services Library (SFUNLINK) and program (SCEXLINK) link libraries.

The Archive Manager uses the archive data set name (see “Defining the archive journal data set template” on page 332) 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.

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.

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 68) suffixed by ”.P01”. For example, if the active journal panel displays “CEX.ACTIVE.HWS1” use: CEX.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
//STEP01 EXEC PGM=CEXARCH, PARM=CEXJARC, REGION=0M
//REPOSTRY DD DSN=CEX.REPOSTRY.MM001 3

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.
Enter the high-level qualifier of the IBM Common Services Library for z/OS (SFUNLINK) and IMS Connect Extensions (SCEXLINK) link libraries.

The Archive Manager uses the archive data set name (see “Defining the archive journal data set template” on page 332) to resolve %ARCDSN. Use the remaining ARCOU 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 `ARCOU DD` for details.

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

```
//ABC JOB (ACCOUNT), 'NAME'
//STEP01 EXEC PGM=FUNEXEC,PARM=CEXJARC,REGION=0M
//STEP01 DD DISP=SHR,DSN=funpre.SFUNLINK
//STEP01 DD DISP=SHR,DSN=cexpre.SCEXLINK
//ARCOUT DD DSN=%ARCDSN, 
//ARCOUT DD DSN=USER.D060206.T150645, 
//ARCOUT DD DSN=USER.D060206.T150645, 
//REPOSTRY DD DISP=SHR,DSN=CEX.REPOSTRY.MM001
//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 68.
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.
Archiving must be the first job step. The job fails otherwise.

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.

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:

```
//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,
// DISP=(,CATLG,DELETE),
// UNIT=(TAPE,,DEFER),
// VOL=(,RETAIN)
//*
//****************************************************************************
//** 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 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'
//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,
// DISP=(,CATLG,DELETE),
// UNIT=(TAPE,,DEFER),
// VOL=(,RETAIN)
//REPOSTRY DD DISP=SHR,DSN=CEx.REPOSTRY.MM001
//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” on page 68.

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

Use the Active Journal Data Set panel to set active journal options and the archive Job Statement Information and JCL skeleton.

**Procedure**

1. From the IMS Connect Extensions ISPF primary option menu, create or edit a system definition using option 1.1 **System Definitions**
2. On the System Definition panel, 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**

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.

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.

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.

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.

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.

Related reference:
“Defining active journal data set processing options” on page 329

The Active Journal Data Set panel contains processing options for the active IMS Connect Extensions journal. To access this ISPF panel, select option 1.1 System Definitions from the IMS Connect Extensions primary menu and then select Active Data Set.

Creating an archive journal template

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

Procedure

1. Using the IMS Connect Extensions ISPF dialog, edit the system definition with the active journal settings you created in “Configuring active journals” on page 68.

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.

3. Set the name template of the archive data set in the Data Set name field (▌). 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 ARCOUNT DD statement of the archive skeleton JCL.

4. Use the Maximum active, Maximum time, 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.
Results

Attention: The V1.1 skeleton fields are only used with archive JCL skeletons from IMS Connect Extensions V1.1. If you have such skeletons, upgrade them rather than using these fields.

Related reference: “Defining the archive journal data set template” on page 332

The Archiv Journal Data Set Template panel contains options for the archive IMS Connect Extensions journals. To access this ISPF panel, select option 1.1 **System Definitions** from the IMS Connect Extensions primary menu and then select Archive Data Set.

### 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

<table>
<thead>
<tr>
<th>File</th>
<th>Menu</th>
<th>Settings</th>
<th>Help</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>EDIT Archive Journal Data Set Template Command ==&gt;</td>
<td>Data Set name . 'USER.&amp;DATE..&amp;TIME'</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Maximum active . 3</td>
<td>(Number of Active data sets in Archive) 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Maximum time . ___</td>
<td>(Seconds to wait before closing Archive)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Maximum size . ___</td>
<td>(Size of Archive Journal - MB)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Maximum volumes . 1</td>
<td>(Number of volumes per Archive)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Archive cleanup retention period . ___</td>
<td>(Number of days)</td>
</tr>
</tbody>
</table>

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 25. The IMS Connect Extensions Archive Journal Data Set Template panel
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 return code of zero (0) 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 to change its active journal settings.

5. Enter / (slash) next to the **Active Data Set** field.

6. In the Active Journal Data Set panel, complete the following steps:
   
   a. 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.
   
   b. Edit the active journal definitions to include the new sizing parameters.
   
   c. Press the Exit function key (F3) to save and exit.

7. Restart the IMS Connect system.
Chapter 5. Reporting IMS Connect activity

Use IBM IMS Performance Analyzer for z/OS and IBM Problem Investigator for z/OS to report on events collected by IMS Connect Extensions.

Related concepts:
“Related IBM tools” on page 20
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.

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.

Note: You can also perform many of these tasks using IBM Transaction Analysis Workbench for z/OS. For more information, see the IBM Transaction Analysis for z/OS User's Guide.

Browse events recorded in the IMS Connect Extensions journal

Use IMS Problem Investigator to browse and analyze the events in an IMS Connect Extensions archive journal data set.

When you select log files to analyze in IMS™ Problem Investigator, the event records are listed in time sequence with their code and a summary description. For a list of events recorded by IMS Connect Extensions, see Chapter 24, “Events recorded by IMS Connect Extensions,” on page 551. To isolate a specific socket session, use the tracking feature:
Scroll through the different views to see additional detail for each event. In the example below, log record “3E” reveals more about this session:

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
<th>Date</th>
<th>Time (Local)</th>
</tr>
</thead>
<tbody>
<tr>
<td>003C</td>
<td>Prepare READ Socket</td>
<td>2018-07-30</td>
<td>13:48.03.428438</td>
</tr>
<tr>
<td>0049</td>
<td>READ Socket</td>
<td></td>
<td>13:48.03.428463</td>
</tr>
<tr>
<td>003D</td>
<td>Message Exit HWSSMPL1 called for READ</td>
<td></td>
<td>13:48.03.428472</td>
</tr>
<tr>
<td>004A</td>
<td>Event Collection IRM Trace</td>
<td></td>
<td>13:48.03.428468</td>
</tr>
<tr>
<td>003A</td>
<td>Event Collection OTMA Trace</td>
<td></td>
<td>13:48.03.428502</td>
</tr>
<tr>
<td>003E</td>
<td>Message Exit HWSSMPL1 returned from READ</td>
<td></td>
<td>13:48.03.428502</td>
</tr>
<tr>
<td>0041</td>
<td>Message sent to OTMA</td>
<td></td>
<td>13:48.03.428570</td>
</tr>
<tr>
<td>0042</td>
<td>Message received from OTMA</td>
<td></td>
<td>13:48.03.452477</td>
</tr>
<tr>
<td>003A</td>
<td>Event Collection OTMA Trace</td>
<td></td>
<td>13:48.03.452537</td>
</tr>
<tr>
<td>003D</td>
<td>Message Exit HWSSMPL1 called for XMIT</td>
<td></td>
<td>13:48.03.452541</td>
</tr>
<tr>
<td>004A</td>
<td>Event Recording EXIT Output Message Trace</td>
<td></td>
<td>13:48.03.452551</td>
</tr>
<tr>
<td>003E</td>
<td>Message Exit HWSSMPL1 returned from XMIT</td>
<td></td>
<td>13:48.03.452554</td>
</tr>
<tr>
<td>004A</td>
<td>WRITE Socket</td>
<td></td>
<td>13:48.03.452585</td>
</tr>
<tr>
<td>0049</td>
<td>READ Socket</td>
<td></td>
<td>13:48.03.457661</td>
</tr>
<tr>
<td>0049</td>
<td>READ Socket</td>
<td></td>
<td>13:48.03.457687</td>
</tr>
<tr>
<td>003D</td>
<td>Message Exit HWSSMPL1 called for READ</td>
<td></td>
<td>13:48.03.457699</td>
</tr>
<tr>
<td>004A</td>
<td>Event Collection IRM Trace</td>
<td></td>
<td>13:48.03.457699</td>
</tr>
<tr>
<td>003A</td>
<td>Event Collection OTMA Trace</td>
<td></td>
<td>13:48.03.457717</td>
</tr>
<tr>
<td>003E</td>
<td>Message Exit HWSSMPL1 returned from READ</td>
<td></td>
<td>13:48.03.457721</td>
</tr>
<tr>
<td>0041</td>
<td>Message sent to OTMA</td>
<td></td>
<td>13:48.03.457781</td>
</tr>
<tr>
<td>0045</td>
<td>OTMA Time-out</td>
<td></td>
<td>13:48.03.709561</td>
</tr>
<tr>
<td>003D</td>
<td>Message Exit HWSSMPL1 called for XMIT</td>
<td></td>
<td>13:48.03.709574</td>
</tr>
<tr>
<td>004A</td>
<td>Event Recording EXIT Output Message Trace</td>
<td></td>
<td>13:48.03.709586</td>
</tr>
<tr>
<td>003E</td>
<td>Message Exit HWSSMPL1 returned from XMIT</td>
<td></td>
<td>13:48.03.709587</td>
</tr>
<tr>
<td>004A</td>
<td>WRITE Socket</td>
<td></td>
<td>13:48.03.709826</td>
</tr>
<tr>
<td>000C</td>
<td>Begin CLOSE Socket</td>
<td></td>
<td>13:48.03.709844</td>
</tr>
<tr>
<td>000D</td>
<td>End CLOSE Socket</td>
<td></td>
<td>13:48.03.710010</td>
</tr>
<tr>
<td>004A</td>
<td>Trigger Event for CLOSE</td>
<td></td>
<td>13:48.03.715305</td>
</tr>
</tbody>
</table>

**Figure 26. IMS Problem Investigator: Sample OTMA event flow recorded in an IMS Connect Extensions archive journal data set**
To view the complete log record, select the record you wish to view. In the example below, you can see the details of the IMS request message (IRM):

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
<th>Date</th>
<th>Time (Local)</th>
</tr>
</thead>
<tbody>
<tr>
<td>003C</td>
<td>Prepare READ Socket</td>
<td>2018-07-30</td>
<td>13.48.03.428438</td>
</tr>
<tr>
<td>0049</td>
<td>READ Socket</td>
<td>2018-07-30</td>
<td>13.48.03.428463</td>
</tr>
<tr>
<td>003D</td>
<td>Message Exit HWSSMPL1 called for READ</td>
<td>2018-07-30</td>
<td>13.48.03.428472</td>
</tr>
<tr>
<td>004A</td>
<td>Event Collection IRM Trace</td>
<td>2018-07-30</td>
<td>13.48.03.428468</td>
</tr>
<tr>
<td>00A3</td>
<td>Event Collection OTMA Trace</td>
<td>2018-07-30</td>
<td>13.48.03.428502</td>
</tr>
<tr>
<td>003E</td>
<td>Message Exit HWSSMPL1 returned from READ</td>
<td>2018-07-30</td>
<td>13.48.03.428502</td>
</tr>
<tr>
<td>0041</td>
<td>Message sent to OTMA</td>
<td>2018-07-30</td>
<td>13.48.03.428570</td>
</tr>
<tr>
<td>0042</td>
<td>Message received from OTMA</td>
<td>2018-07-30</td>
<td>13.48.03.452477</td>
</tr>
<tr>
<td>00A3</td>
<td>Event Collection OTMA Trace</td>
<td>2018-07-30</td>
<td>13.48.03.452537</td>
</tr>
<tr>
<td>003D</td>
<td>Message Exit HWSSMPL1 called for XMIT</td>
<td>2018-07-30</td>
<td>13.48.03.452541</td>
</tr>
<tr>
<td>00A6</td>
<td>Event Recording EXIT Output Message Trace</td>
<td>2018-07-30</td>
<td>13.48.03.452551</td>
</tr>
<tr>
<td>003E</td>
<td>Message Exit HWSSMPL1 returned from XMIT</td>
<td>2018-07-30</td>
<td>13.48.03.452554</td>
</tr>
<tr>
<td>004A</td>
<td>WRITE Socket</td>
<td>2018-07-30</td>
<td>13.48.03.452585</td>
</tr>
<tr>
<td>0049</td>
<td>READ Socket</td>
<td>2018-07-30</td>
<td>13.48.03.457661</td>
</tr>
</tbody>
</table>

Figure 27. IMS Problem Investigator: Sample OTMA event flow recorded in an IMS Connect Extensions archive journal data set (with brief descriptions)
You can also move backwards and forwards through the journal from the detailed view. In the example below, we can see the effects of OTMA routing via the DSORIG and DSTAR fields:

**Note:** For more information on OTMA routing in IMS Connect Extensions, see Chapter 12, “OTMA workload routing in IMS Connect,” on page 245.
<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Record</td>
<td>00063775</td>
<td>Line 00000000</td>
</tr>
<tr>
<td>Line</td>
<td>00000000</td>
<td>Command ===＞ Scroll ===＞ PAGE Form ===＞ + Use Form in Filter Format ===＞ STD</td>
</tr>
<tr>
<td>Date</td>
<td>2018-07-30 Monday</td>
<td>Time: 13:48.03.426502.988</td>
</tr>
<tr>
<td>Code</td>
<td>003E</td>
<td>Message Exit HWSSMPL1 returned from READ</td>
</tr>
<tr>
<td>LSN</td>
<td>00000000043E43</td>
<td></td>
</tr>
<tr>
<td>Date</td>
<td>2018-07-30 Monday</td>
<td>Time: 13:48.03.426502.988</td>
</tr>
<tr>
<td>Code</td>
<td>003E</td>
<td>Message Exit HWSSMPL1 returned from READ</td>
</tr>
<tr>
<td>LSN</td>
<td>00000000043E43</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 29. Viewing additional detail for log record 3E in IMS Problem Investigator**

Here we can see the contents of IMS Connect Extensions trace record “A6”:

Chapter 5. Reporting IMS Connect activity 77
The following sample event flows demonstrate a typical sequence of events that can be captured in IMS Connect with IMS Connect Extensions event collection.

**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.
Using IMS Problem Investigator in an 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.

**Note:** Records relating to READ Socket and WRITE Socket events have been omitted to provide a simpler example.
Here we have the details of a selected DRDA response. You can see structure of the reply, including all of its objects (parameters).
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.

**IMS Connect transaction transit reports produced by IMS Performance Analyzer**

IMS Connect transaction transit reports produced by IMS Performance Analyzer 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, Acknowledgment 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.

**IMS Connect Transit analysis report**

The IMS Connect 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).

**Figure 35. IMS Performance Analyzer: IMS Connect Transit Analysis report (BY (TRANCODE, TIME), PEAK(80))**

**IMS Connect Transit Log report**

The IMS Connect 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 Connect transit extract produced by IMS Performance Analyzer

IMS Connect transit extracts produced by IMS Performance Analyzer gather 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.

IMS Connect resource usage reports produced by IMS Performance Analyzer

IMS Connect resource usage reports produced by IMS Performance Analyzer contain detailed and summary information on the use and availability of IMS Connect resources, including TCP/IP ports and transaction pipes (tpipes).

IMS Connect Port usage report

The IMS Connect 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 4.4
IMS Connect Resume Tpipe report

The IMS Connect Resume Tpipe report provides a summary of resume transaction pipe (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.

<table>
<thead>
<tr>
<th>Tpipe</th>
<th>Count</th>
<th>NResp</th>
<th>Fail</th>
<th>Avg</th>
<th>Max</th>
<th>Timeout</th>
<th>Avg</th>
<th>Max</th>
<th>Timeout</th>
<th>Count</th>
<th>NResp</th>
<th>Fail</th>
<th>Avg</th>
<th>Max</th>
<th>Timeout</th>
<th>Count</th>
<th>NResp</th>
<th>Fail</th>
<th>Avg</th>
<th>Max</th>
<th>Timeout</th>
</tr>
</thead>
<tbody>
<tr>
<td>CEX30001</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>0.25</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>CEX40001</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>0.25</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>CEX40002</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>0.25</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>TXRBS001</td>
<td>17</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.45</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.25</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.25</td>
</tr>
<tr>
<td>TXRBS002</td>
<td>2</td>
<td>0</td>
<td>4</td>
<td>7</td>
<td>0.25</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>0.25</td>
<td>3</td>
<td>1</td>
<td>0.83</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>TXRBS002</td>
<td>2</td>
<td>0</td>
<td>3</td>
<td>5</td>
<td>0.25</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>2.5</td>
<td>5</td>
<td>0.25</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0.60</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 37. IMS Performance Analyzer: IMS Connect Port Usage report

Figure 38. IMS Performance Analyzer: IMS Connect Resume Tpipe report (standard format)

Figure 39. IMS Performance Analyzer: IMS Connect Resume Tpipe report (extended format)
IMS Connect ACK/NAK report

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

<table>
<thead>
<tr>
<th>Transact Code</th>
<th>Target Datastor</th>
<th>Count</th>
<th>ACK</th>
<th>Count</th>
<th>Average Elaps</th>
<th>Count</th>
<th>Average Elaps</th>
<th>Sense Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>*CMD_RHQ IMD3</td>
<td>16</td>
<td>16</td>
<td>1.890</td>
<td>0</td>
<td>0.000</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>*CMD_RHQ IMD4</td>
<td>22</td>
<td>22</td>
<td>0.419</td>
<td>0</td>
<td>0.000</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>*CMD_RTP IMD3</td>
<td>16</td>
<td>16</td>
<td>0.336</td>
<td>0</td>
<td>0.000</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>*CMD_RTP IMD4</td>
<td>18</td>
<td>17</td>
<td>0.367</td>
<td>1</td>
<td>7.368</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DSPALLI IMD3</td>
<td>4</td>
<td>3</td>
<td>270.434</td>
<td>1</td>
<td>275.011</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DSPALLI IMD4</td>
<td>1</td>
<td>0</td>
<td>0.000</td>
<td>1</td>
<td>263.518</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DYPTRANS IMD3</td>
<td>13</td>
<td>12</td>
<td>1.111.570</td>
<td>1</td>
<td>600.739</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DYPTRANS IMD4</td>
<td>2</td>
<td>1</td>
<td>379.666</td>
<td>0</td>
<td>0.000</td>
<td>1</td>
<td>24 Previous conversation still in progress</td>
<td></td>
</tr>
<tr>
<td>IYTCV IMD3</td>
<td>79</td>
<td>74</td>
<td>600.214</td>
<td>1</td>
<td>594.968</td>
<td>4</td>
<td>24 Previous conversation still in progress</td>
<td></td>
</tr>
<tr>
<td>IYTCV IMD4</td>
<td>79</td>
<td>77</td>
<td>2.157.328</td>
<td>0</td>
<td>0.000</td>
<td>2</td>
<td>24 Previous conversation still in progress</td>
<td></td>
</tr>
<tr>
<td>IVTNO IMD3</td>
<td>46</td>
<td>24</td>
<td>322.746</td>
<td>22</td>
<td>234.673</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IVTNO IMD4</td>
<td>46</td>
<td>27</td>
<td>291.475</td>
<td>19</td>
<td>292.862</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PART IMD3</td>
<td>2</td>
<td>2</td>
<td>270.814</td>
<td>0</td>
<td>0.000</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PART IMD4</td>
<td>4</td>
<td>4</td>
<td>253.487</td>
<td>0</td>
<td>0.000</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 40. IMS Performance Analyzer: IMS Connect ACK/NAK report

IMS Connect Exception reports

The IMS Connect Exception reports 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 Connect Gap Analysis report

The IMS Connect 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 Connect Exception List report

**Figure 41. IMS Performance Analyzer: IMS Connect Exception List report**

<table>
<thead>
<tr>
<th>Event Time</th>
<th>System ID</th>
<th>Description</th>
<th>Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>13.16.30.010476</td>
<td>HMSVP3</td>
<td>41 Client send NAK to OTMA</td>
<td>Key=BE5D1498C4638F01 DS=XCXMMD3 Tpipe=TRRSSD01</td>
</tr>
<tr>
<td>13.16.30.011067</td>
<td>HMSVP3</td>
<td>47 Session error</td>
<td>Key=CS5D5E3404040404 Type=LATEMSG</td>
</tr>
<tr>
<td>13.16.30.011078</td>
<td>HMSVP3</td>
<td>41 Client send NAK to OTMA</td>
<td>Key=CS5D5E3404040404 Type=LATEMSG</td>
</tr>
<tr>
<td>13.16.30.261476</td>
<td>HMSVP3</td>
<td>45 OTMA time-out</td>
<td>Key=BE5D1498C4638F01 TOV=19</td>
</tr>
<tr>
<td>13.16.30.371599</td>
<td>HMSVP3</td>
<td>41 Client send NAK to OTMA</td>
<td>Key=BE5D1498C4638F01 TOV=19</td>
</tr>
<tr>
<td>13.16.39.623239</td>
<td>HMSVP3</td>
<td>45 OTMA time-out</td>
<td>Key=BE5D140A8638723 TOV=19</td>
</tr>
<tr>
<td>13.17.13.363930</td>
<td>HMSVP3</td>
<td>45 OTMA time-out</td>
<td>Key=BE5D140A8638723 TOV=19</td>
</tr>
<tr>
<td>13.17.37.141184</td>
<td>HMSVP3</td>
<td>41 Client send NAK to OTMA</td>
<td>Key=BE5D140A8638723 TOV=19</td>
</tr>
<tr>
<td>13.17.37.392280</td>
<td>HMSVP3</td>
<td>45 OTMA time-out</td>
<td>Key=BE5D140A8638723 TOV=19</td>
</tr>
<tr>
<td>13.17.43.453120</td>
<td>HMSVP3</td>
<td>41 Client send NAK to OTMA</td>
<td>Key=BE5D140A8638723 TOV=19</td>
</tr>
<tr>
<td>13.17.43.703713</td>
<td>HMSVP3</td>
<td>45 OTMA time-out</td>
<td>Key=BE5D140A8638723 TOV=19</td>
</tr>
<tr>
<td>13.18.23.895207</td>
<td>HMSVP3</td>
<td>41 Client send NAK to OTMA</td>
<td>Key=BE5D140A8638723 TOV=19</td>
</tr>
<tr>
<td>13.18.24.145457</td>
<td>HMSVP3</td>
<td>45 OTMA time-out</td>
<td>Key=BE5D140A8638723 TOV=19</td>
</tr>
<tr>
<td>13.18.31.817110</td>
<td>HMSVP3</td>
<td>41 Client send NAK to OTMA</td>
<td>Key=BE5D140A8638723 TOV=19</td>
</tr>
<tr>
<td>13.18.32.067250</td>
<td>HMSVP3</td>
<td>45 OTMA time-out</td>
<td>Key=BE5D140A8638723 TOV=19</td>
</tr>
<tr>
<td>13.18.44.960481</td>
<td>HMSVP3</td>
<td>47 Session error</td>
<td>Key=BE5D140A8638723 TOV=19</td>
</tr>
<tr>
<td>13.19.01.567803</td>
<td>HMSVP3</td>
<td>45 OTMA time-out</td>
<td>Key=BE5D140A8638723 TOV=19</td>
</tr>
<tr>
<td>13.21.06.826389</td>
<td>HMSVP3</td>
<td>41 Client send NAK to OTMA</td>
<td>Key=BE5D140A8638723 TOV=19</td>
</tr>
<tr>
<td>13.21.06.827276</td>
<td>HMSVP3</td>
<td>47 Session error</td>
<td>Key=BE5D140A8638723 TOV=19</td>
</tr>
<tr>
<td>13.21.07.076309</td>
<td>HMSVP3</td>
<td>45 OTMA time-out</td>
<td>Key=BE5D140A8638723 TOV=19</td>
</tr>
<tr>
<td>13.21.15.394403</td>
<td>HMSVP3</td>
<td>41 Client send NAK to OTMA</td>
<td>Key=BE5D140A8638723 TOV=19</td>
</tr>
<tr>
<td>13.21.15.645619</td>
<td>HMSVP3</td>
<td>45 OTMA time-out</td>
<td>Key=BE5D140A8638723 TOV=19</td>
</tr>
<tr>
<td>13.21.32.882603</td>
<td>HMSVP3</td>
<td>45 OTMA time-out</td>
<td>Key=BE5D140A8638723 TOV=19</td>
</tr>
<tr>
<td>13.21.37.640221</td>
<td>HMSVP3</td>
<td>45 OTMA time-out</td>
<td>Key=BE5D140A8638723 TOV=19</td>
</tr>
</tbody>
</table>

---

This report contains key information for analyzing IMS Connect issues and understanding system behavior during specific events.
Figure 42. IMS Performance Analyzer: IMS Connect Gap Analysis report

IMS Connect trace reports produced by IMS Performance Analyzer

IMS Connect trace reports produced by IMS Performance Analyzer provide 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.
IBM IMS Connect Extensions for z/OS: User's Guide

**Figure 43. IMS Performance Analyzer: IMS Connect trace report: Conversational transaction with Sync Level=CONFIRM**

**Reporting on OTMA workload routing in IMS Connect using IMS Performance Analyzer**

Use IBM IMS Performance Analyzer for z/OS to generate an IMS Connect form-based transaction transit summary report from the IMS Connect Extensions journal that displays the results of OTMA workload routing performed by IMS Connect Extensions OTMA rules-based routing.

To do this, create a report form that contains the ORIGDS (Original IMS data store referenced in the IRM.IMSSDestID field of the IRM) and TARGDS (Target IMS data store that actually processed the request) form fields. In the following example, 20 messages received by IMS Connect with an original IMS destination ID of EX02 have been distributed across the two data stores in the routing list, DS01 and DS02.
For more information on generating form-based reports in IMS Performance Analyzer, see *Creating List and Summary Report Forms* in the *Guided Tour* section of the IBM IMS Performance Analyzer for z/OS User’s Guide.

**Related reference:**

“Reviewing the active OTMA routing rules in IMS Connect” on page 262

When a system starts that has OTMA routing rules activated, a series of messages is written to CEXPRINT. These messages display the OTMA routing plan that is currently in effect (if any) and provides summarized information about the OTMA routing rules, OTMA routing lists, and IMS data stores that may be in use.
Chapter 6. Tracing message flows

You can use IMS Connect Extensions to capture additional tracing for Open Transaction Manager Access (OTMA) and Open Database Manager (ODBM) workloads by activating the IMS Connect Extensions trace. You can also use IMS Connect Extensions to control the IMS Connect Recorder Trace facility.

Using the IMS Connect Extensions trace

Tracing events are additional event records produced by IMS Connect Extensions when tracing is activated. Use the IMS Connect Extensions ISPF dialog to start and stop tracing for an active IMS Connect system.

Before you begin

You can also use the following alternate methods to start and stop tracing:

- Use the Status Monitor in IMS Connect Extensions Operations Console for z/OS Explorer. For more information, see “Using conditional trace” on page 200.
- Use the “QUERY commands” on page 436 and the “TRACE command” on page 456 in the IMS Connect Extensions host command environment for REXX to query, start, and stop tracing. See members CEXRXC11 and CEXRXC45 in the SCEXSAMP library for details.

About this task

In addition to logging IMS Connect events, IMS Connect Extensions can optionally write trace event records that expose the raw messages entering and exiting IMS Connect. These messages include the IMS request message (or IRM) from the client, the response to the client, and the OTMA messages that IMS Connect exchanges with IMS. These trace event records are useful for diagnosing problems such as malformed messages. Optionally, IMS Connect Extensions can include application data in these trace event records, such as the application data that typically follows an IMS request message.

For a list of the IMS Connect trace records, see “IMS Connect Extensions trace records” on page 557. The current tracing status and tracing level are included in the X’00’ event record.

Procedure

1. From the IMS Connect Extensions primary option menu, select option 2 Operations.
2. Set View to option 2 Systems.
3. To set tracing options for an IMS Connect system, enter line action TR next to the desired system.
The Set Tracing dialog is displayed. The **Activate Tracing** option displays the current tracing status.

Starting the trace:

4. To **start tracing**, complete the following steps:
   a. Enter `/` next to the **Activate Tracing** option.
   b. Set the following conditional tracing options as desired:

   **Reactivate tracing after system restart**
   Select this option to make tracing persist across system restarts.

   **Note:** You can only make a trace persistent at the same time as you activate it.

   **Tracing Level**
   Set the level of tracing for IMS Connect Extensions:

   1. IRM, CSM, RSM, OTMA, XML, and DRDA structures.
   2. IRM, CSM, RSM, OTMA, XML, DRDA structures, and client application data (useful for OTMA workloads).
Port  Apply tracing only to a specific port. Only activities directed at that port are traced. Use an asterisk (*) to trace on all ports.

Conditional trace resources (for OTMA workloads)
IMS Connect Extensions will only record messages that match the specified tracing condition. Specify an optional condition (one only) using the following fields:

Client Name
The name of the client.

Transaction
The transaction.

Message Exit
The message exit.

User ID
The user ID.

LTERM
The logical terminal.

IP Address
The host IP address. If you enter a DNS name it will be sent to the server to be resolved to the IP address.

c. Press Enter to generate the necessary commands. The Command Processor panel is displayed.

d. Type / (slash) next to the listed commands and then press Enter. The Response column shows all zeros (0000), indicating that the commands were successful.

e. To return to the previous screen, press the Exit function key (F3). A slash (/) next to the Activate Tracing option in the Set Tracing panel indicates that tracing is now active.

Tip: You can also review the status of your request by entering line action L next to the IMS Connect system on the Operations dialog.

Stopping the trace:
5. To stop tracing, complete the following steps:
   a. Deselect the Activate Tracing option by removing the slash (/).
   b. Press Enter to generate the necessary command. The Command Processor panel is displayed.
c. Type `/` (slash) next to the listed command and then press Enter. The Response column shows all zeros (0000), indicating that the command was successful.

d. To return to the previous screen, press the Exit function key (F3). A blank next to the Activate Tracing option in the Set Tracing panel indicates that tracing has stopped.

**Related concepts:**

Chapter 24, “Events recorded by IMS Connect Extensions,” on page 551

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.

**Related tasks:**

“Using conditional trace” on page 200

Use the Start Conditional Trace and Stop Conditional Trace context menu options in the Status Monitor of the Operations Console to start and stop tracing of an IMS Connect system based on filtering criteria.

**Related reference:**

“TRACE command” on page 456

The `TRACE` host command for REXX controls the IMS Connect Extensions trace. You can specify filter conditions to control which additional event records are written to the active IMS Connect Extensions journal data set.

---

**Controlling the IMS Connect Recorder Trace facility**

Use IMS Connect Extensions ISPF dialog to start and stop the IMS Connect Recorder Trace facility.

**Before you begin**

You can also use the following alternate methods to start and stop the Recorder Trace facility:

- Use the Status Monitor in IMS Connect Extensions Operations Console. For more information, see “Using Recorder Trace” on page 202.
- Submitting commands via the IMS Connect Extensions command shell.

**About this task**

**Note:** You cannot query the current status of the IMS Connect Recorder Trace facility through IMS Connect Extensions but you can use line action `L` in the Operations dialog to view a history of start and stop requests.

**Procedure**

1. From the IMS Connect Extensions primary option menu, select option 2 Operations.
2. Set View to option 2 Systems.
3. To activate (start) the IMS Connect Recorder Trace facility for an IMS Connect system, enter line action `RS` next to the desired system.
4. Review the status of your request by entering line action `L` next to the same system. For example:

   07.24.51 CEX5050I USR action: HWS Command start recorder trace - successful
Stopping the Recorder Trace facility:
5. To stop the IMS Connect Recorder Trace facility, enter line action RP.
6. Review the status of your request by entering line action L next to the same system. For example:

<table>
<thead>
<tr>
<th>Time</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>07.28.25</td>
<td>CEXS0501 USR request: HWS Command stop recorder trace</td>
</tr>
<tr>
<td>07.28.25</td>
<td>CEXS0531I UPDATE MEMBER TYPE(IMSCON) STOP(TRACE)</td>
</tr>
<tr>
<td>07.28.25</td>
<td>CEXS0561I F GUIAS01B, UPDATE MEMBER TYPE(IMSCON) STOP(TRACE)</td>
</tr>
<tr>
<td>07.28.25</td>
<td>CEXS0561I HWSR08901 RECORDER CLOSED; M=RCODR</td>
</tr>
<tr>
<td>07.28.25</td>
<td>CEXS0501I USR action : HWS Command stop recorder trace - successful</td>
</tr>
</tbody>
</table>

7. In the Set Tracing dialog, select the **Activate Tracing** option.
8. Press **Enter**. The Command Processor panel appears with the command set.
9. 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.
10. Press the Exit function key (F3) twice to return to the **Systems** view.

**Related concepts:**
[“RECORDER trace utilities” on page 579](#)

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.

**Related tasks:**
[“Using Recorder Trace” on page 202](#)

Use the **Start Recorder Trace** and **Stop Recorder Trace** context menu options in the Status Monitor of the IMS Connect Extensions Operations Console to start and stop the IMS Connect Recorder Trace facility.
Chapter 7. Forwarding a live feed of IMS Connect events

The IMS Connect Extensions feed collects events from running IMS Connect Extensions systems, consolidates the events into one record per IMS Connect transaction, and then forwards those records to one of three destinations. The destination can be: a remote analytics platform that is listening on a TCP/IP port for data in JSON Lines format, a data set, or System Management Facilities (SMF).

You can use the feed to analyze the performance of IMS Connect transactions.

![IMS Connect Extensions feed diagram](image)

**Figure 48. IMS Connect Extensions feed**

The feed job continues to run whether or not the source IMS Connect systems are available. If an IMS Connect system is unavailable when the feed job starts or becomes unavailable later, the job polls the system at regular intervals indefinitely. If the IMS Connect system becomes available, the feed job starts (or restarts) monitoring the system for events.
Feed output: IMS Connect transaction index

The consolidated records output by the feed are known as IMS Connect transaction index records. A data set of such records, which is one of the feed destinations, is known as an IMS Connect transaction index.

You can also use other tools to create and process IMS Connect transaction indexes. For example:

- You can use IBM IMS Performance Analyzer for z/OS to create IMS Connect transaction indexes from IMS Connect Extensions archive journals.
- You can use IMS Performance Analyzer to create reports from IMS Connect transaction indexes.
- You can use the following tools to browse IMS Connect transaction indexes:
  - IBM Transaction Analysis Workbench for z/OS
  - IBM IMS Problem Investigator for z/OS
  These tools classify IMS Connect transaction index records as log type IMS, log code CA20.

Note: For simplicity, where the distinction between the feed job and its output is either obvious or not important, this documentation uses the term feed.

Related concepts:
- Chapter 20, “IMS Connect Extensions publisher API,” on page 463
- Use the IMS Connect Extensions publisher API to externalize IMS Connect Extensions and IMS Connect data to third-party applications.

Starting an IMS Connect Extensions feed

To start an IMS Connect Extensions feed, you run a job that specifies a destination and which IMS Connect systems to use as the feed source.

About this task

The IMS Connect systems that you want to use as the feed source must be running in the same z/OS logical partition (LPAR) as the feed job.

The feed job is a client of the IMS Connect Extensions publisher API. The feed job uses the publisher API to get IMS Connect events.

Procedure

1. For each IMS Connect system that you want to use as the feed source, select the Activate Publisher API option.
   See “Defining IMS Connect systems” on page 320.
2. Create and submit JCL to run a feed job.
   See “IMS Connect Extensions feed JCL” on page 101.

What to do next

Check the feed destination for data forwarded by the feed job.

You can display a list of publisher API clients, including feed jobs, in the IMS Connect Extensions ISPF dialog. You can use this list to display client details and to stop clients. The list identifies feed jobs using a concatenation of job name and
job ID. For details, see message CEX7201I in the SYSPRINT output data set of the
feed job.

Related tasks:

“IMS Connect Extensions publisher API client list” on page 170
Use the IMS Connect Extensions publisher API client list to list summary
information about client applications that are connected to the IMS Connect
Extensions publisher API.

**IMS Connect Extensions feed JCL**

The JCL to run the IMS Connect Extensions feed is concise and standalone: an EXEC
statement for the feed program, CEXCA20P; feed configuration parameters in an
in-stream SYSIN data set; and a SYSPRINT data set for output messages.

**Example**

The following JCL defines a feed that forwards data from three IMS Connect
systems: ICONP01, ICONP02, and ICONP03. The feed consists of selected fields in
JSON Lines format sent over unsecure TCP/IP (no SSL/TLS) to port 1514 on the
host named “analytics”.

```
//UIDCEX JOB NOTIFY=&SYSUID
//CEXCA20 EXEC PGM=CEXCA20P
//STEPLIB DD DISP=SHR,DSN=cexpre.SCEXLINK
//* Output messages
//SYSPRINT DD SYSOUT=*  
//* Configuration parameters
//SYSIN DD *
* Source: IMS Connect systems (must be on the same LPAR as this job)
HWSID=ICONP01,ICONP02,ICONP03

* Destination: JSON Lines over TCP/IP
DESTINATION=JSON
HOST=analytics
PORT=1514
FIELDS(hwsname, trancode, resptime) /* Send only the fields you need

* Other destinations (only one destination allowed per feed job):
* DESTINATION=ARCHIVE
* DSN=...
* DESTINATION=SMF
* TYPE=...
*/
```

**DD statements**

**STEPLIB**

The library containing the IMS Connect Extensions executable load modules.
You do not need to specify the name of this library if the modules reside in the
system LNKLST.

**SYSPRINT**

Output messages. Each message is preceded by the current time stamp and
either of the following identifiers:

- The feed job name
- The name (HWSID) of the IMS Connect system to which the message
  applies

**SYSIN**

Configuration parameters.
The SYSIN data set can also contain comment lines, inline comments, and blank lines. A comment line begins with an asterisk (*) in column one. An inline comment begins after column one, and after any parameter on the line, with a slash followed by an asterisk (/*).

//SYSIN DD *
* Comment line
HWSID=ICONPO1 /* Inline comment
/

**IMS Connect Extensions feed configuration parameters**

IMS Connect Extensions feed configuration parameters specify the feed source and destination.

The source is one or more IMS Connect systems. The destination is one of the following: an archive data set, z/OS Systems Management Facilities (SMF), or JSON Lines over TCP/IP. You specify the configuration parameters in the SYSIN data set of the feed job.

The SYSIN data set must contain the following parameters:

- An HWSID parameter that lists the source IMS Connect systems.
- A DESTINATION parameter, and parameters that specify the required destination details.

Parameter names are case-insensitive.

The case sensitivity of parameter values depends on context: case-sensitive values, such as z/OS UNIX paths, are case-sensitive; others, such as data set names, are not (these are uppercased).

You can specify multiple parameters separated by white space on a single line, or specify each parameter on a separate line. Blank lines are allowed.

**Syntax**

```
HWSID= name
  BUFSIZE=1MB
  BUFSIZE=size
  BUFSIZE=MB
  BUFSIZE=KB
  BUFWAIT=30
  BUFWAIT=seconds

POLL=5
  POLL=seconds
  POLL=percent
  POLL=collection_level

THRESHOLD=100
  THRESHOLD=percent

LEVEL=4
  LEVEL=collection_level

DESTINATION=
  ARCHIVE
  SMF
  JSON
```

**Parameters**

**HWSID**

The names of one or more IMS Connect systems to use as the feed source.
Each name must match the ID attribute of an HWSID statement in an IMS Connect configuration member (HWSCFG\text{"x\text{"x}}) of the IMS PROCLIB data set.

The IMS Connect systems must be running in the same z/OS logical partition (LPAR) as the feed job.

**BUFSIZE**
For each source IMS Connect system, the feed job creates a buffer to accumulate data from that system. BUFSIZE specifies the size of each buffer as an integer number of kilobytes (KB) or megabytes (MB). Minimum buffer size is 4 KB, maximum 100 MB, default 1 MB.

**BUFWAIT**
BUFWAIT and THRESHOLD are closely related. The feed job reads the buffer whenever one of the following two conditions is met, whichever happens first:

- When the buffer wait time has elapsed. This is the number of seconds specified by BUFWAIT.
- When the buffer has filled to the percentage specified by THRESHOLD.

Both values must be integers. The buffer wait time is 5 - 90 seconds, default 30 seconds. The default threshold is 100 percent.

When the feed job reads a buffer, it empties the buffer and resets the buffer wait time clock.

**POLL**
The POLL parameter applies to connections to source IMS Connect systems and, for DESTINATION=JSON, the connection to the destination host.

If the feed job cannot connect to a source IMS Connect system or the destination host for DESTINATION=JSON, or a connection fails while the job is running, the job attempts to connect at the regular intervals specified by POLL. The poll interval is an integer number of seconds, 1 - 900. The default poll interval is 5 seconds.

**LEVEL**
The IMS Connect Extensions collection level. The collection level determines the events that the feed collects. Collection level 2 is the minimum level required to produce feed output. The default value of 4 is the highest collection level.

**DESTINATION**
A feed job can write to one of the following destinations:

**JSON**
Selected fields in JSON Lines format streamed over a TCP/IP network to an analytics platform. For example, Splunk.

**ARCHIVE**
A data set, known as an IMS Connect transaction index, that you can use for long-term archiving.

**SMF**
IMS Connect transaction index records written as System Management Facilities (SMF) records, with standard SMF record headers, to SMF.

Example: Streaming JSON Lines over unsecure TCP

```
//UIDCEX JOB NOTIFY=&SYSUID
//CEXCA20 EXEC PGM=CEXCA20P
//STEPLIB DD DISP=SHR,DSN=cexpre.SCEXLINK
//SYSPRINT DD SYSOUT**
```
IMS Connect Extensions feed parameters for DESTINATION=JSON

If you specify the feed parameter DESTINATION=JSON, then you must also specify at least HOST and PORT. You can optionally specify other parameters, such as secure (SSL/TLS) connection details.

**Syntax**

```plaintext
DESTINATION=JSON HOST=host_name PORT=port_number
TIMEOUT=seconds
FIELDS(field_name)
```

**Parameters that can follow DESTINATION=JSON**

**HOST**

The destination hostname or IP address, up to 255 characters.

**PORT**

The destination port number.

**TIMEOUT**

How long to wait to connect to the destination before timing out. Either:
- An integer number of seconds.
- 0 to wait forever; no timeout. This is the default.

TIMEOUT only applies when the feed job starts. If the destination becomes unavailable later, the feed jobs polls the destination at regular intervals indefinitely. For details, see the POLL parameter described in "IMS Connect Extensions feed configuration parameters" on page 102.

**FIELDS**

A comma-separated list of the fields that you want to include in the feed.

If you omit the FIELDS parameter, then the feed includes all fields. Including all fields can be useful during prototyping to explore all of the available data, but is typically more than you need.

To optimize performance and conserve storage, specify a FIELDS parameter with only the fields that you need.

You can split the list over multiple lines in the SYSIN data set.

The field names that you specify in the FIELDS parameter are case-insensitive. The case of the names that you specify in the FIELDS parameter does not affect the case of field names in the feed.
The order of fields in the feed is fixed. The output order is not affected by the order in which you specify fields in the FIELDS parameter.

For more details, including a list of fields and their output order, see “IMS Connect Extensions feed fields” on page 110.

Security parameters

Security parameters are required only for secure (SSL/TLS) connections.

SECURITY( 
  TLS*,
  TLSV1.2,
  TLSV1.1,
  TLSV1.0
)

FIPS

KEYRING(
  user_id/
  *TOKEN*/pkcs#11_token_name,
  pkcs#12_unix_file_path,
  key_database_unix_file_path
)

PASSWORD(password)

STASH(stash_unix_file_path)

CERTLABEL(label)

CIPHERS(cipher_suites)

SECURITY

Specifies one or more security protocols to try, in order. The special value TLS* represents the list of all supported versions of TLS, starting with the latest version: TLSV1.2, TLSV1.1, TLSV1.0

If you omit the SECURITY parameter, the feed job attempts to open a connection without SSL/TLS.

FIPS

Sets z/OS System SSL Federal Information Processing Standards (FIPS) mode on. For information about FIPS mode, see the z/OS System SSL documentation.

KEYRING

Specifies a collection of certificates that includes the certificates required for this connection. Can be one of the following:

SAF key ring

Specified in the format owner_user_id/key_ring_name or, if the current user owns the key ring, just the key ring name. For example:

my/cex_keyring

If the current user owns the key ring, the current user must have READ access to the IRR.DIGTCERT.LISTRING resource in the FACILITY class. If another user owns the key ring, the current user must have UPDATE access to that resource.

Certificate private keys are not available when using a SAF key ring owned by another user, except for SITE certificates where CONTROL
authority is given to IRR.DIGTCERT.GENCERT in the FACILITY class
or for user certificates where READ or UPDATE authority is given to
ring_owner.ring_name.LST resource in the RDATALIB class.

Key database
A key database created by the z/OS gskkyman utility. The key
database is specified as a z/OS UNIX file path. For example:
/u/my/sslcerts/cex.kdb

PKCS #12 file
Specified as a z/OS UNIX file path. For example:
/u/my/sslcerts/cex.p12

PKCS #11 token
Specified in the format *TOKEN*/token_name. For example:
*TOKEN*/cex.pkcs11.token

The *TOKEN* qualifier indicates that the value refers to a PKCS #11
token rather than a SAF key ring.

If you specify a key database or PKCS #12 file, but you do not specify either a
STASH parameter or a PASSWORD parameter, then the feed job looks for a
stash file in the same directory as the key database or PKCS #12 file, and with
the same base file name, but with .sth extension. For example, if the
KEYRING parameter specifies the following z/OS UNIX file path:
/u/my/sslcerts/cex.kdb

or:
/u/my/sslcerts/cex
(with no extension)
then the feed job looks for a stash file at the following path:
/u/my/sslcerts/cex.sth

STASH
Specifies the z/OS UNIX path of the stash file that contains the password for
the key database or PKCS #12 file.

If KEYRING specifies a SAF key ring or PKCS #11 token, STASH is ignored.

The stash file name must have a .sth extension. If the specified file name has a
different extension, that extension is ignored and replaced with the .sth
extension.

If the PASSWORD parameter is specified, STASH is ignored.

PASSWORD
Specifies the password for the key database or PKCS #12 file.

If KEYRING specifies a SAF key ring or PKCS #11 token, PASSWORD is
ignored.

CIPHERS
Specifies a list of candidate cipher suites to try, in order. The list is a
concatenation of 4-digit hexadecimal cipher suite numbers supported by z/OS
System SSL. For example:
CIPHERS(000A000D001000130016)

If you omit CIPHERS, the feed job uses the system default list of cipher suites.
That list changes depending on whether or not FIPS mode is on.
Tip: To match a z/OS System SSL cipher suite number to the corresponding OpenSSL cipher suite name, go to the z/OS System SSL documentation and look up the "short name" for that cipher suite in the table of cipher suite definitions. The short name is the name that is defined in the associated RFC. Then go to the OpenSSL documentation for the `ciphers` command, and use the RFC name to find the corresponding OpenSSL name.

For more information on cipher suite definitions, see the z/OS System SSL documentation.

CERTLABEL
Specifies the label of the client certificate that is used to authenticate the IMS Connect Extensions feed job (the client, in this context) to the destination host (server). The client certificate, and its private key, must be in the collection that is specified by the KEYRING parameter.

CERTLABEL is only used if the destination host requires client authentication.

If the destination host requires client authentication, but you omit CERTLABEL, then the feed job uses the default certificate from the collection that is specified by the KEYRING parameter.

Example: Streaming JSON Lines over unsecure TCP

```
//UIDCEX JOB NOTIFY=&SYSUID
//CEXCA20 EXEC PGM=CEXCA20P
//STEP LIB DD DISP=SHR,DSN=cexpre.SCEXLINK
//SYSPRINT DD SYSOUT=* 
//SYSIN DD *
HWSID=ICONP01,ICONP02,ICONP03

DESTINATION=JSON 
    HOST=analytics 
    PORT=1514 
    FIELDS(HWSNAME, 
        TRANCODE, 
        RESPTIME)
/*

Example output line:

{"time":"2018-11-01T15:25:03.123456Z","type":"ims-ca20","hwsname":"ICONP01","resptime":0.654321,"trancode":"TRNA"}
```

Example: Secure TCP

For a secure TCP port, insert the following parameters after DESTINATION=JSON in the previous example, where my/cex.analytics identifies a keyring that you have defined:

```
SECURITY(TLS*) 
FIPS
KEYRING(my/cex.analytics)
```

IMS Connect Extensions feed parameters for DESTINATION=ARCHIVE

If you specify the feed parameter DESTINATION=ARCHIVE, then you must also specify at least DSN. You can optionally specify other parameters for the archive data set attributes.
Syntax

DESTINATION=ARCHIVE—DSN=data_set_name_pattern

DATACLAS=data_class_name—DSNTYPE=BASIC

MGMTCLAS=management_class_name—SPACE=space_specifications

STORCLAS=storage_class_name—UNIT=unit_specifications

VOLSER=volume_serial_number,

Parameters that can follow DESTINATION=ARCHIVE

DSN

Data set name pattern for the output IMS Connect transaction index.

A pattern that ends with (+1) refers to the next generation in a generation data group (GDG). You must have already built the GDG base entry. For information about building GDG base entries, see the z/OS MVS JCL user documentation.

If the pattern does not end with (+1), the data set pattern indicates a dynamic data set.

For a dynamic data set, the pattern must contain substitution variables to ensure a unique data set name.

For a GDG, the pattern can optionally contain substitution variables to refer to different GDG bases.

The following substitution variables apply only to dynamic data sets. A dynamic data set name pattern must include all of these substitution variables:

Table 3. Substitution variables that apply only to dynamic data sets and are required

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description of substituted value</th>
<th>Example substituted value</th>
</tr>
</thead>
<tbody>
<tr>
<td>+DATE</td>
<td>Current date in the format Ddymmdd</td>
<td>01080720</td>
</tr>
<tr>
<td>+TIME</td>
<td>Current time in the format Thhmmss</td>
<td>T134723</td>
</tr>
<tr>
<td>+GEN</td>
<td>Generation number in the format Xnnn where nnn is a 3-digit number starting at 001 and incremented by 1 for each generation. When nnn reaches 999, subsequent numbers begin again at 001. The generation number ensures that if two data sets need to be allocated at the same time then the names are unique.</td>
<td>X001</td>
</tr>
</tbody>
</table>

Any pattern can optionally contain the following substitution variables:
Table 4. Substitution variables that can be used in any data set name pattern

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description of substituted value</th>
<th>Example substituted value</th>
</tr>
</thead>
<tbody>
<tr>
<td>+HWSID</td>
<td>IMS Connect system ID. See the HWSID feed parameter.</td>
<td>ICONP01</td>
</tr>
<tr>
<td>+USERID</td>
<td>The user ID of the feed job.</td>
<td>CEX</td>
</tr>
</tbody>
</table>

The feed allocates and writes to the data set specified by the pattern. When the current data set is full, the feed allocates and writes to a new data set. For a GDG pattern, the feed allocates and writes to the next generation. For a dynamic data set pattern, the feed reevaluates the substitution variables and allocates and writes to a data set with the new name based on those new values: the current date and time, and incremented generation number.

The feed job records the archive data set names in the SYSPRINT output data set.

DSN is the only parameter that must be specified with DESTINATION=ARCHIVE.

The remaining parameters are optional and correspond to parameters of the same name of the JCL DD statement:
- DATACLAS
- DSNTYPE limited to the values BASIC, EXTPREF, EXTRQ, and LARGE
- EATTR
- MGMTCLAS
- SPACE
- STORCLAS
- UNIT
- VOLSER corresponds to the VOLUME=SER parameter of the DD statement

If you omit these parameters, then your installation defaults apply. For more information on these parameters, see the z/OS MVS JCL reference documentation for the DD statement.

Example: Dynamic data set

```plaintext
//UIDCEX JOB NOTIFY=&SYSUID
//CEXCA20 EXEC PGM=CEXCA20P
//STEPLIB DD DISP=SHR,DSN=cexpre.SCEXLINK
//SYSPRINT DD SYSOUT=
//SYSIN DD *
HWSID=ICONP01

DESTINATION=ARCHIVE
DSN=CEX.+HWSID.+DATE.+TIME.+GEN
UNIT=(3390,2)
VOLSER=(96341,96342)
/*

Example output data set names:
- CEX.ICONP01.D180720.T134723.X001
- CEX.ICONP01.D180720.T135720.X002
- CEX.ICONP01.D180720.T140717.X003
```

Example: GDG

This example assumes that GDG base entries exist for CEX.ICONP01.ARCHIVE, CEX.ICONP02.ARCHIVE, and CEX.ICONP03.ARCHIVE.
IMS Connect Extensions feed parameters for DESTINATION=SMF

If you specify the feed parameter DESTINATION=SMF, then you must also specify TYPE.

Syntax

DESTINATION=SMF—TYPE=decimal_smf_record_type

SUBTYPE=decimal_smf_record_subtype

Parameters that can follow DESTINATION=SMF

TYPE
Decimal SMF record type, 127 - 1151. Required.

SUBTYPE
Decimal SMF record subtype, 0 - 32767. Default 0.

Example

DESTINATION=SMF
TYPE=127
SUBTYPE=1000

IMS Connect Extensions feed fields

If you specify JSON as the destination of the IMS Connect Extensions feed, then you can select which fields to forward. Other destinations receive complete IMS Connect transaction index records.

The following table lists the fields that you can specify in the feed FIELDS parameter.
### Table 5. IMS Connect Extensions feed fields

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Form field in IMS Performance Analyzer</th>
<th>Parent field</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>time</td>
<td>Event time stamp. The IMS Connect transaction start time as a date and time of day string value in ISO 8601 extended format: &quot;yyyy-mm-ddThh:mm:ss.SSSSSSZ&quot;</td>
<td>STARTCON</td>
<td></td>
<td>Identifiers</td>
</tr>
<tr>
<td>taskid</td>
<td>Task ID</td>
<td></td>
<td></td>
<td>Identifiers</td>
</tr>
<tr>
<td>collecttask</td>
<td>Collect task number</td>
<td></td>
<td></td>
<td>Identifiers</td>
</tr>
<tr>
<td>cextecltask</td>
<td>CEXTECL task number</td>
<td></td>
<td></td>
<td>Identifiers</td>
</tr>
<tr>
<td>hwsname</td>
<td>Connect system name</td>
<td>HWSID</td>
<td></td>
<td>Identifiers</td>
</tr>
<tr>
<td>logontk</td>
<td>EVKEY SVT STCK token</td>
<td>CONNLTK</td>
<td></td>
<td>Identifiers</td>
</tr>
<tr>
<td>otmassn</td>
<td>EVT 41 SSN</td>
<td></td>
<td></td>
<td>Identifiers</td>
</tr>
<tr>
<td>port</td>
<td>Port number</td>
<td>PORT</td>
<td></td>
<td>Identifiers</td>
</tr>
<tr>
<td>socket</td>
<td>Socket number</td>
<td>SOCKET</td>
<td></td>
<td>Identifiers</td>
</tr>
<tr>
<td>trancode</td>
<td>SVT_TXNAME Transaction Code</td>
<td>TRANCODE</td>
<td></td>
<td>Identifiers</td>
</tr>
<tr>
<td>userid</td>
<td>SVT_UID User ID</td>
<td>USERID</td>
<td></td>
<td>Identifiers</td>
</tr>
<tr>
<td>originds</td>
<td>SVT_DSORIG Datastore (Origin)</td>
<td>ORIGDS</td>
<td></td>
<td>Identifiers</td>
</tr>
<tr>
<td>targetds</td>
<td>SVT_DSTARG Datastore (Target)</td>
<td>TARGDS</td>
<td></td>
<td>Identifiers</td>
</tr>
<tr>
<td>clientid</td>
<td>IRM_CLID Client ID</td>
<td>CLIENTID</td>
<td></td>
<td>Identifiers</td>
</tr>
<tr>
<td>tpipe</td>
<td>TPIPE name</td>
<td>TPIPE</td>
<td></td>
<td>Identifiers</td>
</tr>
<tr>
<td>xmladapter</td>
<td>XML Adapter name</td>
<td>XMLADAPT</td>
<td></td>
<td>Identifiers</td>
</tr>
<tr>
<td>altlterm</td>
<td>Alternate LTERM</td>
<td>LTERMOVR</td>
<td></td>
<td>Identifiers</td>
</tr>
<tr>
<td>reroute</td>
<td>Re-route name</td>
<td>REROUTNM</td>
<td></td>
<td>Identifiers</td>
</tr>
<tr>
<td>readexit</td>
<td>Read exit name</td>
<td>EXITNAME</td>
<td></td>
<td>Identifiers</td>
</tr>
<tr>
<td>tmember</td>
<td>Tmember (and others) name</td>
<td>TMEMBERD</td>
<td></td>
<td>Identifiers</td>
</tr>
<tr>
<td>tdds</td>
<td>The tmember field value refers to a Datastore</td>
<td>DSIDFlag</td>
<td></td>
<td>Identifiers</td>
</tr>
<tr>
<td>idscri</td>
<td>The tmember field value refers to SCI</td>
<td>DSIDFlag</td>
<td></td>
<td>Identifiers</td>
</tr>
<tr>
<td>idmember</td>
<td>The tmember field value refers to a Member</td>
<td>DSIDFlag</td>
<td></td>
<td>Identifiers</td>
</tr>
<tr>
<td>idtmember</td>
<td>The tmember field value refers to a TMEMBER</td>
<td>DSIDFlag</td>
<td></td>
<td>Identifiers</td>
</tr>
<tr>
<td>idodbm</td>
<td>The tmember field value refers to ODBM</td>
<td>DSIDFlag</td>
<td></td>
<td>Identifiers</td>
</tr>
<tr>
<td>idmsc</td>
<td>The tmember field value refers to MSC</td>
<td>DSIDFlag</td>
<td></td>
<td>Identifiers</td>
</tr>
<tr>
<td>ipaddress</td>
<td>IP address (IPv4 or IPv6)</td>
<td>IPADDR</td>
<td></td>
<td>Identifiers</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Form field in IMS Performance Analyzer</td>
<td>Parent field</td>
<td>Category</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------------------------------------</td>
<td>--------------</td>
<td>---------------</td>
</tr>
<tr>
<td>ipaddrv6</td>
<td>ipaddress field contains IPv6 address (IPv4 if not set)</td>
<td>Aflag2</td>
<td>Identifiers</td>
<td></td>
</tr>
<tr>
<td>inputtk</td>
<td>Input Msg EVKEY SVT Token (58)</td>
<td>CONNOTOK</td>
<td>Identifiers</td>
<td></td>
</tr>
<tr>
<td>type</td>
<td>Fixed string value: &quot;ims:ca20&quot;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Here, ims is the log type and ca20 is the log code.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>otmaack</td>
<td>OTMA acknowledgment</td>
<td>IMSACK</td>
<td>Aflag1</td>
<td>Characteristics</td>
</tr>
<tr>
<td>istran</td>
<td>Transaction type message</td>
<td>Aflag1</td>
<td>Characteristics</td>
<td></td>
</tr>
<tr>
<td>rtpnoauto</td>
<td>Resume Tpipe Noauto command</td>
<td>Aflag2</td>
<td>Characteristics</td>
<td></td>
</tr>
<tr>
<td>rtpauto</td>
<td>Resume Tpipe Auto command</td>
<td>Aflag2</td>
<td>Characteristics</td>
<td></td>
</tr>
<tr>
<td>rtpsingle</td>
<td>Resume Tpipe Single command</td>
<td>Aflag2</td>
<td>Characteristics</td>
<td></td>
</tr>
<tr>
<td>sendonly</td>
<td>Send-Only message</td>
<td>Aflag2</td>
<td>Characteristics</td>
<td></td>
</tr>
<tr>
<td>lptran</td>
<td>Local Port Transaction</td>
<td>PORTTYPE</td>
<td>Aflag2</td>
<td>Characteristics</td>
</tr>
<tr>
<td>timerset</td>
<td>IRM Timer value set</td>
<td>Aflag2</td>
<td>Characteristics</td>
<td></td>
</tr>
<tr>
<td>pstran</td>
<td>Persistent Socket Transaction</td>
<td></td>
<td>Pflag1</td>
<td>Characteristics</td>
</tr>
<tr>
<td>sortcand</td>
<td>SendOnly-ResTpipe candidate</td>
<td></td>
<td>Pflag2</td>
<td>Characteristics</td>
</tr>
<tr>
<td>sortmerge</td>
<td>SendOnly-ResTpipe merged record</td>
<td></td>
<td>Pflag2</td>
<td>Characteristics</td>
</tr>
<tr>
<td>tovalue</td>
<td>Timeout value</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cm0</td>
<td>Commit mode 0</td>
<td>SyncFlag</td>
<td>Characteristics</td>
<td></td>
</tr>
<tr>
<td>cm1</td>
<td>Commit mode 1</td>
<td>SyncFlag</td>
<td>Characteristics</td>
<td></td>
</tr>
<tr>
<td>sl0</td>
<td>Synch level 0 (NONE)</td>
<td>SyncFlag</td>
<td>Characteristics</td>
<td></td>
</tr>
<tr>
<td>sl1</td>
<td>Synch level 1 (CONFIRM)</td>
<td>SyncFlag</td>
<td>Characteristics</td>
<td></td>
</tr>
<tr>
<td>sl2</td>
<td>Synch level 2 (SYNCPOINT)</td>
<td>SyncFlag</td>
<td>Characteristics</td>
<td></td>
</tr>
<tr>
<td>purgend</td>
<td>Purge if not deliverable</td>
<td>CLIACK</td>
<td>SyncFlag</td>
<td>Characteristics</td>
</tr>
<tr>
<td>clirr</td>
<td>Client re-route</td>
<td>SyncFlag</td>
<td>Characteristics</td>
<td></td>
</tr>
<tr>
<td>npsocket</td>
<td>Non-Persistent Socket</td>
<td>SocketFlag</td>
<td>Characteristics</td>
<td></td>
</tr>
<tr>
<td>psocket</td>
<td>Persistent Socket</td>
<td>SocketFlag</td>
<td>Characteristics</td>
<td></td>
</tr>
<tr>
<td>tssocket</td>
<td>Transaction Socket</td>
<td>SocketFlag</td>
<td>Characteristics</td>
<td></td>
</tr>
<tr>
<td>msgrr</td>
<td>Re-Route this message</td>
<td>CLIACK</td>
<td>SocketFlag</td>
<td>Characteristics</td>
</tr>
<tr>
<td>cfsendonly</td>
<td>Send only message</td>
<td>ClientFlag</td>
<td>Characteristics</td>
<td></td>
</tr>
<tr>
<td>cfsetageing</td>
<td>Message aging indicator for an input transaction. If used for a NAK of a synchronous callout msg, OTMA will keep the msg until the timeout is reached or a new Resume Tpipe for callout msg is processed.</td>
<td>ClientFlag</td>
<td>Characteristics</td>
<td></td>
</tr>
<tr>
<td>cfreroute</td>
<td>CM0 Reroute request</td>
<td>ClientFlag</td>
<td>Characteristics</td>
<td></td>
</tr>
<tr>
<td>cfpurge</td>
<td>Purge if not deliverable</td>
<td>ClientFlag</td>
<td>Characteristics</td>
<td></td>
</tr>
<tr>
<td>cfwlmcorr</td>
<td>EWLM correlator present in OMUSR_EWLMCORR field of User data segment</td>
<td>ClientFlag</td>
<td>Characteristics</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Form field in IMS Performance Analyzer</td>
<td>Parent field</td>
<td>Category</td>
</tr>
<tr>
<td>------------------</td>
<td>------------------------------------------</td>
<td>----------------------------------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>cfignpurge</td>
<td>Force IMS to ignore Purge</td>
<td></td>
<td>ClientFlag</td>
<td>Characteristics</td>
</tr>
<tr>
<td>cfsonlyenq</td>
<td>Send-only ordered serial enqueue</td>
<td></td>
<td>ClientFlag</td>
<td>Characteristics</td>
</tr>
<tr>
<td>irmtime</td>
<td>IRM_TIMER</td>
<td></td>
<td>ClientFlag</td>
<td>Characteristics</td>
</tr>
<tr>
<td>otmadelay</td>
<td>OTMA Delay (01-41 + 42-31)</td>
<td>DLAYOTMA</td>
<td></td>
<td>Elapsed times</td>
</tr>
<tr>
<td>inputelap</td>
<td>41-3C Input</td>
<td>PREOTMA</td>
<td></td>
<td>Elapsed times</td>
</tr>
<tr>
<td>rdsockelap</td>
<td>49-3C Input READ Socket</td>
<td>INREAD</td>
<td></td>
<td>Elapsed times</td>
</tr>
<tr>
<td>readxelap</td>
<td>3E-3D Message Exit READ</td>
<td>READEXIT</td>
<td></td>
<td>Elapsed times</td>
</tr>
<tr>
<td>rxmlxelap</td>
<td>3E-3D Message Exit RXML</td>
<td>RXMLEXIT</td>
<td></td>
<td>Elapsed times</td>
</tr>
<tr>
<td>safelap</td>
<td>40-3F SAF Call</td>
<td>SAFTIME</td>
<td></td>
<td>Elapsed times</td>
</tr>
<tr>
<td>otmaelap</td>
<td>42-41 OTMA</td>
<td>PROCOTMA</td>
<td></td>
<td>Elapsed times</td>
</tr>
<tr>
<td>xmitxelap</td>
<td>3E-3D Message Exit XMIT</td>
<td>XMITEXIT</td>
<td></td>
<td>Elapsed times</td>
</tr>
<tr>
<td>rdackelap</td>
<td>49-4A Acknowledgment READ Socket</td>
<td>ACKREAD</td>
<td></td>
<td>Elapsed times</td>
</tr>
<tr>
<td>confelap</td>
<td>41-42 Transaction Confirm</td>
<td>CONFIRM</td>
<td></td>
<td>Elapsed times</td>
</tr>
<tr>
<td>trackelap</td>
<td>42-41 ACK Transaction Confirm</td>
<td>CONFOTMA</td>
<td></td>
<td>Elapsed times</td>
</tr>
<tr>
<td>rtpelap</td>
<td>42-48 Resume-Tpipe time</td>
<td>OUTRCON</td>
<td></td>
<td>Elapsed times</td>
</tr>
<tr>
<td>outputelap</td>
<td>48-42 Output</td>
<td>POSTOTMA</td>
<td></td>
<td>Elapsed times</td>
</tr>
<tr>
<td>resptime</td>
<td>48-3C Response time</td>
<td>RSPCON</td>
<td></td>
<td>Elapsed times</td>
</tr>
<tr>
<td>portdepth</td>
<td>Port depth</td>
<td>PORTDEP</td>
<td></td>
<td>Statistics</td>
</tr>
<tr>
<td>rtpmsgct</td>
<td>Resume Tpipe message count</td>
<td></td>
<td></td>
<td>Statistics</td>
</tr>
<tr>
<td>cliackct</td>
<td>Client ACK count</td>
<td></td>
<td></td>
<td>Statistics</td>
</tr>
<tr>
<td>clinakct</td>
<td>Client NAK count</td>
<td></td>
<td></td>
<td>Statistics</td>
</tr>
<tr>
<td>otmanakct</td>
<td>OTMA NAK count</td>
<td></td>
<td></td>
<td>Statistics</td>
</tr>
<tr>
<td>complvl</td>
<td>Completion level</td>
<td>COMPLVLC</td>
<td></td>
<td>Statistics</td>
</tr>
<tr>
<td>rejected</td>
<td>Transaction was rejected</td>
<td>REJECT</td>
<td></td>
<td>Status</td>
</tr>
<tr>
<td>timedout</td>
<td>Time out</td>
<td>TIMEOUT</td>
<td>Aflag1</td>
<td>Status</td>
</tr>
<tr>
<td>failed</td>
<td>Transaction failed</td>
<td>FAILED</td>
<td>Aflag1</td>
<td>Status</td>
</tr>
<tr>
<td>trcactive</td>
<td>Trace started for this SVT</td>
<td></td>
<td>Pflag1</td>
<td>Status</td>
</tr>
<tr>
<td>trcoflow</td>
<td>Trace record has overflowed</td>
<td></td>
<td>Pflag1</td>
<td>Status</td>
</tr>
<tr>
<td>tranactiv</td>
<td>Tran was active this session</td>
<td></td>
<td>Pflag1</td>
<td>Status</td>
</tr>
<tr>
<td>closedport</td>
<td>Transaction closed the port</td>
<td></td>
<td>Pflag1</td>
<td>Status</td>
</tr>
<tr>
<td>writeread</td>
<td>WRITE-READ (READ ACK)</td>
<td></td>
<td>Pflag1</td>
<td>Status</td>
</tr>
<tr>
<td>psacksent</td>
<td>Persistent Socket (41) Ack sent</td>
<td></td>
<td>Pflag1</td>
<td>Status</td>
</tr>
<tr>
<td>pstransent</td>
<td>Persistent Socket (41) Trans sent</td>
<td></td>
<td>Pflag1</td>
<td>Status</td>
</tr>
<tr>
<td>eottrigger</td>
<td>End of Transaction (Trigger)</td>
<td></td>
<td>Pflag2</td>
<td>Status</td>
</tr>
<tr>
<td>naksc</td>
<td>OTMA NAK Sense Code (OMCTLSNC)</td>
<td></td>
<td></td>
<td>Status</td>
</tr>
<tr>
<td>sensecode</td>
<td>Sense Code (OMCTLSNC)</td>
<td></td>
<td></td>
<td>Status</td>
</tr>
<tr>
<td>cficalresp</td>
<td>ICAL response in error</td>
<td></td>
<td>ClientFlag</td>
<td>Status</td>
</tr>
</tbody>
</table>
Categories are descriptive only

The “Category” column in the previous table groups fields according to the nature of the information they contain about a transaction. These categories exist for documentation purposes only, to help you select fields that are of interest to you.

Field output order

The previous table does not reflect the order of fields in the feed JSON Lines output.

Each line of the feed JSON Lines output contains fields in the following order:
1. time
2. type
3. Remaining field names in alphabetical order

The time and type fields are always included

Each line of the feed JSON Lines output always begins with a time field followed by a type field:

{"time":"2018-11-01T15:25:03.123456Z","type":"ims-ca20",... }

Do not specify time or type in the FIELDS parameter.

Field names

In general, the feed field names match IMS Connect transaction index field names, with the following variations and exceptions:

- The field names in the feed JSON Lines output are all lowercase.
- The time field in the feed corresponds to the starttime field in an IMS Connect transaction index.
- The type field has no corresponding field in an IMS Connect transaction index.
- The fields that you can specify in the FIELDS parameter are a subset of the IMS Connect transaction index fields.

Some feed fields match form fields in IBM IMS Performance Analyzer for z/OS. For more detailed descriptions of these fields, see the IMS Performance Analyzer documentation.

Fields with parent fields

Fields in the previous table with a “Parent field” are either flag bits or flag byte values. In the feed JSON Lines output, these fields have the boolean value true or false.

The “Parent field” column makes it easier to find these child fields when browsing IMS Connect transaction index records in IBM Transaction Analysis Workbench for z/OS or IBM IMS Problem Investigator for z/OS. You browse a record, and then select a parent field to browse its child field values.

Elapsed time values

Elapsed time field values are represented as a number of seconds to microsecond precision; six decimal places.
Fields with hexadecimal values

Some fields, such as computer-generated identifiers or correlation tokens, are represented as a string of hexadecimal digits. Each pair of digits represents a byte. For example, the 4-character string value "10FF" represents a sequence of two bytes, decimal values 16 and 255.

Elapsed time fields that cannot be calculated are omitted

Elapsed times are typically calculated from the difference between event time stamps. If an event that is required to calculate an elapsed time did not occur or was not collected, then the elapsed time cannot be calculated. Rather than including a bogus value, such as 0.0, which might skew statistics for the field, the feed omits the field. The line of JSON for that transaction does not include a property for that field.

Empty string fields are omitted

If a string field value is empty (""), the feed omits the field. The line of JSON for that transaction does not include a property for that field.

Collection level can determine whether a field has a value

The LEVEL configuration parameter of the feed job specifies the IMS Connect Extensions collection level. The collection level controls which events the feed collects.

Some fields, such as some elapsed time fields, rely on events that are only collected at a certain minimum collection level. For details, see Chapter 24, "Events recorded by IMS Connect Extensions," on page 551.
Part 3. Managing IMS Connect systems

These topics describe ways in which you can monitor and manage IMS Connect systems using IMS Connect Extensions.

Topics:
- Chapter 8, “ISPF Operations dialog,” on page 119
- Chapter 9, “IMS Connect Extensions Operations Console for z/OS Explorer,” on page 175
- Chapter 10, “Automating IMS Connect operations with REXX,” on page 221
- Chapter 11, “Command access in IMS Connect Extensions,” on page 235
Chapter 8. ISPF 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:

Figure 49. 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.

Using the Operations dialog

Use the primary menu of the IMS Connect Extensions ISPF dialog to access the Operations dialog.

About this task

Note the following information when using the Operations dialog:

- **Grouping systems**
  You can view the IMS Connect systems defined in your IMS Connect Extensions individually, or by group. 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 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. 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.

  For more information on grouping systems, see “Defining system groups” on page 355.

- **Access control**
  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.
Procedure

1. From the IMS Connect Extensions primary menu, select option 2 Operations.

2. View your systems individually or by group:
   - To view a list of IMS Connect systems in a group:
     a. Select view 1 Groups.

   ```
   File Menu Help
   Command ===> Operations - Groups View Row 1 from 5
   Scroll ===> CSR
   View . . 1. Groups 2. Systems
   Filter . . ______
   / Group Systems Description
   DEV 4 Development environment
   NY 2 New York sysplex
   PROD 4 Production environment
   TEST 4 Test environment
   TOKYO 2 Tokyo sysplex
   ****************************************** Bottom of data ******************************************
   Figure 50. Example of groups view in the Operations dialog
   
   b. Enter line action 5 next to the group you are interested in.

   ```
   File Menu Help
   Command ===> Operations - Systems View Filter Mode: More Scroll ===> PAGE
   Group . : NY
   Filter . . ______ Exclude inactive systems . . 2 1. Yes 2. No
   / System Status Events Level Active Used Used Plan Plan
   RAPI03 ACTIVE OFF 4
   RAPI04 ACTIVE ON 4 P03 48% 88% PEAK
   RAPI05 ACTIVE ON 4 P03 2% 67% PEAK
   RAPI06 ACTIVE ON 4 P03 82% 6% OFFPEAK
   ****************************************** Bottom of data ******************************************
   Figure 51. The IMS Connect Extensions Operations - Systems View panel accessed by selecting a group
   
   • To view a list of all IMS Connect systems in your repository, select view 2 Systems.
The following information is displayed on the Operations - Systems panel:

**Group**  
If you entered the Operations - Systems View panel by selecting a group, the group name is displayed at the top of the panel.

**Filter**  
You can limit which systems appear by specifying a filter. 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.

**Exclude inactive systems**  
To exclude inactive systems from the display, select 1. To show all systems, select 2.

**Systems list**  
You can view information for each system in the systems list, sort systems by column by tabbing to the line under the column and pressing Enter, or view additional columns of information by pressing the Right function key (F11). The systems list displays the following information.

**System**  
IMS Connect system name.

**Status**  
The status of the IMS Connect system. A system can be ACTIVE or INACT (inactive).

**Events**  
Status of IMS Connect Extensions event collection. Event collection can be ON or OFF.

**Coll. Level**  
The current event collection level. See "Defining IMS Connect systems” on page 320.

- **0 - Minimum Level**  
  Collects startup and shutdown events along with some infrequent error events.
1 - Accounting Level
Collects Return from Exit events, Open Transaction Manager Access (OTMA) timeout and session error events, and Open Database Manager (ODBM) registration and routing events. This level provides accounting information in terms of the number of messages by Transaction, User Exit, and more.

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 Connect connectivity.

4 - Maximum Level
Collects all event records.

Journal
For IMS Connect Extensions V3.1 systems and higher, the status of the IMS Connect Extensions journal.

Active The name of the active IMS Connect Extensions journal.

Used The percentage of the active journal that is used.

Sockets Used
For IMS Connect Extensions V3.1 systems and higher, the current number of sockets used as a percentage of MAXSOC.

OTMA Plan
The currently active OTMA routing plan. See Chapter 12, “OTMA workload routing in IMS Connect,” on page 245.

ODBM Plan
The currently active ODBM routing plan. See Chapter 14, “ODBM workload routing in IMS Connect,” on page 279.

VRM The version of IMS Connect Extensions that is running.

Super Member
Name if SMEMBER is defined in the system configuration.

Port Port number for the IMS Connect Extensions console.

Host IP address or DNS name of the IMS Connect system server.

Start Time The date and time that the IMS Connect was started

Description System description.

3. Enter one of the following line actions next to a system, or enter / for a complete list of options:

$M or $S
Open the Status Monitor. Use the Status Monitor to view the components of the system and their activity. Use this option to manage exits and IMS data stores. See “Systems Overview” on page 126.

Tip: You can also use line action $M on a group of IMS Connect systems.
AS  Display active sessions. View active sessions on the system and perform operations such as stopping sessions. See “Active Sessions” on page 130.

Tip: You can also use line action AS on a group of IMS Connect systems.

DS  Open the Datastore Monitor. Use the Datastore Monitor to view detailed information about IMS data stores and start, stop, drain, or resume selected IMS data stores. See “Datastore Monitor” on page 141.

CX  Open the Commands panel. The Commands panel is used to control the operation of IMS Connect Extensions and user message exits. See “Issuing IMS Connect Extensions commands from the Commands panel” on page 145.

L  Open the message log. See “Browsing the message log for an IMS Connect” on page 164.

SH  Open a command shell to issue IMS Connect and IMS type-1 commands. See “Issuing IMS Connect and IMS type-1 commands from the CONSOLE dialog” on page 165.

PU  Display publisher client status list. See “IMS Connect Extensions publisher API client list” on page 170.

TR  Start, stop, and set conditional trace parameters for the IMS Connect Extensions trace. See “Using the IMS Connect Extensions trace” on page 93.

RS  Start the IMS Connect Recorder Trace facility. See “Controlling the IMS Connect Recorder Trace facility” on page 96.

Note: You cannot query the current status of the IMS Connect Recorder Trace facility through IMS Connect Extensions but you can use line action L in the Operations dialog to view a history of start and stop requests. For more advanced tracing, use the IMS Connect Extensions.

RP  Stop the IMS Connect Recorder Trace facility. See “Controlling the IMS Connect Recorder Trace facility” on page 96.

J  Switch the current active IMS Connect Extensions journal to the next journal. See “Managing the IMS Connect Extensions journal” on page 163.

Tip: Use the Journal Used column to monitor journal usage.

P  Stop the system.

F  Force stop the system.

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

AC  Display ACCE cache statistics, including cache space usage and statistics on different types of cache services requests. See “Caching user credentials to improve performance” on page 301.
Status Monitor

Use the Status Monitor in the IMS Connect Extensions ISPF dialog to view activity on one or more IMS Connect systems, and to perform actions such as starting and stopping IMS data stores, 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).

Notes:
1. To display activity for a system, statistics collection must be active. For more information, see “Defining IMS Connect systems” on page 320.
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).

Related concepts:
"Status Monitor” on page 187

The Status Monitor in the IMS Connect Extensions Operations Console allows you to monitor and control IMS Connect systems and their components.

Ports Overview

To view the active TCP/IP ports of the IMS Connect system in the IMS Connect Extensions Status Monitor, select the Ports view.

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

Figure 53. 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.
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.

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).

The suffix on the port number identifies the port type.
- 0 indicates a DRDA port.
- $ 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.

**Related tasks:**
- “Managing IMS Connect TCP/IP ports” on page 203
- Use the Ports tab in Status Monitor of the Operations Console to start and stop IMS Connect TCP/IP ports from the IMS Connect Extensions Operations Console.

**Systems Overview**
To view the components of the IMS Connect system in the IMS Connect Extensions Status Monitor, select the Systems view.

The following figure shows an example of the System Overview panel:
1. The interval determines the time window that the summary statistics represent.

2. The panel summarizes statistics for each system component:
   - HWS: IMS Connect system
   - DS: IMS data store
   - DSG: IMS data store group
   - EXIT: User exit
   - ODBM: Open Database Manager (ODBM) instance
   - ALIAS: The alias name of an IMS data store
   - MSC: IMS Multiple Systems Coupling (MSC) 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 “Using the Operations dialog” on page 120 for a summary card or enter / (slash) next to a component to view all commands available for that component.

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.

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.

Toogle a filter on and off with these primary commands:
FILTER ON
FILTER OFF

### 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.

<table>
<thead>
<tr>
<th>Command</th>
<th>Form Definition</th>
<th>Row 1 to 4 of 4</th>
<th>Scroll</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count/</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/</td>
<td>Name</td>
<td>Length</td>
<td>Func</td>
<td>Description</td>
</tr>
</tbody>
</table>
- INPUT  | COUNT           | TOTAL           | Input Messages |
- INPUT  | LENGTH          | TOTAL           | Input Messages |
- INPUT  | LENGTH          | MIN             | Input Messages |
- INPUT  | LENGTH          | MAX             | Input Messages |

Figure 55. Status Monitor: Ports Overview

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 6. Message types for which statistics are collected

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Count functions?</th>
<th>Length functions?</th>
</tr>
</thead>
<tbody>
<tr>
<td>INPUT</td>
<td>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.</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>RPIPE</td>
<td>Resume TPIPE messages processed by the READ exit.</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>SENDONLY</td>
<td>Send-Only messages processed by the READ exit.</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>SENTERR</td>
<td>“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.</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>SENTOK</td>
<td>“Sent OK” messages are any messages not containing the RSM ID string. These include responses, application data, command responses, and DFS messages.</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>ACCEPTED</td>
<td>Messages processed by the READ exit with RC=0 that were not then rejected by IMS Connect Extensions because of pacing, security, or routing.</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>REJECTED</td>
<td>Total of messages rejected.</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>ROUTED</td>
<td>Messages routed by IMS Connect Extensions to a target datastore that is different from the original destination.</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>ACK</td>
<td>Positive acknowledgment (ACK) from the remote client.</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>NAK</td>
<td>Negative acknowledgment (NAK) from the remote client.</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>DEALLOC</td>
<td>Deallocation response from remote client.</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>RETURNED</td>
<td>Input messages for which the READ exit returned RC=4. Such messages include an RSM, ping response, or user data.</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>REJEXER</td>
<td>Messages rejected by the EXER exit routine because of a protocol error.</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>REJSEC</td>
<td>Messages rejected by IMS Connect Extensions security processing.</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>REJPACE</td>
<td>Messages rejected by IMS Connect Extensions pacing.</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>REJRUTE</td>
<td>Messages rejected by IMS Connect Extensions routing processing.</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>REQUEST</td>
<td>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.</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>IGNORE</td>
<td>Messages passed to the message exit routine that were not processed by IMS Connect Extensions. Includes cases where the exit was marked as Inactive.</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>REJOTHER</td>
<td>“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.</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>

**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.

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.

Active Sessions

Use the Active Sessions panel in the IMS Connect Extensions ISPF dialog to manage and monitor active OTMA, ODBM, and MSC sessions in IMS Connect. 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, the Active Sessions panel is of most use when you are aware of a transaction that is taking longer to complete than expected.

Note: To display activity for a system, event collection must be active and the collection level must be set to a value of 1 or higher. For more information, see "Defining IMS Connect systems" on page 320.

Displaying active sessions

To list active sessions in the IMS Connect Extensions ISPF dialog:
1. From the IMS Connect Extensions primary menu, 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.
3. To refresh the list at any time, press Enter. To view additional columns of information about sessions, scroll right using the Right function key (F11). To sort active sessions by a given column, tab to the column heading and press Enter.
4. To view additional details on a session, enter line action $ next to the session.
You can use the following fields to customize the display:

**Session Wait Time**
List only those sessions that have been waiting since the last event for longer than a given time (in seconds)

**Include persistent sockets**
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.

**Commands**
You can enter the following commands on the Command line:

**FORM** Change the order of columns or to perform additional customization. See “Customizing the Active Sessions panel” on page 138. You can also press the Form function key (F5) to perform this task.

**FILTER** Filter the display. See “Filtering active sessions” on page 136. You can also press the Filter function key (F6) to perform this task.
Line actions

You can perform the following line actions on an active session:

S  View information about the session. See “Viewing additional information for an active session.”

N  View network statistics (TCP/IP information similar to the NETSTAT command). See “Viewing TCP/IP information for a session” on page 135.

D  Drain (close) the active session after the next inbound message. Sessions may only be drained if they are persistent, do not have an active IMS conversation, and do not have active Resume TPIPE requests.

P  Stop the active session.

Related concepts:

“Active sessions” on page 209

Active sessions in the IMS Connect Extensions Operations Console allows you to monitor 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.

Related reference:

“Active session utility” on page 571

The active session utility assists in problem determination by analyzing an IMS Connect Extensions journal data set and listing details of active sessions.

Viewing additional information for an active session

You can view additional information on OTMA, ODBM, and MSC active sessions using the IMS Connect Extensions ISPF dialog.

About this task

The following figure is an example of a customized Active Sessions form showing ODBM activity.

Procedure

1. Display a list of active sessions. See “Active Sessions” on page 130.

---

Figure 57. The IMS Connect Extensions Active Sessions panel showing ODBM activity (customized view)
2. Next to the session you are interested in, enter line action S.

Tip: The 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 the OTMA example below, when it returns from the exit the session contains more information.

Example

The following figure shows an example of a detailed view of an OTMA session.

![OTMA Session Details](image)

**Figure 58. Viewing OTMA session details in the IMS Connect Extensions ISPF dialog**

The following figure shows an example of a detailed view of an ODBM session.
The following figure shows an example of a detailed view of an MSC session.

Figure 59. Viewing ODBM session details in the IMS Connect Extensions ISPF dialog

The following figure shows an example of a detailed view of an MSC session.
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

1. Display a list of active sessions. See “Active Sessions” on page 130.
2. Next to the session you are interested in, enter line action N.

Figure 60. Viewing MSC session details in the IMS Connect Extensions ISPF dialog
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 opens the Active Sessions Filter panel.

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
Less than
Less than or equal to
Greater than
Greater than or equal to
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
  UserID  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:
When you enter different criteria on different rows, an 'AND' relationship applies.

**Customizing the Active Sessions panel**

Use the Sessions Form panel in the IMS Connect Extensions ISPF dialog 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.

![Sessions Form panel](image)

Select from the following fields on the Sessions Form panel:

**System**

The IMS Connect system name.
Type  The session type: Open Transaction Manager Access (OTMA) or Open
       Database Manager (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.ccccccc
       (where ccccccc=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.ccccccc (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 Multiple Systems Coupling (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 hexadecimal 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 276.

Tasks
You can perform the following tasks from the Sessions Form panel.

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 restores the Active Sessions panel to its default values, 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. To remove the column, enter O next to the name of the column. The letters OMIT appear in the Usage column. If the form is on, the column will not appear.

2. To restore the field, enter O.

**Tip:** You can also type OMIT directly in the Usage column.

### 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

### 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.

---

**Datastore Monitor**

Use the Datastore Monitor in the IMS Connect Extensions ISPF dialog to manage and monitor your IMS data stores. Use the Datastore Monitor to view detailed information about an IMS data store, to suspend and resume routing of messages to an IMS data store using the drain feature, and to start and stop an IMS data store.

To display the Datastore Monitor in the IMS Connect Extensions ISPF dialog, complete the following steps:

1. From the IMS Connect Extensions primary menu, select option 2 Operations.
2. Enter line action **DM** next to a system or a group.

The Datastore Monitor panel provides additional options for managing IMS data stores.

![Datastore Monitor Panel](image)

**Figure 64. The Datastore Monitor panel in the IMS Connect Extensions ISPF dialog**

The Datastore Monitor provides the following additional status information:

**ICON status**

This field shows the status of the XCF connection to IMS Connect. **ICON status** can display one of the following values:

- **Active** The XCF connection is active.
- **Inactive** The XCF connection is inactive.
- **Discon** The XCF connection is disconnected.

**IMS Status**

This field shows the availability of an active IMS system for processing OTMA workloads.

- **Normal** The IMS system is available and is processing OTMA messages normally.
- **Degraded** IMS is processing OTMA messages slowly. OTMA issues a degraded state protocol command when one or more conditions indicate that IMS is not processing OTMA messages as quickly as it should. For more information, see *OTMA resource monitoring* in the IMS user documentation.

**MemberFloodWarn**

The IMS data store is in a Member Flood Warning state.

**MemberFloodSev**

The IMS data store is in a Member Flood Severe state.
GlobalFloodWarn
The IMS data store is in a Global Flood Warning state.

GlobalFloodSevr
The IMS data store is in a Global Flood Severe state.

Unavailable
IMS can no longer accept OTMA transactions for processing. OTMA issues the unavailable state protocol command to alert the OTMA client that one or more severe conditions prevent IMS from processing OTMA messages. For more information, see OTMA resource monitoring in the IMS user documentation.

Unknown
The target IMS system is V9 or earlier.

blank
IMS Connect is inactive or disconnected. No status is shown.

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 data store is potentially a candidate for routing.

NormalWLB
The data store is potentially a candidate for routing, and workload balancing is active.

Degraded
The data store 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 data store is not a candidate for routing. This could be due to a severe-level IMS status.

SusCWRZero
The data store has been suspended as a candidate for routing. Workload balancing is active and the data store has a capacity weight ratio of zero.

Suspended
The data store has been suspended as a candidate for routing. The data store has been drained without the autoresume option. The data store might be stopped (see ICON Status).

SusAutoRes
The data store has been suspended as a candidate for routing. It will automatically resume routing when the data store has been stopped and then becomes active again.

AutoRes or AutoResume
The data store is stopped but if it is started it will automatically become a candidate for routing.

Super Member
The OTMA super member name defined in your IMS Connect system configuration. The super member name is specified in the SMEMBER parameter in the DATASTORE statement in the HWSCFGxx member of the IMS PROCLIB data set.
CWR  The capacity weight rating (CWR) of the IMS data store relative to other IMS data stores.

The capacity weight rating is used by workload balancing to reroute incoming message to alternate IMS data stores. A value of zero indicates that the datastore is not currently a candidate for routing.

To learn more about setting the capacity weight rating for an IMS data store, see “Setting the capacity of an IMS data store for OTMA workloads” on page 254.

Waiting Reply  The number of OTMA sessions waiting for a reply from the IMS data store.

If IMS Connect Extensions in the target IMS Connect system does not support this field, all of the values in the Waiting Reply column will be spaces.

Line actions

T  Start the IMS data store.

P  Stop the IMS data store.

D and DA  Drain the IMS data store of in-progress transactions. The “drain” feature suspends routing of messages to an IMS data store. Once an IMS data store has been “drained” of in-progress transactions, you can perform maintenance tasks (such as an IMS shut down) without risk of causing interruptions to connected clients.

Important: The drain facility only works on messages handled by IMS Connect Extensions OTMA rules-based routing. It does not work on messages that circumvent a routing rule or with exits customized with their own routing mechanisms. To ensure that all of your workload is handled by IMS Connect Extensions, see “Rejecting transactions for data stores with no OTMA routing rule” on page 265.

To drain and then shut down an IMS system, perform the following steps:
1. Use line action D (or DA, see note below) to drain the IMS data stores associated with the IMS system you wish to shut down.
2. Monitor the values in the Waiting Reply column. When all IMS data stores have a Waiting Reply count of 0, use line action P to stop the IMS data stores.
3. To shut down the IMS system, issue the /CHECKPOINT (or /CHE) command.

Using D versus DA: Line action D suspends routing to the IMS data store. Line action DA performs the same actions as D, but will automatically resume routing to the selected IMS data store if that IMS data store is stopped and then is subsequently returned to service. To achieve this, IMS Connect Extensions uses the IMS Datastore Available event record (X’10’) to detect when the IMS data store is ready to return to service.

Tip: To resume routing to an IMS data store, use line action R.

R  Resume routing to the IMS data store.
For IMS data stores that have been drained of in-progress transactions and are in the **Suspended** or **SusAutoRes** states, the “resume” feature returns an IMS data store to service where it once again becomes a candidate for message routing.

**Related concepts:**

- “Suspending OTMA workload routing to IMS data stores using drain” on page 273
- “OTMA flood control processing in IMS Connect Extensions” on page 267

The drain facility in IMS Connect Extensions temporarily suspends OTMA rules-based routing of messages to a data store (or a set of data stores), 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.

IMS Connect Extensions OTMA rules-based routing can be used to avoid message flood conditions in IMS.

**Related tasks:**

- “Managing IMS data stores” on page 204
- “Dynamically adding IMS data stores as targets for OTMA routing” on page 274

Use the **Datastores** and **Datastore Groups** tab in Status Monitor to perform management tasks on IMS data stores from the IMS Connect Extensions Operations Console. Use this facility to start and stop IMS data stores, suspend and resume routing to a data store, and to dynamically update a data stores capacity weight rating (CWR) in response to periods of changing demand.

**Issuing IMS Connect Extensions commands from the Commands panel**

Use the Commands panel in the IMS Connect Extensions ISPF dialog to generate and then issue IMS Connect Extensions commands for managing user exits, rebuilding in-memory system and rule definitions from the IMS Connect Extensions repository, reloading SAF class rules, temporarily overriding settings in IMS Connect Extensions definitions, activating OTMA and ODBM routing plans, and switching the IMS Connect Extensions journal.

**Before you begin**

Activate command access. See “Activating command access to an IMS Connect system” on page 235.

**Procedure**

1. From the IMS Connect Extensions primary option menu, select option 2 **Operations**.
2. Set **View** to option 2 **Systems**.
3. To open the Commands panel, enter line action CX next to the desired system.
The Commands panel is displayed. The system you selected is displayed in the System field.

![Image of the Commands panel]

**Figure 65. Opening the Commands panel using line action CX in the Operations dialog**

4. Select one of the following options by entering its associated number:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Exits</strong></td>
<td>Display the User Exit Commands panel, used to 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. See &quot;Managing user exits&quot; on page 147.</td>
</tr>
<tr>
<td><strong>Refresh</strong></td>
<td>Display the Refresh Commands panel, used to refresh IMS Connect Extensions in-memory definitions for the selected system from definitions stored in the IMS Connect Extensions repository. See &quot;Refreshing in-memory definitions from the IMS Connect Extensions repository&quot; on page 150.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Security</td>
<td>Display the Security Commands panel, used to refresh SAF class rules or clear security credentials from the ACEE cache based on a specified user ID or pattern.</td>
</tr>
<tr>
<td></td>
<td>See “Issuing security commands” on page 155.</td>
</tr>
<tr>
<td>Set</td>
<td>Display the Set Commands panel, used to set IMS Connect Extensions in-memory definition settings for the selected system. Use this panel to activate IMS Connect Extensions conditional tracing and to activate OTMA and ODBM routing plans.</td>
</tr>
<tr>
<td></td>
<td>See “Setting in-memory IMS Connect Extensions definitions” on page 156.</td>
</tr>
<tr>
<td>Journal</td>
<td>Display the Command Processor panel, used to switch the active IMS Connect Extensions journal data set or refresh some IMS Connect Extensions journal options.</td>
</tr>
<tr>
<td></td>
<td>See “Managing the IMS Connect Extensions journal” on page 163.</td>
</tr>
<tr>
<td>Datastores</td>
<td>Display the Datastore Monitor panel.</td>
</tr>
<tr>
<td></td>
<td>See “Datastore Monitor” on page 141.</td>
</tr>
</tbody>
</table>

### Managing user exits

Use the User Exit Commands panel to add, delete, enable, disable, and reload user exits without interrupting the execution of IMS Connect. To access this ISPF panel, select option 1 Exits on the Commands panel.

The User Exit Commands panel lists exits that are already loaded. The Status field displays the current status of the user exit. A value of Active indicates that the user exit is active and processing incoming messages.

![User Exit Commands panel](image)

**Figure 67. The User Exit Commands panel in the IMS Connect Extensions ISPF dialog**
You can manage user exits in the following ways:

**Adding a user exit**

To load a new user exit executable, perform the following steps:

1. Enter the `ADD` command on the **Command** line. A list of defined exits is displayed.
2. To select the user exit, enter line action S next to the name of the exit.

To define a new user exit, see “Defining user exits” on page 336.

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 not already known to IMS Connect or the exit is not found in the STEPLIB data set concatenation, the `ADD` command 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:**

- If the new user exit uses Message IDs that are used by existing user exits, the `ADD` command is rejected.
- IMS Connect Extensions does not automatically update the IMS Connect configuration member with the name of the user exit. You must do this manually.
- 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.

**Reload a user exit**

To reload a new copy of the user exit executable, complete the following steps:

1. Enter line action / (slash) next to the user exit you wish to reload.
2. Select option 1 **Reload**. The Command Processor panel is displayed.
3. Select all of the commands displayed on the Command Preprocessor panel by entering line action / next to each command.

4. Press Enter to submit the request. The result for each command is displayed in the **Response** column. A response of 0000 indicates that the command was processed successfully.

**Note:** 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.

Deleting a user exit
To delete the association between the MSG IDs and the user exit, complete the following steps:
1. Enter line action / (slash) next to the user exit you wish to delete.
2. Select option 2 Delete.
3. Select all of the commands displayed on the Command Preprocessor panel by entering line action / next to each command.
4. Press Enter to submit the request. The result for each command is displayed in the Response column. A response of 0000 indicates that the command was processed successfully.

When a user exit has been deleted, all messages for those MSG IDs will be rejected by IMS Connect Extensions.

Note: Deleting a user exit does not physically delete it 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.

Disabling a user exit
To suspend processing for a user exit, complete the following steps:
1. Enter line action / (slash) next to the user exit you wish to disable.
2. Select option 3 Disable.
3. Select all of the commands displayed on the Command Preprocessor panel by entering line action / next to each command.

4. Press Enter to submit the request. The result for each command is displayed in the Response column. A response of 0000 indicates that the command was processed successfully.

IMS Connect Extensions will reject incoming messages for those MSG IDs supported by the user exit. Messages for XMIT or EXER are supported. If the user exit is not in a disabled state, the command is rejected.

Enabling a user exit
To enable a previously disabled user exit and re-associate MSG IDs with the user exit, complete the following steps:
1. Enter line action / (slash) next to the user exit you wish to enable.
2. Select option 4 Enable.
3. Select all of the commands displayed on the Command Preprocessor panel by entering line action / next to each command.
4. Press Enter to submit the request. The result for each command is displayed in the Response column. A response of 0000 indicates that the command was processed successfully.

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 in the system definition, the user exits may begin to receive messages.

Refreshing in-memory definitions from the IMS Connect Extensions repository
Use the Refresh Commands panel to reload IMS Connect Extensions in-memory definitions for the selected system from definitions stored in the IMS Connect Extensions repository. To access this ISPF panel, select option 2 Refresh on the Commands panel.

The Refresh Commands panel displays a list of definitions that can be reloaded from the IMS Connect Extensions repository.
To refresh a set of definitions, complete the following steps:

1. For IMS Connect systems definitions and routing rules, simply type the associated number and then press Enter. For all other types, use the Prompt function key (F4) first to select a definition of that type and then press Enter. These definitions can only be refreshed for the system on which the command will run.

2. Select all of the commands displayed on the Command Preprocessor panel by entering line action / next to each command.

3. Press Enter to submit the request. The result for each command is displayed in the Response column. A response of 0000 indicates that the command was processed successfully.

   **Note:** Values will be reloaded from the definition in the IMS Connect Extensions repository regardless of whether they have been changed. Any changes previously made by setting override values will be lost. See "Setting in-memory IMS Connect Extensions definitions" on page 156.

4. Use the message log to view the results of the refresh. See "Browsing the message log for an IMS Connect" on page 164.

Use the following information to determine which values in each definition will be refreshed.

**Note:** All definitions in the IMS Connect Extensions repository are refreshed on an IMS Connect system restart, even if they are not listed in the section below.

### 1 System Definition

The **System Definition** option reloads values from the IMS Connect Extensions repository for the following fields in the **System Definition** definition:

- Message recall count
- Activate Mixed Case Passwords
- Collection level
- Activate Pacing
- Interval count
- Warning threshold
- Reject threshold
- Activate Session Message Limit

![Figure 69. The Refresh Commands panel in the IMS Connect Extensions ISPF dialog](image-url)
• Limit threshold
• Activate Security
• Activate ACEE Cache
• Ageing interval
• Activate validation
• Validation type
• Security class
• Activate IP Address Rules
• Activate OTMA rules
• Activate ODBM rules
• Activate Transaction Routing
  • The list of applications defined in the System Definition - Applications panel in the Define Applications for option.
• Activate Workload Balancing
• Activate Statistics Collection

For a description of each field, see “Defining IMS Connect systems” on page 320.

Tip: To refresh IMS Connect Extensions journal options, see “Managing the IMS Connect Extensions journal” on page 163.

Important:
1. Options for pacing, security, transaction routing, workload balancing and statistics collection will not be activated if the IMS Connect system was started with the Activate Advanced Features field set to 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.

2 Datastore
The Datastore option reloads values from the IMS Connect Extensions repository for the following fields in the Datastore definition:
• Datastore Group
• Use as Primary Datastore for IMS Connect System
• Activate Pacing
• Warning threshold
• Reject threshold
• Routing options (including Also route Send Only requests and Route Resume TPIPE requests to Affinity List)
• Capacity weight rating

For a description of each field, see “Defining IMS data stores” on page 339.

Important:
IMS data stores are only available to be refreshed if they are defined both in the IMS Connect Extensions repository and in the IMS Connect configuration member.

If the Datastore Group name, Transaction Routing options or Capacity weight rating were changed in the data store definition in the repository, IMS Connect Extensions will rebuild the Candidate List, updating the Transaction Routing definitions.

You cannot add a new data store using the Datastore option. To add a new data store, see “Dynamically adding IMS data stores as targets for OTMA routing” on page 274.

3 Datastore Group
The Datastore Group option reloads values from the IMS Connect Extensions repository for all fields in the Datastore Group definition:

- Activate pacing
- Warning threshold
- Reject threshold

For a description of each field, see “Defining groups of IMS data stores” on page 344.

Important:
- 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 group using the Datastore Group option. To add a new group, use option 2 Datastores and ensure that your new group is referenced by an IMS data store definition that was not previously defined when IMS Connect Extensions was started.

4 Affinity List
The Affinity List option reloads the specified affinity list definition. The list of data stores defined in the affinity list will be rebuilt in memory, replacing the previous definition. See “Defining affinity lists” on page 346.

Important:
- Affinity lists are only available to be refreshed if they contain an IMS data store that is defined in the IMS Connect configuration member.
- 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. To add a new affinity list, it must be referenced by at least one IMS data store, application, or transaction definition that was not previously defined when IMS Connect Extensions was started. Once the reference is in place, use option 1 Datastore, 5 Application, or 6 Transaction to refresh the definition as appropriate.

5 Application
The Application option reloads values from the IMS Connect Extensions repository for all fields in the Application definition:

- Activate Transaction Routing
- Values specified in the Route transactions to option
- Values specified in the Routing Error processing option

For a description of each field, see “Defining applications” on page 347.
Important:

- Applications are only available to be refreshed if they are included in the IMS Connect Extensions system definition.
- Refreshing an application definition 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. To add a new application, use option 1 System Definition and ensure that your new application is referenced by an IMS Connect system definition.

6 Transaction

The Transaction option reloads values from the IMS Connect Extensions repository for all fields in the Transaction definition:

- Application
- Override Transaction Timer
- Message timeout
- ACK/NAK timeout
- Override Transaction Expiration
- Set F1_TRNEXP
- Override Client ID Cancellation
- Set F3_CANCID
- Activate Transaction Routing
- Values specified in the Route transactions to option
- Values specified in the Routing Error processing option

For a description of each field, see “Defining transactions” on page 349.

Users can also use this option to add a new transaction.

Important: If the Application name or Transaction Routing options have been changed in the transaction definition, IMS Connect Extensions will rebuild the Candidate List, updating the Transaction Routing definitions.

7 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 IMS Connect Extensions repository. See “Defining OTMA routing rules” on page 356.

Tip: Routing rules are written to the IMS Connect Extensions log (CEXPRINT) at system startup and after OTMA routing rules are refreshed. See Chapter 12, “OTMA workload routing in IMS Connect,” on page 245.

Important:

- If you change the active OTMA routing plan you must then refresh the OTMA routing rules to load the new set of rules. See “Setting in-memory IMS Connect Extensions definitions” on page 156.
- 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 using option 1 System Definition and then refresh the routing rules.

8 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 IMS Connect Extensions repository. See “Defining ODBM routing rules” on page 364.

Tip: Routing rules are written to the IMS Connect Extensions log (CEXPRINT) at system startup and after ODBM routing rules are refreshed. See Chapter 14, “ODBM workload routing in IMS Connect,” on page 279.

Important:
- If you change the active ODBM routing plan you must then refresh the ODBM routing rules to load the new set of rules. See “Setting in-memory IMS Connect Extensions definitions” on page 156.
- If you have changed another definition that would affect rules-based routing behavior for ODBM 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 ODBM rules-based routing, you must first refresh the system definition using option 1 System Definition and then refresh the routing rules.

9 IP Address Rules
The IP Address Rules option rebuilds all of the system internal IP address rules for all the IP address rule definitions in the IMS Connect Extensions repository. See “Creating workload rules for specific IP addresses” on page 305.

Related reference:
“Setting in-memory IMS Connect Extensions definitions” on page 156

Use the Set Commands panel to set in-memory IMS Connect Extensions definition settings for the selected system. To access this ISPF panel, select option 4 Set on the Commands panel.
“REFRESH command” on page 449

The REFRESH host command for REXX rebuilds the in-memory copy of the selected definitions in the target IMS Connect system from the definitions stored in your IMS Connect Extensions repository.

Issuing security commands
Use the Security Commands panel to refresh SAF Class rules and to delete any cached User ID security profiles. To access this ISPF panel, select option 3 Security on the Commands panel.

The Security Commands panel displays the security commands that can be issued to the selected IMS Connect system.
The Security Commands panel has 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.

To refresh SAF Class rules, complete the following steps:
1. Enter line action / (slash) next to Refresh SAF Class rules.
2. Select the command displayed on the Command Preprocessor panel by entering line action / next to the command.
3. Press Enter to submit the request. The result is displayed in the Response column. A response of 0000 indicates that the command was processed successfully.

**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.

To clear specified ACEE cache User ID elements, complete the following steps:
1. Enter a User ID or prefix in the space next to the Clear ACEE for User ID (or prefix) option. If you use a prefix, 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
2. Enter line action / (slash) next to Clear ACEE for User ID (or prefix).
3. Select the command displayed on the Command Preprocessor panel by entering line action / next to the command.
4. Press Enter to submit the request. The result is displayed in the Response column. A response of 0000 indicates that the command was processed successfully.

**Setting in-memory IMS Connect Extensions definitions**
Use the Set Commands panel to set in-memory IMS Connect Extensions definition settings for the selected system. To access this ISPF panel, select option 4 Set on the Commands panel.
For IMS Connect systems definitions, IMS Connect Extensions tracing, and routing plans, simply type the associated number and then press Enter.

For all other types, use the Prompt function key (F4) first to select a definition of that type and then press Enter. This list only includes definitions that can be set for the system on which the command will run.

The Set Commands panel lists the following options:

1 **System Definition**
   To temporarily override IMS™ Connect Extensions in-memory definition settings for an IMS Connect system, select option 1. See “Overriding IMS Connect system settings” on page 158.

2 **Tracing**
   To configure and activate IMS Connect Extensions trace, select option 2. For more information on the trace and a complete list of tracing options, see “Using the IMS Connect Extensions trace” on page 93.

3 **Datastore**
   To temporarily override IMS Connect Extensions in-memory definition settings for an IMS data store definition, use the Prompt function key (F4) to select a definition and then select option 3. See “Overriding IMS data store definitions” on page 159.

   **Note:** An IMS data store is only listed if it is defined in both the IMS Connect Extensions repository and in the IMS Connect configuration member.

4 **Datastore Group**
   To temporarily override IMS Connect Extensions in-memory definition settings for an IMS data store group definition, use the Prompt function key (F4) to select a definition and then select option 4. See “Overriding IMS data store group definitions” on page 160.

   **Note:** Datastore groups are only listed if they contain at least one IMS data store that is defined in both the IMS Connect Extensions repository and in the IMS Connect configuration member.

5 **Application**
   To temporarily override IMS Connect Extensions in-memory definition
settings for an application definition, use the Prompt function key (F4) to select a definition and then select option 5. See “Overriding application definitions” on page 161.

6 Transaction
To temporarily override IMS Connect Extensions in-memory definition settings for a transaction definition, use the Prompt function key (F4) to select a definition and then select option 6. See “Overriding transaction definitions” on page 162.

7 Routing Plan
To activate a new routing plan or to switch plans, select option 7. See “Activating OTMA routing rules in IMS Connect” on page 259 and “Activating ODBM routing rules in IMS Connect” on page 294.

Related reference:
“Refreshing in-memory definitions from the IMS Connect Extensions repository” on page 150

Use the Refresh Commands panel to reload IMS Connect Extensions in-memory definitions for the selected system from definitions stored in the IMS Connect Extensions repository. To access this ISPF panel, select option 2 Refresh on the Commands panel.

Overriding IMS Connect system settings
Use the Set System Definition panel to temporarily override IMS Connect Extensions in-memory definition settings for an IMS Connect system. The IMS Connect system definition in the IMS Connect Extensions repository remains unchanged. To access this ISPF panel, select option 1 System Definition on the Set Commands panel.

The Set System Definition panel displays the current in-memory values for the selected IMS Connect system in IMS Connect Extensions.
To temporarily override the in-memory values for the selected system, complete the following steps:

1. Review the settings on the panel and make changes as desired. For a description of each field, see “Defining IMS Connect systems” on page 320. When you are finished making changes, press Enter. The Command Preprocessor panel is displayed.

2. Select all of the commands displayed on the Command Preprocessor panel by entering line action / next to each command.

3. Press Enter to submit the request. The result for each command is displayed in the Response column. A response of 0000 indicates that the command was processed successfully.

Notes:

1. If the IMS Connect system started with event collection inactive then you cannot change the collection level.

2. If the IMS Connect system was started with advanced features inactive, then none of the advanced features can be overridden.

Overriding IMS data store definitions

Use the Set Datastore panel to temporarily override IMS Connect Extensions in-memory definition settings for an IMS data store. The IMS data store definition in the IMS Connect Extensions repository remains unchanged. To access this ISPF panel, select option 3 Datastore on the Set Commands panel.

The Set Datastore panel displays the current in-memory values for the selected IMS data store definition in IMS Connect Extensions.
To temporarily override the in-memory values for the selected IMS data store, complete the following steps:

1. Review the settings on the panel and make changes as desired. For a description of each field, see "Defining IMS data stores" on page 339. When you are finished making changes, press Enter. The Command Preprocessor panel is displayed.

2. Select all of the commands displayed on the Command Preprocessor panel by entering line action / next to each command.

3. Press Enter to submit the request. The result for each command is displayed in the Response column. A response of 0000 indicates that the command was processed successfully.

**Overriding IMS data store group definitions**

Use the Set Datastore Group panel to temporarily override IMS Connect Extensions in-memory definition settings for a group of IMS data stores. The group definition in the IMS Connect Extensions repository remains unchanged. To access this ISPF panel, select option 4 Datastore Group on the Set Commands panel.

The Set Datastore Group panel displays the current in-memory values for the selected group in IMS Connect Extensions.

**Figure 73. The Set Datastore panel in the IMS Connect Extensions ISPF dialog**

To temporarily override the in-memory values for the selected group, complete the following steps:

1. Review the settings on the panel and make changes as desired. For a description of each field, see "Defining groups of IMS data stores" on page 344. When you are finished making changes, press Enter. The Command Preprocessor panel is displayed.
2. Select all of the commands displayed on the Command Preprocessor panel by entering line action / next to each command.

3. Press Enter to submit the request. The result for each command is displayed in the Response column. A response of 0000 indicates that the command was processed successfully.

**Overriding application definitions**

Use the Set Application panel to temporarily override IMS Connect Extensions in-memory definition settings for an application definition. The application definition in the IMS Connect Extensions repository remains unchanged. To access this ISPF panel, select option 5 Application on the Set Commands panel.

The Set Application panel displays the current in-memory values for the selected application in IMS Connect Extensions.

```
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 75. The Set Application panel in the IMS Connect Extensions ISPF dialog*

To temporarily override the in-memory values for the application, complete the following steps:

1. Review the settings on the panel and make changes as desired. For a description of each field, see "Defining applications" on page 347. When you are finished making changes, press Enter. The Command Preprocessor panel is displayed.

2. Select all of the commands displayed on the Command Preprocessor panel by entering line action / next to each command.

3. Press Enter to submit the request. The result for each command is displayed in the Response column. A response of 0000 indicates that the command was processed successfully.

**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.

**Overriding transaction definitions**

Use the Set Transaction panel to temporarily override IMS Connect Extensions in-memory definition settings for a transaction definition. The transaction definition in the IMS Connect Extensions repository remains unchanged. To access this ISPF panel, select option 6 **Transaction** on the Set Commands panel.

The Set System Definition panel displays the current in-memory values for the selected transaction definition in IMS Connect Extensions.

![Set Transaction panel](image)

**Figure 76. The Set Transaction panel in the IMS Connect Extensions ISPF dialog**

To temporarily override the in-memory values for the selected transaction, complete the following steps:

1. Review the settings on the panel and make changes as desired. For a description of each field, see ["Defining transactions" on page 349](#). When you are finished making changes, press Enter. The Command Preprocessor panel is displayed.

2. Select all of the commands displayed on the Command Preprocessor panel by entering line action / next to each command.

3. Press Enter to submit the request. The result for each command is displayed in the **Response** column. A response of 0000 indicates that the command was processed successfully.

**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 349

The Transactions panel allows you to define transactions to be routed by IMS Connect Extensions through the legacy transaction routing feature. To access this ISPF panel, select option 1.7 *Transactions* from the IMS Connect Extensions primary menu.

## Managing the IMS Connect Extensions journal

Use option 5 *Journal* on the Commands panel to dynamically switch the active IMS Connect Extensions journal and refresh journal settings from the repository.

```
<table>
<thead>
<tr>
<th>Command</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>/ JOURNAL SWITCH</td>
<td>HWS1 JOURNAL SWITCH</td>
</tr>
<tr>
<td>- JOURNAL REFRESH</td>
<td>HWS1 JOURNAL REFRESH</td>
</tr>
</tbody>
</table>
```

Figure 77. The Command Processor panel showing commands for an IMS Connect Extensions journal switch

To select one or more commands for processing, enter line action / next to each command you are interested in. To issue those commands, press Enter. The following commands can be issued from this panel:

### JOURNAL SWITCH
Switches the current active journal to the next journal.

If the *Activate journal archiving* option is enabled in the system definition, IMS Connect Extensions will start a new archive job and archive the current active journal. See “Defining active journal data set processing options” on page 329.

### JOURNAL REFRESH
Reload IMS Connect Extensions in-memory journal settings from the system definition stored in the IMS Connect Extensions repository. The following values are reloaded into memory:

- In the Active Journal Data Set panel:
  - Journal full option
  - Job Statement Information
  - Archive JCL skeleton
- In the Archive Journal Data Set panel:
  - Data Set name
  - Device type

See “Defining active journal data set processing options” on page 329 and “Defining the archive journal data set template” on page 332.
Tip: You can change other archive journal options by modifying the archive JCL skeleton. See “Creating an archive JCL skeleton” on page 63.

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 tasks:

“Using the Operations dialog” on page 120
Use the primary menu of the IMS Connect Extensions ISPF dialog to access the Operations dialog.

Related reference:

“SWITCH command” on page 456
The SWITCH host command for REXX switches the active journal for an IMS Connect system.

---

**Browsing the message log for an IMS Connect**

The message log for an IMS Connect system reveals activity within IMS Connect and IMS Connect Extensions. Use the message log to understand more about the status of archiving, to view error messages, or to display a history of commands that have been issued and their responses.

**Opening the message log**

To open the message log for an IMS Connect system, use one of the following methods:

- In ISPF screens that list an IMS Connect system or one of its components, enter line action **L**.
- In ISPF screens that show information about an IMS Connect, type **LOG** and press Enter.

**Navigating the message log**

The following figure shows a message log for IMS Connect system RAPI04. To navigate the message log, use standard ISPF Browse commands. To search for a text string within the log, enter **FIND string** on the **Command** line.
Issuing IMS Connect and IMS type-1 commands from the CONSOLE dialog

Users who have the appropriate level of authority can use the IMS Connect Extensions ISPF command shell to issue IMS Connect WTOR commands, IMS Connect z/OS commands, and IMS type-1 commands from IMS Connect Extensions.
Before you begin

Activate command access. See “Activating command access to an IMS Connect system” on page 235.

About this task

For more information on the commands that can be issued using the IMS Connect Extensions command shell, review the following topics in the IBM IMS documentation collection in the IBM Knowledge Center:

- For IMS Connect WTOR commands, see the topic “IMS Connect WTOR commands”.
- For IMS Connect z/OS commands, see the topic “IMS Connect z/OS commands”.
- For IMS type-1 commands, see the topic “IMS Commands”.

Procedure

1. From the IMS Connect Extensions primary option menu, select option 2 Operations.

2. To start the shell, enter line action $# next to the IMS Connect system. The system you have selected (and its description, if available) is displayed at the top of the CONSOLE dialog and provides the context for subsequent commands you will issue.

3. The method of issuing the command depends on its type:
   - **To enter an IMS Connect WTOR or IMS Connect z/OS command**, type a slash (/) followed by the command you wish to issue. To issue the command, press Enter.
   - **To enter an IMS type-1 command**, you must first change the system context to an IMS data store. To do this, complete the following steps:
     a. Type a slash (/) and then press Enter.
     b. In the field beneath the text **Enter system and command below**, type the name of the IMS data store, or place the cursor on the field and press PF4 for a list of options.
c. Press PF3 to return to the CONSOLE dialog. The IMS data store you selected is displayed at the top of the dialog.

d. Type your IMS type-1 command and then press Enter. Be sure to include the slash (/) when formulating your command.

Tip: Use the command shell to query the status of IMS Connect. Many IMS Connect WTOR commands have equivalent IMS Connect z/OS commands and you can use these commands to query the status of IMS Connect. For example, to display detailed information on TCP/IP port 3702, you could issue IMS Connect WTOR command /VIEWPORT 3702, or you could issue IMS Connect z/OS command /QUERY PORT NAME(3702). For additional command suggestions, press PF1.

4. Use the following features of the IMS Connect Extensions command shell to continue your analysis:

**FIND string and RFIND**
Search the output for a string.

**CLEAR**
Clear the display.

**HEX ON/OFF**
Enable/disable hexadecimal output.

**/**
Show command history or change the system context.

**Example commands and resulting output**

**/VIEWHWS**

The following figure shows an example of issuing the /VIEWHWS IMS Connect WTOR command.
The following figure shows an example response from issuing the /QUERY PORT NAME (22941) IMS Connect z/OS command.
The following figure shows an example response from issuing the /DISPLAY ACTIVE IMS type-1 command.

```
08.36.16 MOBICON CEX5053I /DISPLAY ACTIVE
08.36.16 MOBICON f MOBICON,QUERY PORT NAME(22941) PORT=22941 STATUS=ACTIVE KEEPAV=500 NUMSOC=1 EDIT=TIMEOUT=0
08.36.16 MOBICON NO ACTIVE CLIENTS
08.36.16 MOBICON CEX5055I MOBICON Command complete
```

Figure 83. Example: Issuing an IMS Connect z/OS command to IMS Connect system MOBICON from the IMS Connect Extensions command shell

/DISPLAY ACTIVE

The following figure shows an example response from issuing the /DISPLAY ACTIVE IMS type-1 command.

```
01.14.00 IMSA CEX5053I /DISPLAY ACTIVE
01.14.00 IMSA REGID JOBNAME TYPE TRAN/STEP PROGRAM CLASS
01.14.00 IMSA 3 IEDAMPP2 TP VIPTRAN1 VIPPGM01 WAIT-MESSAGE 1
01.14.00 IMSA 2 IEDAMPP3 TP WAITING 1
01.14.00 IMSA 1 IEDAMPP1 TP WAITING 1
01.14.00 IMSA JMPRGN JMP NONE
01.14.00 IMSA JBPREG JBP NONE
01.14.00 IMSA BATCHREG BMP NONE
01.14.00 IMSA FPRGN FP NONE
01.14.00 IMSA DBTRGN DBT NONE
01.14.00 IMSA IEDADBRC DBRC
01.14.00 IMSA IEDADLIS DLS
01.14.00 IMSA VTAM ACB OPEN -LOGONS ENABLED
01.14.00 IMSA IMSLU=N/A N/A APPC STATUS=DISABLED TIMEOUT=0 MAXC=5000
01.14.00 IMSA OTMA GROUP=XCFGDEVTEST STATUS=ACTIVE
01.14.00 IMSA APPC/OTMA SHARED QUEUE STATUS - LOCAL=INACTIVE GLOBAL=INACTIVE
01.14.00 IMSA APPC/OTMA RRS MAX TCBS - 40 ATTACHED TCBS - 2 QUEUED RRSKWS- 0
01.14.00 IMSA APPLID=IEDAETV1 GRSNAME= STATUS=DISABLED
01.14.00 IMSA TCPIP.GENIMSID=GIMS STATUS=ACTIVE
01.14.00 IMSA LINE ACTIVE-IN - 1 ACTIV-OUT 0
01.14.00 IMSA NODE ACTIVE-IN - 0 ACTIV-OUT 0
01.14.00 IMSA LINK ACTIVE-IN - 0 ACTIV-OUT 0
01.14.00 IMSA +18276/091400*
01.14.00 IMSA CEX5055I HWSOPGSI Command complete
```

Figure 84. Example: Output from issuing the /DISPLAY ACTIVE IMS type-1 command to IMS system IMSA from the IMS Connect Extensions command shell

/DISPLAY TMEMBER ALL

The following figure shows an example response from issuing the /DISPLAY TMEMBER ALL IMS type-1 command.

```
01.14.00 IMSA CEX5053I /DISPLAY TMEMBER ALL
01.14.00 IMSA REGID JOBNAME TYPE TRAN/STEP PROGRAM CLASS
01.14.00 IMSA 3 IEDAMPP2 TP VIPTRAN1 VIPPGM01 WAIT-MESSAGE 1
01.14.00 IMSA 2 IEDAMPP3 TP WAITING 1
01.14.00 IMSA 1 IEDAMPP1 TP WAITING 1
01.14.00 IMSA JMPRGN JMP NONE
01.14.00 IMSA JBPREG JBP NONE
01.14.00 IMSA BATCHREG BMP NONE
01.14.00 IMSA FPRGN FP NONE
01.14.00 IMSA DBTRGN DBT NONE
01.14.00 IMSA IEDADBRC DBRC
01.14.00 IMSA IEDADLIS DLS
01.14.00 IMSA VTAM ACB OPEN -LOGONS ENABLED
01.14.00 IMSA IMSLU=N/A N/A APPC STATUS=DISABLED TIMEOUT=0 MAXC=5000
01.14.00 IMSA OTMA GROUP=XCFGDEVTEST STATUS=ACTIVE
01.14.00 IMSA APPC/OTMA SHARED QUEUE STATUS - LOCAL=INACTIVE GLOBAL=INACTIVE
01.14.00 IMSA APPC/OTMA RRS MAX TCBS - 40 ATTACHED TCBS - 2 QUEUED RRSKWS- 0
01.14.00 IMSA APPLID=IEDAETV1 GRSNAME= STATUS=DISABLED
01.14.00 IMSA TCPIP.GENIMSID=GIMS STATUS=ACTIVE
01.14.00 IMSA LINE ACTIVE-IN - 1 ACTIV-OUT 0
01.14.00 IMSA NODE ACTIVE-IN - 0 ACTIV-OUT 0
01.14.00 IMSA LINK ACTIVE-IN - 0 ACTIV-OUT 0
01.14.00 IMSA +18276/091400*
01.14.00 IMSA CEX5055I HWSOPGSI Command complete
```

Figure 85. Example: Output from issuing the /DISPLAY TMEMBER ALL IMS type-1 command to IMS system IMSA from the IMS Connect Extensions command shell
Use the Connect Commands tab in the IMS Connect Extensions Operations Console to issue IMS Connect WT or commands and IMS Connect z/OS command.

Use the IMS Commands tab in the IMS Connect Extensions Operations Console to issue IMS type-1 commands directly to an IMS data store.

The SHELL host command for REXX runs an IMS Connect command or IMS type-1 command on a specified target system or IMS data store. You can use this to automate some procedures such as stopping all the IMS data stores associated with an IMS system across multiple IMS Connect instances.

**Related tasks:**

- “Issuing IMS Connect WT or commands and IMS Connect z/OS commands” on page 212
- “Issuing IMS type-1 commands” on page 215

**Related reference:**

- “SHELL command” on page 454

**IMS Connect Extensions publisher API client list**

Use the IMS Connect Extensions publisher API client list to list summary information about client applications that are connected to the IMS Connect Extensions publisher API.

**Before you begin**

Use the Activate Publisher API option to enable the IMS Connect Extensions publisher API. See “Defining IMS Connect systems” on page 320.

**Procedure**

To list active sessions in the IMS Connect Extensions ISPF dialog:

1. From the IMS Connect Extensions primary menu, select option 2 Operations.
2. Enter **PU** next to a system or a group. The Publisher Status - Client List panel is displayed.

<table>
<thead>
<tr>
<th>Thread</th>
<th>Client</th>
<th>AuthID</th>
<th>Collection Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>EG AGENT1</td>
<td>USER1</td>
<td>SYNC(512)</td>
</tr>
<tr>
<td>2</td>
<td>EG AGENT2</td>
<td>USER2</td>
<td>SYNC(4K) ASYNC(1,75%,8M)</td>
</tr>
<tr>
<td>3</td>
<td>EG AGENT3</td>
<td>USER3</td>
<td>SYNC(4K) ASYNC(2,60%,512K)</td>
</tr>
<tr>
<td>4</td>
<td>EG AGENT4</td>
<td>USER4</td>
<td>SYNC(4K) ASYNC(3,50%,512K)</td>
</tr>
<tr>
<td>5</td>
<td>EG AGENT5</td>
<td>USER5</td>
<td>SYNC(1K) ASYNC(4,50%,512K)</td>
</tr>
<tr>
<td>6</td>
<td>EG AGENT6</td>
<td>USER6</td>
<td>SYNC(4K) ASYNC(0,50%,4K)</td>
</tr>
<tr>
<td>7</td>
<td>EG AGENT7</td>
<td>USER7</td>
<td>SYNC(4K) ASYNC(1,50%,512K)</td>
</tr>
</tbody>
</table>

**Figure 86. The IMS Connect Extensions Publisher Status - Client List panel**

The following fields are displayed on the panel:

**Publisher status**

The status of the publisher API server. The status can be one of the following values:

**LOGON**

Accepting client connections.

**NOLOGON**

Rejecting new client connections because Maximum clients is set to zero. See “Setting the maximum number of publisher API clients” on page 172.

**SHUTDOWN**

IMS Connect is shutting down.

**Collection mode**

The data collection mode. The following information is displayed:

**SYNC(nnnn)**

Size of the synchronous buffer in kilobytes.

**ASYNC(c,xx, nnnn)**

Parameters from the STARTMON function:

- **c** Event collection level (LEVEL).
- **xx** Buffer-full threshold percentage (THRESH).
- **nnnn** Size of the STARTMON buffer (BUFSIZE).

See “STARTMON” on page 475.

3. To view additional details on a client, enter line action **S** next to the client.
Many fields on the Client Detail panel correspond to publisher API functions.

<table>
<thead>
<tr>
<th>Field</th>
<th>API function</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client name</td>
<td>CONNECT</td>
<td>AGENT</td>
</tr>
<tr>
<td>Authorization ID</td>
<td>CONNECT</td>
<td>AUTHID</td>
</tr>
<tr>
<td>Thread number</td>
<td>Generated by IMS Connect Extensions</td>
<td>N/A</td>
</tr>
<tr>
<td>Data collection mode</td>
<td>SYNC/ASYNC when STARTMON established, otherwise SYNC</td>
<td>N/A</td>
</tr>
<tr>
<td>Buffer size (synchronous monitoring)</td>
<td>CONNECT</td>
<td>BUFSIZE</td>
</tr>
<tr>
<td>Buffer size (asynchronous monitoring)</td>
<td>STARTMON</td>
<td>BUFSIZE</td>
</tr>
<tr>
<td>Collection level</td>
<td>STARTMON</td>
<td>LEVEL</td>
</tr>
<tr>
<td>Buffer threshold</td>
<td>STARTMON</td>
<td>THRESH</td>
</tr>
<tr>
<td>Client status</td>
<td>Generated by IMS Connect Extensions</td>
<td>-</td>
</tr>
</tbody>
</table>

Related concepts:
Chapter 20, “IMS Connect Extensions publisher API,” on page 463
Use the IMS Connect Extensions publisher API to externalize IMS Connect Extensions and IMS Connect data to third-party applications.

Related tasks:
“Starting an IMS Connect Extensions feed” on page 100
To start an IMS Connect Extensions feed, you run a job that specifies a destination and which IMS Connect systems to use as the feed source.

**Setting the maximum number of publisher API clients**

There are several ways to limit the number of clients that may connect to the publisher API. If the maximum number of clients is reached, new connections are rejected.
Procedure

To dynamically set the maximum number of clients at runtime, use the Publisher Status - Client List:

1. Dynamically set the maximum number of clients at runtime using the Publisher Status - Client List panel. To do this, complete the following steps:
   a. Open the Publisher Status - Client List panel. See “IMS Connect Extensions publisher API client list” on page 170.
   b. Enter a number in the Maximum clients field.

   Notes:
   • If you change the maximum number of clients to a value less than the number of currently connected clients, 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 (0), the status of the publisher API changes to NOLOGON.
   • Any value you set in the Maximum clients field will be reset when IMS Connect restarts to the value stored in the IMS Connect system definition.

2. Set a maximum value that persists across IMS Connect system restarts in the IMS Connect system definition. To do this, complete the following tasks:
   a. From the primary option menu, select option 1.1 System Definitions.
   b. Select the IMS Connect system you wish to modify using line action S.
   c. On the System Definition panel, enter a value in the Maximum clients field. The maximum number of clients is set to this value every time you restart IMS Connect. See “Defining IMS Connect systems” on page 320.

Stopping a publisher API client

You can stop a publisher API client when you want it to disconnect.

About this task

For more information on how clients disconnect from IMS Connect Extensions, see “Termination procedure” on page 465.

Procedure

To stop a client:

1. Open the Publisher Status - Client List panel. See “IMS Connect Extensions publisher API client list” on page 170.
2. To stop a client, enter line action P next to one or more client names. If the client is not stopped, use the force stop command:

<table>
<thead>
<tr>
<th>Command ===</th>
<th>Publisher Status - Client List</th>
<th>Row 1 of 7</th>
<th>Scroll ===</th>
<th>CSR</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMS Connect system . : HWS104</td>
<td>Maximum clients . : 1 (0-99)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Publisher status . : LOGON</td>
<td>Connected clients . : 1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

/ Thread  Client  AuthID  Collection Mode
---  -------  --------  -------------------
  P  1  EG AGENT1  USER1  SYNC(512)

Figure 89. Stopping a client on the IMS Connect Extensions Publisher Status - Client List panel

3. To force stop a client, enter line action F next to the client name

**Note:** A force-stop clears the client's entry from IMS Connect Extensions. Force-stopping a client that has not disconnected can cause the client to 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.
Chapter 9. IMS Connect Extensions Operations Console for z/OS Explorer

IBM IMS Connect Extensions for z/OS Operations Console is a plug-in for z/OS Explorer that provides you with additional capabilities beyond the IBM IMS Connect Extensions for z/OS ISPF-based interface. Use the Operations Console to view real-time IMS Connect information, stop and start IMS data stores, suspend and resume routing, and issue IMS Connect WTOR command or IMS Connect z/OS command and IMS type-1 commands.

Use the Operations Console to perform the following tasks:
- Conduct status monitoring of IMS Connect systems and IMS data stores.
- Identify all IMS data stores across all your IMS Connect systems that relate to a specific IMS system.
- Suspend and resume routing to multiple IMS data stores at once to “drain” them of in-progress transactions, typically before IMS shut down.
- Update capacity weight ratings (CWR) across multiple IMS data stores for multiple IMS Connect systems.
- Stop and start IMS data stores, ports, and exits.
- Monitor and view details of active sessions (connections) between IMS Connect and its clients, and then sessions that may be taking longer than expected to complete.
- Review event statistics for a selected IMS Connect system.
- Activate and set session message limits.

Getting started with Operations Console

This topic explains how to get started with the IMS Connect Extensions Operations Console.
Before you begin

If you already use Operations Console and 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

1. Double-click the z/OS Explorer shortcut. If you are prompted to select a workspace folder, select a location anywhere on a local drive. The workspace folder is used to save details of your customizations to the workbench layout and working sets.

   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. You can run old and new versions of the Operations Console plug-in in parallel by installing the plug-in to separate instances of z/OS Explorer, each in a different location.

2. Select Window > Open Perspective > Other (or Window > Perspective > Open Perspective > Other, depending on your version of z/OS Explorer).

3. Select IMS Connect Extensions, and then click OK.

   Figure 91. The IMS Connect Extensions perspective in IMS Connect Extensions Operations Console for z/OS Explorer

4. If this is a new workspace, follow the prompts to complete the User ID and Password fields using the TSO credentials you normally use when using the IMS Connect Extensions ISPF dialog.

   Note: To add additional connection profiles, click the View menu ( ) in the Navigation toolbar and then select Preferences > Connection Profiles. For more information on connection profiles, see “Managing connection profiles” on page 180.
5. If this is the first time you are using the Operations Console, you will need to add your IMS Connect systems. You can do this in the following ways:

- By importing a configuration file. See “Importing systems from a configuration file.”
- By adding systems manually. See “Adding systems manually.”

### Importing systems from a configuration file

You can quickly populate the Operations Console with IMS Connect systems using a configuration file created by IMS Connect Extensions.

#### Before you begin

Configuration files can be used to distribute a standard configuration to many Operations Console users at once. You can obtain a configuration file in the following ways:

- By generating one from definitions stored in your IMS Connect Extensions repository using the definition extract utility. For more information, see Example JCL: create a configuration file for import in the Operations Console.
- By exporting a definitions file from an existing installation of the Operations Console. To do this, click in the Navigation toolbar, and then click Export IMS Connect Extensions Configuration.

#### Procedure

1. From the Navigation view toolbar, click , and then click Import IMS Connect Extensions Configuration.

![Image of Import IMS Connect Extensions Configuration

Figure 92. Importing an IMS Connect Extensions configuration into the IMS Connect Extensions Operations Console for z/OS Explorer

2. In the Import dialog, navigate to the configuration file that you want to import and then click Open. The systems and groups in this configuration file are added to your workspace.

### Adding systems manually

You can add IMS Connect systems manually to the IMS Connect Extensions Operations Console using the Navigation view.

#### Procedure

1. Click in the Navigation view, and then click IMS Connect. The Enter System Details dialog is displayed.
2. Enter the following system information:
System Name
The name of the IMS Connect system.

System Description
A description for this IMS Connect system.

Console Host
The host name or IP address of the mainframe in which the IMS Connect system (with IMS Connect Extensions) resides.

Console Port
The port number in IMS Connect that is reserved for the IMS Connect Extensions Operations Console. You can find this port number in the following locations:
- The system definition in the IMS Connect Extensions ISPF dialog.
- The IMS Connect Extensions log
- The CEXPRINT DD for the IMS Connect job

Connection Profile
Select Default Connection Profile to use the default user ID and password to connect to this system. If other connection profiles have been added, you can select one of them to use for this IMS Connect system.

Note: Passwords must conform to the Activate Mixed Case Passwords setting in the IMS Connect Extensions system definition:
- If Activate Mixed Case Passwords is enabled, you must enter each letter in the password in the correct case.
- If Activate Mixed Case Passwords is not enabled, the password is converted to all uppercase so it does not matter how it is entered.
3. Click **Test Connection** to try to connect to the system using the selected connection profile. If the connection is unsuccessful, check for additional messages in the **Common Services Library Console** view.

4. When you are finished making changes, click **Save**.

**What to do next**

You can also organize your IMS Connect systems into groups. See "**Grouping systems**".

**Grouping systems**

The IMS Connect Extensions Operations Console allows you to arrange your IMS Connect systems into logical groups. Once organized into groups, you can perform group operations on multiple IMS Connect systems at once.

**About this task**

A group can represent any arrangement, such as 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. For example, you can:

- View the systems in the group.
- View the components of all the systems in the group and their activity with the Status Monitor. Use this option to manage exits, super members, and IMS data stores for an entire group.
- View and perform operations on active sessions within the group.

**Procedure**

To create a group:

1. Click the **New** button ( ![New Button](image)) in the **Navigation** view, and then select **Group**.
2. Enter a name for the group and then click **OK**. The new group as added to the **Navigation** view.

To assign IMS Connect systems to the new group:

3. Right-click on the group in the **Navigation** view, and then select **Edit**.
4. Select the IMS Connect systems you wish to add to the group.
5. When you are finished making changes, click **OK**.

6. To access the systems in the group, click the right-arrow icon next to the group name (Warehouse).

**Note:** To refresh the contents of the **Navigation** view, click 🔄.

**Related concepts:**

- "Status Monitor" on page 187
  The Status Monitor in the IMS Connect Extensions Operations Console allows you to monitor and control IMS Connect systems and their components.

- "Active sessions" on page 209
  Active sessions in the IMS Connect Extensions Operations Console allows you to monitor 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.

**Managing connection profiles**

The IMS Connect Extensions Operations Console uses connection profiles to log in to your z/OS systems.

**About this task**

If you only maintain a single set of credentials to log into your z/OS systems, use the **Default Connection Profile** in the procedure below.
Procedure

1. Select Preferences from the View menu.

2. Select Connection Profiles.

3. Select the Default Connection Profile, or click New to create a new profile. If you selected New, enter a name for the new profile and click OK. This can be useful, for example, if you have more than one set of credentials you need to use.

4. Enter the following information next to the profile name:

   **User ID**
   Enter the user ID that you would normally use to connect to z/OS systems.

   **Password**
   Enter the password.

   **Note:** Passwords must conform to the Activate Mixed Case Passwords setting in the system definition and the PWCase parameters on the SECURITY control input data set option.
   - If Activate Mixed Case Passwords is enabled, you must enter each letter in the password in the correct case.
   - If Activate Mixed Case Passwords is not enabled, the password is converted to all uppercase so it does not matter how it is entered.

   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. See “Defining IMS Connect systems” on page 320 and “SECURITY option” on page 568.

5. Click OK to save your changes.

6. Return to the View menu and select Refresh Connection Statuses.

**Troubleshooting connection failures**

If you are having trouble connecting to z/OS or your IMS Connect systems, check for messages in the Common Services Library console view.

---

*Figure 95. Console messages in IMS Connect Extensions Operations Console for z/OS Explorer*
The most likely reasons for a failed connection are described below:

- 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 IMS Connect Extensions system definition:

- If **Activate Mixed Case Passwords** is enabled, you must enter each letter in the password in the correct case.
- If **Activate Mixed Case Passwords** is not enabled, the password is converted to all uppercase so it does not matter how it is entered.

- The specified system name or IMS Connect Extensions console host name is incorrect.
- The console port number is incorrect. Make sure you are using the port defined in IMS Connect Extensions, not one of the IMS Connect ports defined in the IMS Connect configuration member.
- The specified system is not active.
- The number of sockets available in IMS Connect has been exhausted. See “IMS Connect reaches MAXSOC” on page 484.

After you have corrected the system or connection details, select **Refresh Connection Statuses** from the **View** menu in the Navigation view to attempt to reconnect using the new settings.

**Related concepts:**

“Connection failure between a client and an IMS Connect Extensions console listener” on page 483

If an attempt by a client to connect to the IMS Connect Extensions console listener on an IMS Connect system was unsuccessful, use the following information to help resolve the issue.

**Exploring the Operations Console**

IMS Connect Extensions Operations Console is implemented as an Eclipse plug-in for use in IBM Explorer for z/OS (z/OS Explorer).
By default the Operations Console has the following 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** that open when an IMS Connect system is selected from the **Navigation** view.
- At the bottom: a **Console** view for displaying Common Services Library and z/OS Explorer messages.

The layout is highly configurable. For example, you can open the **Properties** view to see a detailed view of the attributes of a selected item. Right-click a listed item in a table view and select **Properties** to display the Properties view. The Manage List Layout window allows a particular arrangement of tabular data in each view to be saved and reused.

Configuration settings that are changed during a session are saved to the user’s specified workspace folder.

**Navigation view**

The **Navigation** view is a container for data sources, including all your IMS Connect systems and groups. Any systems that you have defined are displayed in the **Navigation** view. You can perform the following functions from the **Navigation** view:

- Import or export systems from an IMS Connect Extensions configuration file. See “Importing systems from a configuration file” on page 177.
- Manually define IMS Connect systems and groups and assign systems to one or more groups. See “Adding systems manually” on page 177.
- View details of an IMS Connect system. This opens a new editor in which you can monitor and interact with IMS Connect systems, components, and resources. See “Browsing systems” on page 184.
- Define connection profiles and set startup options and other preferences for Common Services Library features. See “Managing connection profiles” on page 180.

**Context menus**

Right-clicking on most list items and data rows in the Operations Console displays a context menu showing the available options for the current panel or selected item.

**Panel size and position**

Panels within the Operations Console 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. You can maximize a panel by double-clicking its title. To restore it to its previous position, double-click the title again.

**Displaying messages**

IMS Connect Extensions messages display in the Common Services Library Console view at the bottom, below the editor panel. The context menu includes options to:

- Search for a text string
- Clear all messages
- Lock the display so that the current messages remain visible as new messages are appended

**Browsing systems**

The IMS Connect Extensions Operations Console allows you to browse systems and view connection status using the **Navigation** view.

**Procedure**

Find the system you are interested in by expanding the items in the **Navigation** view.

1. Click the icon next to the **All Sources** entry to see its contents. The **IMS Connect** category is displayed.

   **Note:** If you have the IBM IMS Configuration Manager plug-in for z/OS Explorer installed, **IMS Connect** is displayed between the IMS and IMSplex categories.

2. Continue to expand the entries in the **Navigation** view until you find the system you are interested in.
3. Double-click an entry to view additional details.

Results

Status icons may appear next to IMS Connect entries displayed in the Navigation view.

Green circle (●)
A connection with the IMS Connect system has been established.

Red square (◆)
IMS Connect Extensions could not connect to the IMS Connect system.

What to do next

Search for a particular IMS Connect system by typing search terms into the search list. A history of search terms can be accessed by clicking the arrow.

Exporting data to a CSV file

Use the Export option to save the data that is currently displayed in the IMS Connect Extensions Operations Console to a file in CSV (comma-separated values) format.
Before you begin

Adjust the display so that the data you desire is shown. You can export data from the Status Monitor, Sessions (active sessions), and Message Log tabs.

Procedure

1. In the Operations Console toolbar, click the Export button ( ). The Export dialog is displayed.
2. Select a location and filename for the exported data.

   Tip: To associate the file with your spreadsheet software, use the “.csv” file extension.
3. Click Save.
4. Navigate to the saved location and double click on the file to open it.

Related concepts:

“Message Log” on page 217

The Message Log tab in the IMS Connect Extensions Operations Console allows you to view information about IMS Connect and IMS Connect Extensions activity.

Related tasks:

“Displaying statistics on IMS Connect activity” on page 188

Events statistics displayed in the Status Monitor can be accessed by first selecting an IMS Connect system or group followed by selecting the Status Monitor tab in the IMS Connect Extensions Operations Console.

“Displaying active sessions” on page 210

Active sessions can be viewed by first selecting an IMS Connect system or group followed by selecting the Sessions tab.

Related reference:

“QUERY commands” on page 436

The QUERY host commands for REXX reports on statistics and settings for IMS Connect Extensions resources.
Status Monitor

The Status Monitor in the IMS Connect Extensions Operations Console allows you to monitor and control IMS Connect systems and their components.

Use the Status Monitor to perform the following tasks:

- View and export event statistics for IMS Connect systems and groups of systems, TCP/IP ports, user exits, and IMS data stores
- Manage IMS Connect systems:
  - View and clear ACEE cache statistics
  - Switch from the active journal to the next
  - Update event collection levels
  - Update session message limits
  - Update ODBM and OTMA routing plans
  - Activate tracing based on filtering criteria (conditional tracing)
  - Diagnose problems with IMS Connect user exits via recorder tracing
- Start and stop IMS Connect TCP/IP ports
- Start, stop, and reload user exits
- View the status of IMS data stores as candidates for OTMA routing
- Manage IMS data stores:
  - Start and stop IMS data stores
  - Suspend and resume routing of messages to an IMS data store (drain)
  - Make adjustments to workload balancing by changing capacity weights for IMS data stores

Related concepts:
Use the Status Monitor in the IMS Connect Extensions ISPF dialog to view activity on one or more IMS Connect systems, and to perform actions such as starting and stopping IMS data stores, 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.

**Displaying statistics on IMS Connect activity**

Events statistics displayed in the Status Monitor can be accessed by first selecting an IMS Connect system or group followed by selecting the **Status Monitor** tab in the IMS Connect Extensions Operations Console.

**Before you begin**

To display activity for a system, statistics collection must be active. For more information, see “Defining IMS Connect systems” on page 320.

**Procedure**

1. Open an IMS Connect system or group using the **Navigation** view. See “Browsing systems” on page 184.

2. Select the **Status Monitor** tab. The **Status Monitor** tab is located at the bottom of the editor. If you enter the status monitor from a group, all components within that group are displayed.

3. Select a tab at the top of the display to view event statistics. The following views are available:

   - **IMS Connects**
     - IMS Connect (HWS) systems.
   - **Ports**
     - TCP/IP ports (and local adaptor).
   - **Exits**
     - User exits.
   - **Datastores**
     - IMS data stores.
   - **Datastore Groups**
     - IMS data store groups.
   - **ODBM**
     - Open Database Manager (ODBM) targets.
   - **Aliases**
     - IMS aliases.
   - **MSCs**
     - IMS Multiple Systems Coupling (MSC) links.
   - **Remote Connects**
     - Remote IMS Connect systems.
4. Select a statistics reporting window and refresh the display.
   • To do this in the Ports tab, select a value from the Ports interval dropdown.
   • For all other tabs, select a value from the Other Interval dropdown.

After you have selected a value, force a refresh by clicking the Refresh button.

Event statistics in IMS Connect Extensions are collected every 20 seconds. The value you select in the dropdown determines the reporting window that the summary statistics represent.

The following values are available in the Other Interval dropdown.

20 seconds
   The last (most recent) 20 second interval.

1 minute
   The last minute (sum of the last three 20 second intervals).

1-15 minutes
   The last quarter hour (sum of the most recent 15 minutes).

16-30 minutes
   The second-last quarter hour (sum of 16-30 minutes ago).

31-45 minutes
   The third-last quarter hour (sum of 31-45 minutes ago).

46-60 minutes
   The fourth-last quarter hour (sum of 46-60 minutes ago).

1 hour
   Sum of activity in the last hour.

For the Ports interval dropdown, you can select from 20 seconds and 1 minute.

5. Optional: To refresh the statistics automatically every 20 seconds, click the Auto button so that it displays Auto Off. To stop the display being automatically refreshed, click the Auto button again.

6. Use the following additional features to aid your analysis:

   Adjust table columns
   To sort a column by value, click the column heading. To change the column ordering, drag it to its new position. You can adjust the width of a column by dragging it to the desired width.

   Right click on the table for the following additional options:
• To restore the display, right-click anywhere in the table and select **Reset List to Default Layout**.

• To hide columns, select **Hide Blank Columns** or **Hide all-Zero Value Columns** (columns for which the value is zero for all displayed IMS Connects). To reset the display, select **Show all Columns**.

• To save a customized layout, select **Manage list layout**.

**Find values**

Use the **Find Next** ( ) and **Find Previous** ( ) buttons in the toolbar to search for a text string in the displayed list.

**Filter values**

Click the **Filter** button ( ) and then click **Manage filters** to define selection criteria so that only matching systems and components are displayed or highlighted. To toggle a filter on or off:

a. Click the **Filter** button ( ).
b. Click the name of the saved filter.
c. Select **Activate** or **Deactivate**.

**Summarize and group data**

Use the **Summarize/Group** button ( ) to generate a summary table according to fields you select, and then display records matching the summary filter.

**Export data**

Use the **Export** button ( ) to export all of the displayed data to a comma separated values (CSV) file.

**Save the display**

Click the **Save** ( ) button in the toolbar to save the current editor layout. You can redisplay this layout later by selecting it from the **Saved Displays** -- dropdown.

**Information reported in the Status Monitor of the IMS Connect Extensions Operations Console**

The following information is available in the Status Monitor.

**Information common to all tabs**

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The name of the type of object being displayed in the tab. For example, if you have selected the <strong>Ports</strong> tab, this column shows the name of the port. Similarly, if you have selected the <strong>Datastores</strong> tab, this column shows you the name of the IMS data store.</td>
</tr>
<tr>
<td>Status</td>
<td>The status of the object listed in the <strong>Name</strong> column.</td>
</tr>
<tr>
<td>System</td>
<td>The name of the associated IMS Connect system</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start Time</td>
<td>HWS start time.</td>
</tr>
</tbody>
</table>

---

190 IBM IMS Connect Extensions for z/OS: User’s Guide
<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Super Member</td>
<td>The OTMA super member name defined on the SMEMBER parameter defined in your system configuration.</td>
</tr>
<tr>
<td>OTMA Routing Plan</td>
<td>OTMA routing plan name. See “Activating and deactivating OTMA routing plans” on page 198.</td>
</tr>
<tr>
<td>ODBM Routing Plan</td>
<td>ODBM routing plan name. See “Activating and deactivating ODBM routing plans” on page 197.</td>
</tr>
<tr>
<td>Event Coll. Level</td>
<td>Event collection level. See “Updating the event collection level” on page 195.</td>
</tr>
<tr>
<td>Msg. Limit</td>
<td>Session message limit status (active or inactive).</td>
</tr>
<tr>
<td>Limit Threshold</td>
<td>Session message limit threshold. See “Updating the session message limit” on page 196.</td>
</tr>
<tr>
<td>Journal % Used</td>
<td>For IMS Connect Extensions V3.1 systems and higher, the percentage of the active IMS Connect Extensions journal that is used.</td>
</tr>
<tr>
<td>Sockets % Used</td>
<td>For IMS Connect Extensions V3.1 systems and higher, the current number of sockets used as a percentage of MAXSOC.</td>
</tr>
<tr>
<td>Pre-Exit</td>
<td>Pre-exit status (active or inactive).</td>
</tr>
</tbody>
</table>

**Information specific to the Ports tab**

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keep Alive</td>
<td>KeepAlive value.</td>
</tr>
</tbody>
</table>

**Information specific to the Datastores and Datastore Groups tabs**

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connect Status</td>
<td>Datastore IMS Connect status. This field shows if the Cross-System Coupling Facility (XCF) connection is Active, Inactive, or Discon (disconnected).</td>
</tr>
<tr>
<td>IMS Status</td>
<td>Datastore IMS (OTMA) status. This field shows the availability of an active IMS system for OTMA transactions.</td>
</tr>
<tr>
<td></td>
<td><strong>Normal</strong>&lt;br&gt;The IMS system is available for work.</td>
</tr>
<tr>
<td></td>
<td><strong>MemberFloodWarn</strong>&lt;br&gt;The datastore is in a Member Flood Warning state.</td>
</tr>
<tr>
<td></td>
<td><strong>MemberFloodSevr</strong>&lt;br&gt;The datastore is in a Member Flood Severe state.</td>
</tr>
<tr>
<td></td>
<td><strong>GlobalFloodWarn</strong>&lt;br&gt;The datastore is in a Global Flood Warning state.</td>
</tr>
<tr>
<td></td>
<td><strong>GlobalFloodSevr</strong>&lt;br&gt;The datastore is in a Global Flood Severe state.</td>
</tr>
<tr>
<td></td>
<td><strong>Unknown</strong>&lt;br&gt;The target IMS system is V9 or earlier.</td>
</tr>
</tbody>
</table>
### Field name | Description
--- | ---
Routing Status | Datastore routing status. This field shows the status of the datastore as a candidate for OTMA routing.
Normal | The data store is potentially a candidate for routing.
NormalWLB | The data store is potentially a candidate for routing, and workload balancing is active.
Degraded | The data store 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 data store is not a candidate for routing. This could be due to a severe-level IMS status.
SusCWRZero | The data store has been suspended as a candidate for routing. Workload balancing is active and the data store has a capacity weight ratio of zero.
Suspended | The data store has been suspended as a candidate for routing. The data store has been drained without the autoresume option. The data store might be stopped (see ICON Status).
SusAutoRes | The data store has been suspended as a candidate for routing. It will automatically resume routing when the data store has been stopped and then becomes active again.
 AutoRes or AutoResume | The data store is stopped but if it is started it will automatically become a candidate for routing.
Waiting Reply | Datastore pending response count
CWR | Datastore capacity weight rating. See “Defining IMS data stores” on page 339.
Member | Datastore XCF member
TMember | Datastore XCF Tmember
XCF Group | Datastore XCF group
Super Member | The OTMA super member name defined on the SMEMBER parameter defined in your system configuration

### Information specific to the Alias tab

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alias</td>
<td>IMSAlias name</td>
</tr>
</tbody>
</table>
### Information common to the IMS Connects, Ports, Exits, Datastores, and Datastore Groups tabs

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accepted Count</td>
<td>Accepted requests count</td>
</tr>
<tr>
<td>ACK Count</td>
<td>ACK count</td>
</tr>
<tr>
<td>DEALLOC count</td>
<td>DEALLOC count</td>
</tr>
<tr>
<td>Ignore Count</td>
<td>Ignored count</td>
</tr>
<tr>
<td>Input Count</td>
<td>Input count</td>
</tr>
<tr>
<td>Input Bytes</td>
<td>Input cumulative bytes</td>
</tr>
<tr>
<td>Input Max.</td>
<td>Input maximum length</td>
</tr>
<tr>
<td>Input Min.</td>
<td>Input minimum length</td>
</tr>
<tr>
<td>NAK Count</td>
<td>NAK count</td>
</tr>
<tr>
<td>Rejected Count</td>
<td>Rejected count</td>
</tr>
<tr>
<td>Rej. EXER Count</td>
<td>Rejected by EXER exit count</td>
</tr>
<tr>
<td>Rej. Other Count</td>
<td>Rejected other count</td>
</tr>
<tr>
<td>Rej. Pace Count</td>
<td>Rejected by pacing count</td>
</tr>
<tr>
<td>Rej. Route Count</td>
<td>Rejected by routing count</td>
</tr>
<tr>
<td>Rej. Sec. Count</td>
<td>Rejected by security count</td>
</tr>
<tr>
<td>Request Count</td>
<td>Request count</td>
</tr>
<tr>
<td>Returned Count</td>
<td>Returned count</td>
</tr>
<tr>
<td>Routed Count</td>
<td>Routed count</td>
</tr>
<tr>
<td>Res. TPIPES Count</td>
<td>RTPPIPE count</td>
</tr>
<tr>
<td>Res. TPIPES Bytes</td>
<td>RTPPIPE cumulative bytes</td>
</tr>
<tr>
<td>Res. TPIPES Max</td>
<td>RTPPIPE maximum length</td>
</tr>
<tr>
<td>Res. TPIPES Min</td>
<td>RTPPIPE minimum length</td>
</tr>
<tr>
<td>Send Only Count</td>
<td>Send only count</td>
</tr>
<tr>
<td>Send Only Bytes</td>
<td>Send only cumulative bytes</td>
</tr>
<tr>
<td>Send Only Max.</td>
<td>Send only maximum length</td>
</tr>
<tr>
<td>Send Only Min.</td>
<td>Send only minimum length</td>
</tr>
<tr>
<td>Sent Err. Count</td>
<td>Sent error count</td>
</tr>
<tr>
<td>Sent Err. Bytes</td>
<td>Sent error cumulative bytes</td>
</tr>
<tr>
<td>Sent Err. Max.</td>
<td>Sent error maximum length</td>
</tr>
<tr>
<td>Sent Err. Min.</td>
<td>Sent error minimum length</td>
</tr>
<tr>
<td>Sent OK Count</td>
<td>Send OK count</td>
</tr>
<tr>
<td>Sent OK Bytes</td>
<td>Send OK cumulative bytes</td>
</tr>
<tr>
<td>Sent OK Max.</td>
<td>Send OK maximum length</td>
</tr>
<tr>
<td>Sent OK Min.</td>
<td>Send OK minimum length</td>
</tr>
</tbody>
</table>

**Related tasks:**

["Exporting data to a CSV file" on page 185](#)

Use the **Export** option to save the data that is currently displayed in the IMS Connect Extensions Operations Console to a file in CSV (comma-separated values) format.
Managing IMS Connect systems

Use the IMS Connects tab in Status Monitor to perform management tasks on IMS Connect systems from the IMS Connect Extensions Operations Console.

Procedure

1. Open the IMS Connects tab. See “Displaying statistics on IMS Connect activity” on page 188. A list of IMS Connect systems is displayed.

2. Right-click on an IMS Connect to see a list of available actions:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Switch Journal</td>
<td>Switch from the current active IMS Connect Extensions journal to the next journal.</td>
</tr>
<tr>
<td>Update &gt; Event Collection Level</td>
<td>Change the IMS Connect Extensions event collection level. The event collection level determines which events are actually recorded in the journal.</td>
</tr>
<tr>
<td>Update &gt; Session Message Limit</td>
<td>Update the session message limit.</td>
</tr>
<tr>
<td>Update &gt; ODBM routing plan</td>
<td>Activate or deactivate an ODBM routing plan.</td>
</tr>
<tr>
<td>Update &gt; OTMA routing plan</td>
<td>Activate or deactivate an OTMA routing plan.</td>
</tr>
<tr>
<td>ACEE Cache Statistics</td>
<td>View accessor environment element (ACEE) cache statistics.</td>
</tr>
<tr>
<td>Clear ACEE Cache</td>
<td>Remove selected security credentials from the ACEE cache.</td>
</tr>
<tr>
<td>Start Conditional Trace</td>
<td>Activate tracing on an IMS Connect system based on filtering criteria.</td>
</tr>
<tr>
<td>Stop Conditional Trace</td>
<td>Stop tracing on an IMS Connect system.</td>
</tr>
<tr>
<td>Start Recorder Trace</td>
<td>Start the IMS Connect Recorder Trace facility. The Recorder Trace is used primarily for diagnosing problems with the IMS Connect user exit.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Stop Recorder Trace</td>
<td>If the IMS Connect Recorder Trace facility is active, stop the Recorder Trace.</td>
</tr>
</tbody>
</table>

**Updating the event collection level**
Use the Update > Event Collection Level context menu option in the Status Monitor of the Operations Console to change the IMS Connect event collection level on selected IMS Connect systems. The setting is changed both in-memory and in the repository definition.

**About this task**
The event collection level determines which events are actually recorded in the journal.

**Procedure**
1. Right click on an IMS Connect system and select Update > Event Collection Level from the context menu. The Collection Level dialog is displayed.

![Collection Level](image)

**Figure 101. Updating the IMS Connect event collection level in IMS Connect Extensions Operations Console for z/OS Explorer**

2. Select a new event collection level. The following levels are available:
   - **0 - Minimum Level**
     Collects startup and shutdown events along with some infrequent error events.
   - **1 - Accounting Level**
     Collects Return from Exit events, Open Transaction Manager Access 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 Connect connectivity.
   - **4 - Maximum Level**
     Collects all event records.
(OTMA) timeout and session error events, and Open Database Manager (ODBM) registration and routing events. This level provides accounting information in terms of the number of messages by Transaction, User Exit, and more.

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 Connect connectivity.

4 - Maximum Level
Collects all event records.

Tip: For more information on how you can use the collected event data to report on IMS Connect performance, resource usage, and transit event tracing, see “IMS Connect Extensions event collection” in the IBM IMS Performance Analyzer for z/OS Report Reference.

3. Click Next. The old and new values are displayed for each selected system.
4. To confirm the change in collection level, click Finish.

Note: To retain the old settings, click Back, select No Change, and then click Cancel.

Related concepts:
[Chapter 24, “Events recorded by IMS Connect Extensions,” on page 551]

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.

**Updating the session message limit**
Use the Update > Session Message Limit context option in the Status Monitor of the Operations Console to activate or deactivate session message limits on selected systems or to change the threshold. The setting is changed both in-memory and in the repository definition.

**About this task**

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. The Limit threshold specifies the maximum number of input messages for a persistent session. When the 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 Read Exit Msg Cnt field on the Sessions tab displays the number of input messages received so far by the session.

**Procedure**

To implement automatic session re-balancing, perform the following steps:
1. Right-click on an IMS Connect system and select Update > Session Message Limit from the context menu.
2. Under Session Message Limit:
   - To activate session message limiting, select Active. In the Limit threshold field, specify the number of input messages a persistent session can receive before it will be closed. This value must be between 1 and 999999.

![Session Message Limit](image)

Figure 102. Updating the session message limit in IMS Connect Extensions Operations Console for z/OS Explorer

   - To deactivate session message limiting, select Inactive.
3. Click Next. The old and new values are displayed for each selected system.
   - To confirm the changes, click Finish.
   - To retain the old settings, click Back, select No Change, and then click Cancel.

Activating and deactivating ODBM routing plans

Use the Update > ODBM Routing Plan context option in the Status Monitor of the Operations Console to activate or deactivate ODBM routing plan on several IMS Connect systems at once.

About this task

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 ODBM routing rule definition.

When no routing plan is activated, only unconditional rules are active. Unconditional rules are rules that are not assigned to any routing plan.

Procedure

1. Right-click on an IMS Connect system and select Update > ODBM Routing Plan from the context menu.
2. Under ODBM Routing Plan:
   - To activate a routing plan, select Active and then choose the routing plan name.
• To have no routing plan active on the selected systems, select Inactive.

3. Click Next. The old and new values are displayed for each selected system.
• To confirm the changes, click Finish.
  The corresponding ODBM routing rules are automatically rebuilt.
• To retain the old settings, click Back, select No Change, and then click Cancel.

Related tasks:
“Activating ODBM routing rules in IMS Connect” on page 294
ODBM routing rules defined in the IMS Connect Extensions repository only take effect in IMS Connect once you have activated ODBM rules-based routing.

Related reference:
“SET command” on page 453
The SET host command for REXX activates or deactivates an OTMA or ODBM routing plan.

Activating and deactivating OTMA routing plans
Use the Update > OTMA Routing Plan context option in the Status Monitor of the Operations Console to activate or deactivate OTMA routing plan on several IMS Connect systems at once.

About this task
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 routing rule definition.

When no routing plan is activated, only unconditional rules are active. Unconditional rules are rules that are not assigned to any routing plan.

Procedure
1. Right-click on one or more IMS Connect systems and select Update > OTMA Routing Plan from the context menu.
2. Under OTMA Routing Plan:
   • To activate a routing plan, select Active and then choose the routing plan name.
   • To have no routing plan active on the selected systems, select Inactive.
3. Click Next. The old and new values are displayed for each selected system.
   • To confirm the changes, click Finish.
     The corresponding OTMA routing rules are automatically rebuilt.
   • To retain the old settings, click Back, select No Change, and then click Cancel.

Related tasks:
“Activating OTMA routing rules in IMS Connect” on page 259
OTMA routing rules defined in the IMS Connect Extensions repository only take effect in IMS Connect once you have activated OTMA rules-based routing.

Related reference:
“SET command” on page 453
The SET host command for REXX activates or deactivates an OTMA or ODBM routing plan.
Viewing ACEE cache statistics

Use the ACEE Cache Statistics context menu option in the Status Monitor of the Operations Console to display information about the caching of security credentials in IMS Connect.

![ACEE Cache Statistics in IMS Connect Extensions Operations Console for z/OS Explorer](image)

**Figure 103. ACEE Cache Statistics in IMS Connect Extensions Operations Console for z/OS Explorer**

**Cache Space**

**Element length**

The number of bytes each cached credential requires.

**Initial size**

The initial cache size in bytes.

**Expansion size**

The size in bytes of each expansion of the cache.

**Maximum expansions**

The maximum number of times the cache can be expanded.

**Number of expansions**

The number of times the cache has been expanded.

**Percentage used**

The percentage of the allocated space which is currently used.

**Accounting stats**

**Insert requests**

Total number of elements inserted into the cache.

**Get requests**

Number of get requests.

**Get next requests**

Number of get next requests.

**Delete requests**

Number of elements deleted from the cache.
Stat requests
  Number of stat requests.

Elements in cache
  Number of currently cached credentials.

Clearing the ACEE cache
Use the Clear ACEE Cache context option in the Status Monitor of the Operations Console to clear the security credentials of specified user IDs from the ACEE cache.

Procedure
1. Right-click on an IMS Connect system and select Clear ACEE Cache from the context menu.
2. To clear the credentials of a single user from the cache for the selected system, simply enter the user ID. You can also delete multiple users by specifying a prefix followed by a trailing asterisk (*). Examples:
   - To delete all user IDs starting with “AB”, enter AB*.
   - To delete all ACEE structures, enter *.

![Clear ACEE Cache](image)

Figure 104. Clearing the ACEE cache in IMS Connect Extensions Operations Console for z/OS Explorer

Using conditional trace
Use the Start Conditional Trace and Stop Conditional Trace context menu options in the Status Monitor of the Operations Console to start and stop tracing of an IMS Connect system based on filtering criteria.

About this task
Tracing records are written to the Active Journal data set and then, if archiving is enabled, archived to the Archive Journals. For a list of the IMS Connect trace records, see Chapter 24, “Events recorded by IMS Connect Extensions,” on page 551.

Procedure
To specify filtering options and start the trace:
1. Right-click on the IMS Connect system and select Start Conditional Trace from the context menu.
2. Select from the following conditional tracing options:
Reactivate Tracing after System Restart
Select this option to make tracing persist across system restarts.

Note: You can only make a trace persistent at the same time as you activate it.

Level
Set the level of tracing for IMS Connect Extensions:
1 IRM, CSM, RSM, OTMA, XML, and DRDA structures.
2 IRM, CSM, RSM, OTMA, XML, DRDA structures, and client application data.

Port
Apply tracing only to a specific port. Only activities directed at that port are traced. Use an asterisk (*) to trace on all ports, or LOCAL for a local port.

Filter Criterion (for OTMA workloads)
IMS Connect Extensions will only record messages that match the specified tracing filter. Specify an optional filter (one only) using the following fields:

Client name
The name of the client.

Transaction
The transaction.

Message exit
The message exit.

User ID
The user ID.
**LTERM**

The logical terminal.

**IP address**

The host IP address. If you enter a DNS name it will be sent to the server to be resolved to the IP address.

3. Click **Start Trace**.

To stop conditional tracing:

4. Right-click on the IMS Connect system and select **Stop Conditional Trace** from the context menu.

**Related concepts:**

Chapter 24, “Events recorded by IMS Connect Extensions,” on page 551

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.

**Related tasks:**

“Using the IMS Connect Extensions trace” on page 93

Tracing events are additional event records produced by IMS Connect Extensions when tracing is activated. Use the IMS Connect Extensions ISPF dialog to start and stop tracing for an active IMS Connect system.

**Using Recorder Trace**

Use the **Start Recorder Trace** and **Stop Recorder Trace** context menu options in the Status Monitor of the IMS Connect Extensions Operations Console to start and stop the IMS Connect Recorder Trace facility.

**Procedure**

To start the IMS Connect Recorder Trace facility:

1. Right-click on the IMS Connect system and select **Start Recorder Trace** from the context menu.

To stop the IMS Connect Recorder Trace facility:

2. Right-click on the IMS Connect system and select **Stop Recorder Trace** from the context menu.

**What to do next**

Review the Common Services Library **Console** to see the status of the IMS Connect Recorder Trace facility.

**Related tasks:**

“Controlling the IMS Connect Recorder Trace facility” on page 96

Use IMS Connect Extensions ISPF dialog to start and stop the IMS Connect Recorder Trace facility.

“Controlling the IMS Connect Recorder Trace facility” on page 96

Use IMS Connect Extensions ISPF dialog to start and stop the IMS Connect Recorder Trace facility.

**Related reference:**

“Recorder Trace conversion utility (CEXRTCNV)” on page 580

The CEXRTCNV utility converts RECORDER trace data into the format of IMS Connect Extensions event records.
Managing IMS Connect TCP/IP ports

Use the Ports tab in Status Monitor of the Operations Console to start and stop IMS Connect TCP/IP ports from the IMS Connect Extensions Operations Console.

Procedure

1. Open the Ports tab. See “Displaying statistics on IMS Connect activity” on page 188. A list of ports is displayed.

   Tip: A suffix on the port number identifies the port type. The suffix may be one of the following values:

   D   Distributed Relational Data Architecture (DRDA)
   S   Secure Sockets Layer (SSL)

   If the port number has no suffix, the port has been defined on the PORT or PORTID parameter of the TCPIP statement defined in the HWSCFGxx member of the IMS PROCLIB data set.

2. Right-click on a port to see a list of available actions.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start</td>
<td>Start communication on the port</td>
</tr>
<tr>
<td>Stop</td>
<td>Stop communication on the port</td>
</tr>
</tbody>
</table>

Note:

If Show confirmation on Stop/Start is selected, a confirmation dialog is displayed before the port is updated.

Figure 106. Stopping a port in IMS Connect Extensions Operations Console for z/OS Explorer

Related concepts:
“Ports Overview” on page 125

To view the active TCP/IP ports of the IMS Connect system in the IMS Connect Extensions Status Monitor, select the Ports view.

Managing user exits

Use the Exits tab in Status Monitor tab to manage user exits from the IMS Connect Extensions Operations Console.
Procedure

1. Open the Exits tab. See “Displaying statistics on IMS Connect activity” on page 188. A list of exits is displayed.

2. Right-click on an exit to see a list of available actions.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reload</td>
<td>Reload the user exit</td>
</tr>
<tr>
<td>Start</td>
<td>Start the exit</td>
</tr>
<tr>
<td>Stop</td>
<td>Stop the exit</td>
</tr>
</tbody>
</table>

Note:

If Show confirmation on Stop/Start is selected, a confirmation dialog is displayed before the exit is updated.

Managing IMS data stores

Use the Datastores and Datastore Groups tab in Status Monitor to perform management tasks on IMS data stores from the IMS Connect Extensions Operations Console. Use this facility to start and stop IMS data stores, suspend and resume routing to a data store, and to dynamically update a data stores capacity weight rating (CWR) in response to periods of changing demand.

Procedure

1. Open the Datastores or Datastore Groups tab. See “Displaying statistics on IMS Connect activity” on page 188. A list of IMS data stores is displayed. In the following example, a filter has been applied to highlight the routing status of each data store.
Use the following columns to review the status of each IMS data store:

**Status**  The status of the data store. A data store may be in one of the following states:

- **Green circle ( )**  The data store is active.

- **Yellow diamond ( )**  The data store is experiencing a flood condition.

- **Red square ( )**  The data store is inactive.

**IMS Status**  This field shows the availability of an active IMS system for OTMA transactions. If the ICON status value is inactive or disconnected, IMS Status will be blank. If the ICON status value is active, the possible values of IMS Status are:

- **Normal**  The IMS system is available for work.

- **MemberFloodWarn**  The data store is in a Member Flood Warning state.

- **MemberFloodSevr**  The data store is in a Member Flood Severe state.

- **GlobalFloodWarn**  The data store is in a Global Flood Warning state.

- **GlobalFloodSevr**  The data store is in a Global Flood Severe state.

- **Unknown**  The target IMS system is V9 or earlier.

See **IMS Communications and Connections** for more information on the Degraded (Warning) and Unavailable (severe) flood states.

**Routing Status**  The status of the IMS data store as a candidate for OTMA routing in IMS Connect Extensions. A data store may be in one of the following routing states:

- **Normal**  The data store is potentially a candidate for routing.

---

*Figure 108. Viewing the status of IMS data stores in the IMS Connect Extensions Operations Console for z/OS Explorer.*
NormalWLB
The data store is potentially a candidate for routing, and workload balancing is active.

Degraded
The data store 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 data store is not a candidate for routing. This could be due to a severe-level IMS status.

SusCWRZero
The data store has been suspended as a candidate for routing. Workload balancing is active and the data store has a capacity weight ratio of zero.

SusPend
The data store has been suspended as a candidate for routing. The data store has been drained without the autoresume option. The data store might be stopped (see ICON Status).

AutoRes or AutoResume
The data store has been stopped but if it is started it will automatically become a candidate for routing.

Waiting reply
A count of pending responses for each data store.

2. Right-click on a data store to see a list of available actions:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start</td>
<td>Start the data store.</td>
</tr>
<tr>
<td>Stop</td>
<td>Stop the data store.</td>
</tr>
<tr>
<td>Route Drain</td>
<td>Suspend routing to the IMS data store. The data store is withdrawn as a candidate for OTMA routing and is “drained” of in-progress transactions (which are allowed to complete normally). Use the Waiting reply column to monitor the number of pending responses. When the IMS data store has a Waiting Reply count of 0, use the Stop action to stop the data store. Important: The drain facility only works on messages handled by IMS Connect Extensions OTMA rules-based routing. It does not work on messages that circumvent a routing rule or with exits customized with their own routing mechanisms. To ensure that all of your workload is handled by IMS Connect Extensions, see “Rejecting transactions for data stores with no OTMA routing rule” on page 265.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Route Drain with AUTORESUME</td>
<td>Suspend routing to the IMS data store until an “IMS data store available” event record (X’10’) is received at which point routing is automatically resumed.</td>
</tr>
<tr>
<td>Route Resume</td>
<td>Resume routing to the IMS data store (if routing was suspended). The IMS data store is once again a candidate for OTMA routing.</td>
</tr>
<tr>
<td>Update Capacity Weight</td>
<td>Change the processing capacity (capacity weight) for the selected IMS data store. Use this feature to redistribute workloads in response to periods of demand. Enter a new capacity weight value, click Next, and then click Finish. To retain the current value, click Back, select No Change, and then click Cancel.</td>
</tr>
</tbody>
</table>

**Figure 109. Draining multiple IMS data stores at once using the IMS Connect Extensions Operations Console for z/OS Explorer**

**Note:** If Show confirmation on Stop/Start is selected, a confirmation dialog is displayed before the data store is updated.

**Related concepts:**

“Datastore Monitor” on page 141

Use the Datastore Monitor in the IMS Connect Extensions ISPF dialog to manage and monitor your IMS data stores. Use the Datastore Monitor to view detailed information about an IMS data store, to suspend and resume routing of messages to an IMS data store using the drain feature, and to start and stop an IMS data store.

“Suspending OTMA workload routing to IMS data stores using drain” on page 273

The drain facility in IMS Connect Extensions temporarily suspends OTMA rules-based routing of messages to a data store (or a set of data stores), 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.

“OTMA flood control processing in IMS Connect Extensions” on page 267

IMS Connect Extensions OTMA rules-based routing can be used to avoid message flood conditions in IMS.
Related tasks:

- “Dynamically adding IMS data stores as targets for OTMA routing” on page 274

IMS Connect Extensions supports rules-based routing of OTMA messages to dynamically added IMS data stores without restarting IMS Connect.

Managing ODBM systems

Use the ODBMs tab in Status Monitor to perform management tasks on ODBM systems from the IMS Connect Extensions Operations Console.

Procedure

1. Open the ODBMs tab. See “Displaying statistics on IMS Connect activity” on page 188. A list of ODBMs is displayed. Use the following columns to review the status of the connection between IMS Connect and each ODBM:

<table>
<thead>
<tr>
<th>Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Green circle (■)</td>
<td>The connection is active.</td>
</tr>
<tr>
<td>Red square (■)</td>
<td>The connection is inactive.</td>
</tr>
</tbody>
</table>

   | System             | The IMS Connect system.            |
   | Name               | The ODBM name.                     |

2. Right-click on a row to see a list of available actions:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start</td>
<td>Start the ODBM.</td>
</tr>
<tr>
<td>Stop</td>
<td>Stop the ODBM.</td>
</tr>
</tbody>
</table>

Managing ODBM systems by IMS alias

Use the Aliases tab in Status Monitor to perform management tasks on ODBM systems listed by IMS alias from the IMS Connect Extensions Operations Console.

Procedure

1. Open the Aliases tab. See “Displaying statistics on IMS Connect activity” on page 188. A list of IMS aliases is displayed. Use the following columns to review the status of the connection between IMS Connect and each ODBM:

<table>
<thead>
<tr>
<th>Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Green circle (■)</td>
<td>The connection is active.</td>
</tr>
<tr>
<td>Red square (■)</td>
<td>The connection is inactive.</td>
</tr>
</tbody>
</table>

   | System             | The IMS Connect system.            |
   | Name               | The ODBM name.                     |
   | Alias              | The IMS alias name.                |

2. Right-click on a row to see a list of available actions:
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start</td>
<td>Start the ODBM.</td>
</tr>
<tr>
<td>Stop</td>
<td>Stop the ODBM.</td>
</tr>
</tbody>
</table>

### Managing MSC
Use the **MSCs** tab in Status Monitor to view MSC information from the IMS Connect Extensions Operations Console.

**Procedure**
1. Open the **MSCs** tab. See “Displaying statistics on IMS Connect activity” on page 188. A list is displayed.
2. Right-click on a row to see a list of available actions.

### Managing remote IMS Connect systems
Use the **Remote Connects** tab in Status Monitor to view remote IMS Connect systems from the IMS Connect Extensions Operations Console.

**Procedure**
1. Open the **Remote Connects** tab. See “Displaying statistics on IMS Connect activity” on page 188. A list of remote IMS connect systems is displayed.
2. Right-click on a row to see a list of available actions.

### Active sessions
Active sessions in the IMS Connect Extensions Operations Console allows you to monitor 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.

Use Active Sessions to perform the following tasks:
- List sessions waiting for longer than a given time.
- Display the network status for a selected session.
- Stop selected sessions.
- Drain eligible active sessions.

**Related concepts:**
"Active Sessions” on page 130

Use the Active Sessions panel in the IMS Connect Extensions ISPF dialog to manage and monitor active OTMA, ODBM, and MSC sessions in IMS Connect. 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, the Active Sessions panel is of most use when you are aware of a transaction that is taking longer to complete than expected.

**Related reference:**
"Active session utility” on page 571

The active session utility assists in problem determination by analyzing an IMS Connect Extensions journal data set and listing details of active sessions.
Displaying active sessions

Active sessions can be viewed by first selecting an IMS Connect system or group followed by selecting the Sessions tab.

Before you begin

To display activity for a system, event collection must be active. For more information, see “Defining IMS Connect systems” on page 320.

Procedure

1. Open an IMS Connect system or group using the Navigation view. See “Browsing systems” on page 184.
2. Select the Sessions tab. The Sessions tab is located at the bottom of the editor. If you enter active sessions from a group, all components within that group are displayed.
3. Select a tab at the top of the display to view the active sessions. The following views are available:
   - **All Sessions**: All active sessions across OTMA, ODBM and MSC.
   - **OTMA Sessions**: Display only the Open Transaction Manager Access (OTMA) sessions.
   - **ODBM Sessions**: Display only the Open Database Manager (ODBM) sessions.
   - **MSC**: Display only the Multiple Systems Coupling (MSC) sessions.
4. Select from the following additional options:
   - **Session wait time (seconds)**: List only those sessions waiting for longer than the specified time.
   - **Include persistent sockets**: When checked, this option will include all persistent sockets in the resulting list, even if they are in the READ PREPARE (idle) state. Active persistent sockets are always displayed, regardless of setting.
   - **Display Limit (total sessions)**: Limit the number of results returned to the specified value.

After you have selected a value, force a refresh by clicking the Refresh button.

5. Optional: To refresh the list of sessions automatically every 20 seconds, click the Auto button so that it displays Auto Off. To stop the display being automatically refreshed, click the Auto button again.

6. Use the following additional features to aid your analysis:
   - **Adjust table columns**: To sort a column by value, click the column heading. To change the column ordering, drag it to its new position. You can adjust the width of a column by dragging it to the desired width.

   Right click on the table for the following additional options:
   - To restore the display, right-click anywhere in the table and select Reset List to Default Layout.
- To hide columns, select **Hide Blank Columns** or **Hide all-Zero Value Columns** (columns for which the value is zero for all displayed IMS Connects). To reset the display, select **Show all Columns**.

- To save a customized layout, select **Manage list layout**.

**Find values**

Use the **Find Next** ( לגמרי) and **Find Previous** (_above) buttons in the toolbar to search for a text string in the displayed list.

**Filter values**

Click the **Filter** button (.sep) and then click **Manage filters** to define selection criteria so that only matching systems and components are displayed or highlighted. To toggle a filter on or off:

   a. Click the **Filter** button (sep).
   b. Click the name of the saved filter.
   c. Select **Activate** or **Deactivate**.

**Summarize and group data**

Use the **Summarize/Group** button (sep) to generate a summary table according to fields you select, and then display records matching the summary filter.

**Export data**

Use the **Export** button (sep) to export all of the displayed data to a CSV file.

**Save the display**

Click the **Save** (sep) button in the toolbar to save the current editor layout. You can redisplay this layout later by selecting it from the **Saved Displays** dropdown.

**Related tasks:**

[“Exporting data to a CSV file” on page 185](#)

Use the **Export** option to save the data that is currently displayed in the IMS Connect Extensions Operations Console to a file in CSV (comma-separated values) format.

**Managing active sessions**

Use the **Sessions** tab to manage your active sessions from the IMS Connect Extensions Operations Console.

**Procedure**

1. Display a list of active sessions. See [“Displaying active sessions” on page 210](#).
2. Right-click on a session to see a list of available actions.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Status</td>
<td>Provides network status information. To refresh the display, click <strong>Refresh</strong>.</td>
</tr>
</tbody>
</table>
Option | Description
--- | ---
Drain Eligible Sessions | Mark a session for closure at the next inbound message. This option allows safe closure of the session without interrupting in-flight messages. To drain a session, the session must meet the following criteria:
- Be persistent
- Cannot have an active IMS conversation
- Cannot have active Resume TPIPE requests
- Cannot be an IMS-to-IMS or MSC sessions

Stop Selected Sessions | Stop the session.

Properties | View detailed information about a session. The state of the session dictates the level of detail displayed in the Properties dialog. For example, a session in the READ socket event state contains only basic information, but when it returns from the exit the session contains more information.

**Note:** You can also access session properties by selecting Window > Show View > Other > General > Properties.

---

**Issuing IMS Connect WTOR commands and IMS Connect z/OS commands**

Use the Connect Commands tab in the IMS Connect Extensions Operations Console to issue IMS Connect WTOR commands and IMS Connect z/OS command.

**Procedure**

Note: You must select an IMS Connect system, not a group.

2. Select the **Connect Commands** tab. The **Connect Commands** tab is located at the bottom of the editor.

3. Select a command (or type your own) from the **Command** dropdown.

4. Replace any parameters in the command with your desired values as required. Parameters are shown as lowercase text enclosed in angle brackets ( `<` and `>`). For example, in the command `VIEWPORT <port_id>`, replace `<port_id>` with the desired port.

5. To issue the command, click the **Run** button ( ⏯️ ).
What to do next

Use the following features to aid your analysis:

- Use the Find Next (🔍) and Find Previous (🔍) buttons in the toolbar to search for a text string in the output.
- Use the Export button (��) to export the displayed data to a CSV file.

Note: You can define granular access control to IMS Connect commands in RACF so that users’ access to some commands from the shell is restricted.

Related tasks:
- "Issuing IMS Connect and IMS type-1 commands from the CONSOLE dialog" on page 165

Users who have the appropriate level of authority can use the IMS Connect Extensions ISPF command shell to issue IMS Connect WTOR commands, IMS Connect z/OS commands, and IMS type-1 commands from IMS Connect Extensions.
Issuing IMS type-1 commands

Use the **IMS Commands** tab in the IMS Connect Extensions Operations Console to issue IMS type-1 commands directly to an IMS data store.

**About this task**

You can run any IMS type-1 command from the command shell.

**Procedure**

1. Open an IMS Connect system using the **Navigation** view. See "Browsing systems" on page 184.

   **Note:** You must select an IMS Connect system, not a group.

2. Select the **IMS Commands** tab. The **IMS Commands** tab is located at the bottom of the editor.

3. Select an IMS type-1 command (or type your own) from the **IMS Command** dropdown.

4. Replace any parameters in the command with your desired values as required. Parameters are shown as lowercase text enclosed in angle brackets (< and >). For example, in the command `/DISPLAY DB <dbname>`, replace `<dbname>` with the desired database.

5. In the **Target Datastores** tab, select the IMS data store to issue the command to.

**Tips:**
a. To refresh the list of IMS data stores, right-click in the list and select **Refresh Datastore Statuses**.
b. Use Ctrl-click to select more than one datastore, or Shift-click to add a range of data stores to the selection.

6. To issue the command, click the **Run** button (حمايطة).

---

**Results**

The command output for each data store is displayed under the **Responses** tab. Use the following features to aid your analysis:
• Use the Find Next ( spoiled ) and Find Previous ( spoiled ) buttons in the toolbar to search for a text string in the output.

• Use the Export button ( spoiled ) to export the displayed data to a CSV file.

Note: Security can be implemented for IMS type-1 commands using RACF or the Command Authorization exit routine or both. Refer to the sections on transaction command security and authorizing commands in IMS System Administration.

Related tasks:
“Issuing IMS Connect and IMS type-1 commands from the CONSOLE dialog” on page 165

Users who have the appropriate level of authority can use the IMS Connect Extensions ISPF command shell to issue IMS Connect WTOR commands, IMS Connect z/OS commands, and IMS type-1 commands from IMS Connect Extensions.

Related reference:
“SHELL command” on page 454

The SHELL host command for REXX runs an IMS Connect command or IMS type-1 command on a specified target system or IMS data store. You can use this to automate some procedures such as stopping all the IMS data stores associated with an IMS system across multiple IMS Connect instances.

Message Log

The Message Log tab in the IMS Connect Extensions Operations Console allows you to view information about IMS Connect and IMS Connect Extensions activity.

Note: You must select an IMS Connect system, not a group, to view the Message Log tab.

Use the Message Log to see the following information:
• The status of archiving.
• A list of recent commands and their outcome.
• Information about IMS Connect Extensions configuration.
• Any errors that may have occurred.
The Message Log has the following additional features:

- Use the Export button ( ) to export the displayed data to a CSV file.

- Use the Find Next ( ) and Find Previous ( ) buttons in the toolbar to search for a text string in the output.

Related concepts:

“Browsing the message log for an IMS Connect” on page 164

The message log for an IMS Connect system reveals activity within IMS Connect and IMS Connect Extensions. Use the message log to understand more about the status of archiving, to view error messages, or to display a history of commands that have been issued and their responses.

Related tasks:

“Exporting data to a CSV file” on page 185

Use the Export option to save the data that is currently displayed in the IMS Connect Extensions Operations Console to a file in CSV (comma-separated values).
format.

Related reference:

Chapter 22, “Messages and codes,” on page 487

Use the information in these messages to help you diagnose and solve IMS Connect Extensions problems.
Chapter 10. Automating IMS Connect operations with REXX

The IMS Connect Extensions host command environment for REXX enables IMS Connect Extensions commands to be embedded in REXX programs. This facility allows you to automate IMS Connect operational tasks and to integrate IMS functions from other REXX command interfaces such as the IMS REXX SPOC API and the SDSF REXX programming interface.

Related concepts:
Chapter 19, “IMS Connect Extensions host command environment for REXX,” on page 429

Use the services of the IMS Connect Extensions host command environment for REXX to write REXX execs that can interact with and control IMS Connect Extensions features. You can use these features to query system statistics, suspend and resume routing to an IMS data store, activate a new routing plan, switch IMS Connect Extensions active journal data sets, start a trace, submit IMS Connect commands, and move toward increased automation of your IMS environment.

IMS Connect Extensions host command summary

The IMS Connect Extensions host command environment for REXX supplies several commands to control operations in IMS, IMS Connect, and in IMS Connect Extensions.

The commands provided with the IMS Connect Extensions host command environment for REXX are summarized below. For a complete command reference, see Chapter 19, “IMS Connect Extensions host command environment for REXX,” on page 429.

REXX environment and system connection management

CONNECT command

Connect to an IMS Connect system so that subsequent host commands can be issued to that system.

OPTION command

Specify run-time options for the REXX host command environment. Use this command to adjust message verbosity and for connection management.

WTO command

Issue a Write To Operator message to the console. You can use this to automate responses to certain conditions.

DELAY command

Pause command execution for a specified number of seconds.

Settings and status queries

QUERY command

Retrieve a variety of IMS Connect statistics, status updates, and settings information via IMS Connect Extensions. The following information can be obtained using this command:

- ACEE cache space and accounting statistics.
- Information and statistics about the active IMS Connect Extensions journal.
- The number of messages that are pending a response from IMS (messages that are in a “Waiting for Datastore” state).
- The names of the currently active OTMA routing plan and ODBM routing plan.
- Statistics and status of active sessions in IMS Connect.
- Information and statistics for IMS Connect socket usage.
- Display a variety of information and statistics from the IMS Connect Extensions status monitor. Retrieve information on TCP/IP ports, IMS Connect systems, IMS data stores, user exits, ODBM targets, IMS aliases, MSC links, and remote IMS Connect systems.
- Status and settings for the IMS Connect Extensions trace.

**UPDATE command**

Allows certain settings to be changed both in-memory and in the IMS Connect Extensions repository definition. The following settings can be updated:

- IMS Connect Extensions event collection level. See “Configuring event collection and journals” on page 61.
- Capacity weight rating (CWR) for an IMS data store or ODBM target. See “Setting the capacity of an IMS data store for OTMA workloads” on page 254 and “Setting the capacity of an ODBM target” on page 290.
- Session message limit and threshold. See “Rebalancing sessions across IMS Connect systems” on page 276.

**ADD command**

Add an entry to the in-memory IMS data store table for a data store that is defined to IMS Connect but was not defined to IMS Connect Extensions at system startup. See “Dynamically adding IMS data stores as targets for OTMA routing” on page 274.

**REFRESH command**

Rebuild the in-memory copy of the selected definitions stored in the IMS Connect Extensions repository for the target IMS Connect system. See “Refreshing in-memory definitions from the IMS Connect Extensions repository” on page 150.

**Journals and tracing**

**SWITCH command**

Switch the active journal for an IMS Connect system. See “Managing the IMS Connect Extensions journal” on page 163.

**TRACE command**

Control the IMS Connect Extensions trace. You can specify filter conditions to control which additional event records are written to the active journal data set. See “Using the IMS Connect Extensions trace” on page 93.
Suspend message routing to an IMS data store to drain it of in-progress transactions, or resume routing as required. See “Suspending OTMA workload routing to IMS data stores using drain” on page 273.

**SET command**

Activate or deactivate an OTMA or ODBM routing plan. See “Setting in-memory IMS Connect Extensions definitions” on page 156.

**DRAIN command**

Causes eligible IMS Connect sessions to be marked for closure at the next inbound message. Sessions are closed normally and not canceled. See “Active Sessions” on page 130.

### Issuing IMS commands

**SHELL command**

Runs an IMS Connect command or IMS type-1 command on a specified target system or IMS data store. You can use this to automate some procedures such as stopping all the IMS data stores associated with an IMS system across multiple IMS Connect instances. See “Issuing IMS Connect and IMS type-1 commands from the CONSOLE dialog” on page 165.

### Security

**CLEAR command**

Clears one or more users from the ACEE cache. See “Issuing security commands” on page 155.

---

**REXX automation samples for the IMS Connect Extensions host command environment**

The SCEXSAMP library includes several sample REXX execs that use the IMS Connect Extensions host command environment. As well as performing useful operations tasks, you can use these samples as a guide to developing your own REXX execs that improve automation across your IMS environment.

**Tip:** Member CEXRXCSJ in the SCEXSAMP sample library is a batch job to execute a specified REXX sample. To run the REXX exec, 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.

**IMS Connect Extensions journal switching**

The following members of the sample library relate to switching the IMS Connect Extensions journal. See “Managing the IMS Connect Extensions journal” on page 163.

**CEXRXC01**

Perform an IMS Connect Extensions journal switch for an IMS Connect system.

**Host commands used:** OPTION, CONNECT, SWITCH.
CEXRXC02
Perform a coordinated IMS Connect Extensions journal and IMS log switch.
This REXX switches the active journal for an IMS Connect system and then issues the IMS /SWITCH OLDS command to switch the IMS log.

Host commands used: OPTION, CONNECT, SWITCH, SHELL.

CEXRXC03
Perform a conditional IMS Connect Extensions journal switch.
The operational use case for this REXX sample is to support a routine journal switch procedure, where the switch is dynamically suppressed if either the active journal has only just been switched or is about to be switched anyway.
The process checks the active journal utilization for an IMS Connect system and then switches the active journal. However, the switch will only be performed if the observed active journal utilization falls within a specified range.

Host commands used: OPTION, CONNECT, QUERY, SWITCH.

IMS Connect Extensions trace
The following member of the sample library relates to the IMS Connect Extensions trace. See “Using the IMS Connect Extensions trace” on page 93.

CEXRXC11
Activate, deactivate, or change IMS Connect Extensions trace settings.

Host commands used: OPTION, CONNECT, TRACE, QUERY.

Draining IMS data stores
The following members of the sample library relates to the data store drain feature. See “Suspending OTMA workload routing to IMS data stores using drain” on page 273.

CEXRXC21
Drain an IMS data store.
A drain temporarily suspends routing of messages to a datastore, allowing it to be “drained” of in-progress transactions. The IMS data store is monitored until there are no in-progress transactions remaining, at this point the data store is stopped.
If the timeout value is exceeded and there are still in-progress transactions then a WTO message is generated and a return code of 4 is set.

Host commands used: OPTION, CONNECT, ROUTE, DELAY, WTO, QUERY, SHELL.

CEXRXC22
Drain an OTMA routing list for an IMS Connect system.
The operational use case for this REXX sample is to provide a mechanism to take an IMS system offline from an IMS Connect system with minimum operational impact upon connected clients.
The process drains a routing list and then stops the routing list’s dependent IMS data stores for the given IMS Connect system.
Draining a routing list causes all its dependent IMS data stores to be drained for the given IMS Connect system. That is, routing of messages to the IMS data stores is temporarily suspended, which results in the IMS data stores being “drained” of in-progress transactions.

The IMS data stores in the routing list are monitored until there are no in-progress transactions remaining, at which point the IMS data stores are stopped. However, if a given time out period is exceeded and there are still in-progress transactions outstanding, then a warning WTO message is generated and the process completes with a return code of 4.

Host commands used: OPTION, CONNECT, ROUTE, DELAY, WTO, QUERY, SHELL.

CEXRXC23

Drain all IMS data stores in a nominated OTMA routing list for one or more IMS Connect systems and then issue an IMS shut down command for the related IMS system.

The operational use case for this REXX sample is to provide a mechanism to shut down an IMS system with minimal operational impact on clients connected to the IMS system via one or more IMS Connect systems. The process drains a routing list across a given set of IMS Connect systems, then goes on to shut down the IMS system associated with the nominated data store, or the IMS system associated with the first available data store in the routing list (if the data store to use when issuing the IMS shut down is not specified). Draining a routing list causes all its dependent data stores to be drained for the given IMS Connect system. That is, routing of messages to the data stores is temporarily suspended, which results in the data store being drained of in-progress transactions. By draining a routing list across the set of IMS Connect systems that utilize it, it is possible to effectively drain the workload of an IMS system in preparation for the IMS system’s shut down.

The data stores in the routing list are monitored across one or more IMS Connect systems until there are no in-progress transactions outstanding, at which point the IMS shutdown command is driven. However, if a given time out period is exceeded and there are still in-progress transactions outstanding, then a warning WTO message is generated and the process can optionally complete with return code of 4. Alternatively, the process can be instructed to continue on to the next stage, which is to drive IMS shutdown by specifying the SesnRemain(SHUTDOWN) option.

Host commands used: OPTION, CONNECT, ROUTE, DELAY, WTO, QUERY, SHELL.

CEXRXC26

Drain persistent session. To drain all sessions specify TOKEN(*).

Host commands used: OPTION, CONNECT, DRAIN, QUERY.

Activating OTMA and ODBM routing plans

The following member of the sample library relates to IMS Connect Extensions routing plans. See “Activating OTMA routing rules in IMS Connect” on page 259 and “Activating ODBM routing rules in IMS Connect” on page 294.

CEXRXC31

Activate or deactivate an OTMA or ODBM routing plan and rebuilds the in-memory routing rules and routing lists.

Host commands used: OPTION, CONNECT, SET, REFRESH, QUERY.
Submitting shell commands

The following member of the sample library demonstrates how to issue IMS Connect commands and IMS type-1 commands via REXX. See also “Issuing IMS Connect and IMS type-1 commands from the CONSOLE dialog” on page 165.

CEXRXC40
Submit shell commands.
This REXX demonstrates how to issue IMS Connect commands and IMS type-1 commands via IMS Connect Extensions.

Host commands used: OPTION, CONNECT, SHELL.

Querying status information

The following members of the sample library demonstrate how to obtain status information from IMS Connect Extensions. See also “Status Monitor” on page 125.

CEXRXC42
Obtain a list of active sessions.
This REXX queries an IMS Connect system and returns session details via the REXXOUT data set.

Host commands used: OPTION, CONNECT, QUERY.

CEXRXC43
Obtain monitoring statistics from the IMS Connect Extensions Status Monitor.
This REXX queries an IMS Connect system and returns specified Status Monitor details via the REXXOUT data set.

Host commands used: OPTION, CONNECT, QUERY.

CEXRXC44
Query the status of security credentials in the 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 statistics. The first relates to the space usage in the ACEE cache. The second presents statistics on different types of cache services requests.

Host commands used: OPTION, CONNECT, QUERY.
See also CEXRXC51

CEXRXC45
Query the status of the IMS Connect Extensions trace.
This REXX queries an IMS Connect system for the status of the IMS Connect Extensions trace. If the trace is active, the current tracing options are also returned.

Host commands used: OPTION, CONNECT, QUERY.
See also CEXRCX11

CEXRXC46
Obtain utilization metrics for the active IMS Connect Extensions journal for an IMS Connect system.
This REXX queries an IMS Connect system and returns utilization details for the active journal data set via the REXXOUT data set.

Host commands used: OPTION, CONNECT, QUERY.

See also IMS Connect Extensions journal switching.

CEXRXC47
Obtain socket utilization metrics for an IMS Connect system.

This REXX queries an IMS Connect system and returns socket utilization details via the REXXOUT data set.

Host commands used: OPTION, CONNECT, QUERY.

See also Managing IMS Connect socket usage.

Managing the ACEE cache

The following member of the sample library relates to management of the ACEE cache. See “Caching user credentials to improve performance” on page 301.

CEXRXC51
Clear security credentials from the ACEE cache.

This REXX clears the ACEE cache of a specific user ID (for example, UID1234) or for a user ID mask (for example, UID*).

Host commands used: OPTION, CONNECT, QUERY, CLEAR.

See also CEXRXC44.

Workload balancing, session message limits, and event collection levels

The following members of the sample library relates to workload balancing, session message limits, and event collection levels.

CEXRXC61
Modify the capacity weight rating (CWR) of an IMS data store or ODBM target.

Use this feature to redistribute workloads in response to periods of changing demand.

Host commands used: OPTION, CONNECT, UPDATE.

See “Setting the capacity of an IMS data store for OTMA workloads” on page 254 and “Setting the capacity of an ODBM target” on page 290.

CEXRXC62
Modify the IMS Connect Extensions event collection level for an IMS Connect system.

The event collection level determines which events are recorded in the IMS Connect Extensions journal.

Host commands used: OPTION, CONNECT, UPDATE.

See “Overriding IMS Connect system settings” on page 158.

CEXRXC63
Modify the session message limit for an IMS Connect system.
The session message limit specifies the number of input messages a persistent session can receive before IMS Connect Extensions automatically closes the session.

**Host commands used:** `OPTION`, `CONNECT`, `UPDATE`.

See “Rebalancing sessions across IMS Connect systems” on page 276.

CEXRXC64

Incremental capacity weight rating (CWR) update for an IMS data store.

This REXX updates the capacity weight rating of an IMS data store by a given static increment or decrement value and ensures that the new value stays within a specified range.

**Host commands used:** `OPTION`, `CONNECT`, `UPDATE`, `QUERY`.

See “Setting the capacity of an IMS data store for OTMA workloads” on page 254.

### Adding definitions dynamically

The following member of the sample library relates to dynamically adding IMS data store definitions. See “Dynamically adding IMS data stores as targets for OTMA routing” on page 274.

CEXRXC71

Add an IMS data store.

This REXX can be used to add an IMS data store to an IMS Connect Extensions system in the following circumstances:

- When a IMS system has been started with an IMS data store that is not in the IMS Connect Extensions repository.
- When a IMS system has had an IMS data store dynamically added that is not in the IMS Connect Extensions repository.

Before you run this program you should first create an IMS data store definition in the IMS Connect Extensions definitions repository. This can be done in the ISPF dialog or by using the definition maintenance utility. You will also need to manually add the new datastore to any routing list it needs to appear in.

**Host commands used:** `OPTION`, `CONNECT`, `ADD`.

### Managing IMS Connect socket usage

The following member of the sample library relates to dynamically managing socket usage in IMS Connect. See “Status Monitor” on page 125 and “Active Sessions” on page 130.

CEXRXC81

Identify and optionally respond to the scenario where an IMS Connect system is approaching MAXSOC.

The operational use case for this REXX sample is to help identify, and optionally facilitate an automated response to resolve, the scenario where an IMS Connect system is approaching MAXSOC.

The process obtains the current socket utilization for an IMS Connect system, where the value is represented as a percentage of the system’s MAXSOC value. If the system’s current socket utilization is found to cross a specified threshold, then a process is undertaken to identify cancelable,
persistent OTMA sessions that have been idle for longer than a specifiable session wait time threshold. These sessions will be reported and can optionally be canceled.

Host commands used: OPTION, CONNECT, QUERY, DELAY, SHELL.

Writing your first REXX execs with IMS Connect Extensions

The following examples show the structure of a REXX exec using commands from the IMS Connect Extensions host command environment for REXX.

From a REXX exec you can establish connections to IMS Connects using the CONNECT command. You can issue IMS Connect Extensions 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.

Example REXX for a single system ACEE cache statistics query

This sample REXX exec shows how to connect to an IMS Connect system, run a query to obtain ACEE cache statistics, and then display the results.

This simple REXX exec performs the following actions:

1. Uses the LINK host command environment to link to IMS Connect Extensions.
2. Uses the OPTION command with MSGLEVEL=VERBOSE to display verbose message output. The execution start time is displayed, and any host commands that are issued will be echoed in the command output.
3. Uses the CONNECT host command to connect to an IMS Connect system HWSOPGS1.
4. Uses the QUERY host command with TYPE=ACEE_CACHE to query the state of the ACEE cache.
5. Formats and displays the fields returned by the QUERY command. To learn more about the response structure, see “REXX variables” on page 234.
6. Cleans up support routines and removes the IMS Connect Extensions entry from the host command environment table.

To learn more about caching of user credentials in IMS Connect Extensions, see “Caching user credentials to improve performance” on page 301.
REXX exec

parse arg PARM
address LINK "CEXRXENV INIT"
| 1 |
address CEX "OPTION MSGLVL=VERBOSE"
| 2 |
address CEX "CONNECT HOST=SRV2,PORT=4199,HWSID=HWSOPGS1"
| 3 |

/* Query Command */
address CEX "QUERY TYPE=ACEE_CACHE"
call CheckCEXResponse
| 4 |

/* Process stem variables from Query command */
say 'IMS Connect Extensions ACEE Cache statistics:

do r=4 to CEX.RESPONSE.FIELD.0
 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"
| 5 |

return
| 6 |

Result

A successful execution of this REXX exec is displayed in the figure below.
Tip: The CEXRXC51 member in the SCEXSAMP library contains a detailed example of how to extend this REXX example to clear the ACEE cache. For more information, see “Managing the ACEE cache” on page 227.

Related reference:

“OPTION command” on page 435
The OPTION host command for REXX is used to specify run-time options for the REXX host command environment. Use this command to adjust message verbosity and for connection management.

“CONNECT command” on page 431
The CONNECT host command for REXX connects to an IMS Connect system so that subsequent host commands can be issued to that system.

“QUERY TYPE=ACEE_CACHE command” on page 436
The QUERY TYPE=ACEE_CACHE host command for REXX retrieves information about the caching of security credentials in IMS Connect to improve performance.

Related information:

“FUN1003I” on page 534
Processing event at time

“CEX5043I” on page 495
REXX host command and batch command control statements

Example REXX query of two systems for messages with responses pending

This sample REXX exec shows how to combine the results of a query run on two IMS Connect systems that returns the number of messages that are pending a response from IMS.

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 command 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.

The **STEM** keywords in the **QUERY** commands are used to distinguish between the variables returned from each system.

This REXX exec performs the following actions:

1. Uses the **CONNECT** host command to connect to the first IMS Connect system HWSOPGS1. The connection is assigned a connection ID of SYS1 using the **CONID=SYS1** parameter.
2. Uses the **CONNECT** host command to connect to the second IMS Connect system HWSOPGS2. The connection is assigned a connection ID of SYS2 using the **CONID=SYS2** parameter.
3. Uses the **QUERY** host command with **TYPE=PENDING_RESPONSES** to query pending responses on the IMS Connect system with connection ID SYS1. The data returned is stored in variables using the stem GS1 (**STEM=GS1**).
4. Uses the **QUERY** host command with **TYPE=PENDING_RESPONSES** to query pending responses on the IMS Connect system with connection ID SYS2. The data returned is stored in variables using the stem GS2 (**STEM=GS2**).
5. Formats and displays the fields returned by the **QUERY** command. Responses from each system are prefixed by the stem values GS1 and GS2. To learn more about the response structure, see "REXX variables" on page 234.

### REXX exec

```rexx
parse arg PARM

address LINK "CEXRXENV INIT"

address CEX "OPTION MSGLEV=QUIET"

/* Define each connection with a unique **CONID** */
address CEX "CONNECT HOST=SRV1,PORT=4199,HWSID=HWSOPGS1," "CONID=SYS1"

address CEX "CONNECT HOST=SRV2,PORT=4299,HWSID=HWSOPGS2," "CONID=SYS2"

/* Query command using **CONID** and unique **STEM** name */
address CEX "QUERY TYPE=PENDING_RESPONSES,DSLIST=RLIST1," "CONID=SYS1," "STEM=GS1"

address CEX "QUERY TYPE=PENDING_RESPONSES,DSLIST=RLIST1," "CONID=SYS2," "STEM=GS2"
```

232 IBM IMS Connect Extensions for z/OS: User's Guide
Result

A successful execution of this REXX exec is displayed in the figure below. The report combines the results of the two queries to display the routing status and the number of message pending for each of the named IMS data stores.

```
/* 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
  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
```

Tip: The CEXRXC21, CEXRXC22, and CEXRXC23 members in the SCEXSAMP library contain detailed examples of how to extend this REXX example to shut down and restart an IMS system once the necessary IMS data stores have been drained of messages pending responses. For more information, see "Draining IMS data stores" on page 224.

Related reference:

"OPTION command" on page 435

The **OPTION** host command for REXX is used to specify run-time options for the REXX host command environment. Use this command to adjust message verbosity and for connection management.

"CONNECT command" on page 431

The **CONNECT** host command for REXX connects to an IMS Connect system so that subsequent host commands can be issued to that system.
The `QUERY TYPE=PENDING RESPONSES` host command for REXX returns the number of messages that are pending a response from IMS (that is, messages that are in a Waiting for Datastore state).

**REXX variables**

Each host command that is issued by IMS Connect Extensions 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 8. Top-level REXX variables**

<table>
<thead>
<tr>
<th>Variable name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>stem.COMMAND</code></td>
<td>The command.</td>
</tr>
<tr>
<td><code>stem.MESSAGE.0</code></td>
<td>The number of CEXPRINT messages.</td>
</tr>
<tr>
<td><code>stem.MESSAGE.n</code></td>
<td>The messages which the batch command utility would have written to CEXPRINT.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> If MSGLEVEL=VERBOSE is specified then the contents of MESSAGE.n will automatically appear in the output.</td>
</tr>
<tr>
<td><code>stem.PRINT.0</code></td>
<td>The number of MSGOUT lines.</td>
</tr>
<tr>
<td><code>stem.PRINT.n</code></td>
<td>The lines of output which the batch command utility would have written to MSGOUT.</td>
</tr>
</tbody>
</table>

2. The following response variables are common to all console commands.

   **Table 9. Console command response variables**

<table>
<thead>
<tr>
<th>Variable name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>stem.RESPONSE.FIELD.0</code></td>
<td>The number of response field variables for this request.</td>
</tr>
<tr>
<td><code>stem.RESPONSE.FIELDNAME.n</code></td>
<td>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.</td>
</tr>
<tr>
<td><code>stem.RESPONSE.FIELDDESC.n</code></td>
<td>The description of the response field variable.</td>
</tr>
<tr>
<td><code>stem.RESPONSE.RET</code></td>
<td>The return code associated with the response from console services.</td>
</tr>
<tr>
<td><code>stem.RESPONSE.RSN</code></td>
<td>The reason code associated with the response from console services.</td>
</tr>
<tr>
<td><code>stem.RESPONSE.TEXT</code></td>
<td>A text explanation for the reason code.</td>
</tr>
</tbody>
</table>

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`
Chapter 11. 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 35
  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

Before you can issue IMS Connect WTOR commands, IMS Connect z/OS commands, IMS type-1 commands, and IMS Connect Extensions commands to an IMS Connect system, you must first activate access control. For security reasons, command access to IMS Connect systems from IMS Connect Extensions is deactivated by default.

About this task

This procedure activates command access for the following types of commands:

- IMS Connect WTOR commands, IMS Connect z/OS commands, and IMS type-1 commands. See “Issuing IMS Connect and IMS type-1 commands from the CONSOLE dialog” on page 165.
- IMS Connect Extensions commands. See “Issuing IMS Connect Extensions commands from the Commands panel” on page 145.

Procedure

To activate command access for an IMS Connect system:

1. From the primary menu of the IMS Connect Extensions ISPF dialog, select the definition repository used by the IMS Connect system for which you want to activate commands using the Definitions repository field.
2. Select option 1 Definitions.
3. Select option 1 System Definitions.
4. Next to the IMS Connect system, enter line action S. The System Definition panel is displayed.

5. Configure the IMS Connect system definition as follows:
   a. Activate command access by entering a / (slash) next to the **Activate Commands** option.
   b. Optional: To use an external security manager (ESM) to authorize users or to support issuing of IMS type-1 commands from the command shell, enter / (slash) next to the **Activate Access Control** option.

   **Note:** 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 238 for details.

c. Optional: To use PassTickets instead of plain text passwords, enter / (slash) next to the **Enable PassTicket Generation**.

   **Important:** 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 240 for details.
6. 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.

**Related reference:**

"Defining IMS Connect systems" on page 320

Use the System Definitions panel to create a new definition for an IMS Connect and to specify 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. To access this ISPF panel, select option 1.1 **System Definitions** from the IMS Connect Extensions primary menu.
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 240 for a sample RACF job to set these profiles.

### Table 10. Resource name and RACF authority for IMS Connect and IMS Connect Extensions commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Type</th>
<th>Verb</th>
<th>Object</th>
<th>Resource name</th>
<th>Authority</th>
</tr>
</thead>
<tbody>
<tr>
<td>CEX</td>
<td>ADD</td>
<td>Datastore</td>
<td>CEX.applid.ADD.DS</td>
<td>UPDATE</td>
<td></td>
</tr>
<tr>
<td>CEX</td>
<td>ADD</td>
<td>User exit</td>
<td>CEX.applid.ADD.EXIT</td>
<td>UPDATE</td>
<td></td>
</tr>
<tr>
<td>CEX</td>
<td>CLEAR</td>
<td>ACEE (user profile)</td>
<td>CEX.applid.CLR.ACEE</td>
<td>UPDATE</td>
<td></td>
</tr>
<tr>
<td>CEX</td>
<td>DELETE</td>
<td>User exit</td>
<td>CEX.applid.DELE.EXIT</td>
<td>UPDATE</td>
<td></td>
</tr>
<tr>
<td>CEX</td>
<td>DRAIN</td>
<td>Session</td>
<td>CEX.applid.DRN.SESSION</td>
<td>UPDATE</td>
<td></td>
</tr>
<tr>
<td>VIEWIA</td>
<td>HWS</td>
<td>QUERY</td>
<td>Alias</td>
<td>CEX.applid.QRY.ALIAS</td>
<td>READ</td>
</tr>
<tr>
<td>VIEWDS</td>
<td>HWS</td>
<td>QUERY</td>
<td>Datastore</td>
<td>CEX.applid.QRY.DS</td>
<td>READ</td>
</tr>
<tr>
<td>VIEWIP</td>
<td>HWS</td>
<td>QUERY</td>
<td>IMSplex</td>
<td>CEX.applid.QRY.IMSPLEX</td>
<td>READ</td>
</tr>
<tr>
<td>VIEWMSC</td>
<td>HWS</td>
<td>QUERY</td>
<td>MSC physical link</td>
<td>CEX.applid.QRY.MSC</td>
<td>READ</td>
</tr>
<tr>
<td>VIEWPORT</td>
<td>HWS</td>
<td>QUERY</td>
<td>Port</td>
<td>CEX.applid.QRY.PORT</td>
<td>READ</td>
</tr>
<tr>
<td>VIEWRMT</td>
<td>HWS</td>
<td>QUERY</td>
<td>Remote IMS Connect instance</td>
<td>CEX.applid.QRY.RMT</td>
<td>READ</td>
</tr>
<tr>
<td>VIEWHWS</td>
<td>HWS</td>
<td>QUERY</td>
<td>System</td>
<td>CEX.applid.QRY.SYSTEM</td>
<td>READ</td>
</tr>
<tr>
<td>VIEWUOR</td>
<td>HWS</td>
<td>QUERY</td>
<td>Unit of recovery (UOR)</td>
<td>CEX.applid.QRY.UOR</td>
<td>READ</td>
</tr>
<tr>
<td>Command</td>
<td>Type</td>
<td>Verb</td>
<td>Object</td>
<td>Resource name</td>
<td>Authority</td>
</tr>
<tr>
<td>------------</td>
<td>---------</td>
<td>--------</td>
<td>-------------------------------</td>
<td>---------------------------</td>
<td>-----------</td>
</tr>
<tr>
<td>CEX</td>
<td>REFRESH</td>
<td>IP address rule</td>
<td>CEX.applid.REF.IPADDR</td>
<td>UPDATE</td>
<td></td>
</tr>
<tr>
<td>CEX</td>
<td>RELOAD</td>
<td>User exit</td>
<td>CEX.applid.RLD.EXIT</td>
<td>UPDATE</td>
<td></td>
</tr>
<tr>
<td>CEX</td>
<td>ROUTE</td>
<td>Datastore</td>
<td>CEX.applid.RTE.DS</td>
<td>UPDATE</td>
<td></td>
</tr>
<tr>
<td>CEX</td>
<td>SET</td>
<td>Application</td>
<td>CEX.applid.SET.APPL</td>
<td>UPDATE</td>
<td></td>
</tr>
<tr>
<td>CEX</td>
<td>SET</td>
<td>Datastore</td>
<td>CEX.applid.SET.DS</td>
<td>UPDATE</td>
<td></td>
</tr>
<tr>
<td>CEX</td>
<td>SET</td>
<td>Datastore group</td>
<td>CEX.applid.SET.DSG</td>
<td>UPDATE</td>
<td></td>
</tr>
<tr>
<td>CEX</td>
<td>SET</td>
<td>Routing plan</td>
<td>CEX.applid.SET.PLAN</td>
<td>UPDATE</td>
<td></td>
</tr>
<tr>
<td>CEX</td>
<td>SET</td>
<td>Publisher</td>
<td>CEX.applid.SET.PUB</td>
<td>UPDATE</td>
<td></td>
</tr>
<tr>
<td>CEX</td>
<td>SET</td>
<td>System definition</td>
<td>CEX.applid.SET.SYSTEM</td>
<td>UPDATE</td>
<td></td>
</tr>
<tr>
<td>CEX</td>
<td>REFRESH</td>
<td>Transaction</td>
<td>CEX.applid.REF.TRAN</td>
<td>UPDATE</td>
<td></td>
</tr>
<tr>
<td>STARTIA</td>
<td>HWS</td>
<td>START</td>
<td>Alias</td>
<td>CEX.applid.STA.ALIAS</td>
<td>UPDATE</td>
</tr>
<tr>
<td>STARTDS</td>
<td>HWS</td>
<td>START</td>
<td>Datastore</td>
<td>CEX.applid.STA.DS</td>
<td>UPDATE</td>
</tr>
<tr>
<td>CEX</td>
<td>ENABLE</td>
<td>User exit</td>
<td>CEX.applid.STA.EXIT</td>
<td>UPDATE</td>
<td></td>
</tr>
<tr>
<td>STARTIP</td>
<td>HWS</td>
<td>START</td>
<td>IMSplex</td>
<td>CEX.applid.STA.IMSPLEX</td>
<td>UPDATE</td>
</tr>
<tr>
<td>STAMSC</td>
<td>HWS</td>
<td>START</td>
<td>MSC logical link</td>
<td>CEX.applid.STA.MSC</td>
<td>UPDATE</td>
</tr>
<tr>
<td>SETAUTO ON</td>
<td>HWS</td>
<td>START</td>
<td>ODBM routing</td>
<td>CEX.applid.STA.OAUTO</td>
<td>UPDATE</td>
</tr>
<tr>
<td>STARTOD</td>
<td>HWS</td>
<td>START</td>
<td>ODBM</td>
<td>CEX.applid.STA.ODBM</td>
<td>UPDATE</td>
</tr>
<tr>
<td>STARTPT</td>
<td>HWS</td>
<td>START</td>
<td>Port</td>
<td>CEX.applid.STA.PORT</td>
<td>UPDATE</td>
</tr>
<tr>
<td>SETPWMC ON</td>
<td>HWS</td>
<td>START</td>
<td>Mixed-case passwords</td>
<td>CEX.applid.STA.PWMC</td>
<td>UPDATE</td>
</tr>
<tr>
<td>SETRACF ON</td>
<td>HWS</td>
<td>START</td>
<td>RACF</td>
<td>CEX.applid.STA.RACF</td>
<td>UPDATE</td>
</tr>
<tr>
<td>SETRMT</td>
<td>HWS</td>
<td>START</td>
<td>Remote IMS Connect instance</td>
<td>CEX.applid.STA.RMT</td>
<td>UPDATE</td>
</tr>
<tr>
<td>SETRRS ON</td>
<td>HWS</td>
<td>START</td>
<td>RRS</td>
<td>CEX.applid.STA.RRS</td>
<td>UPDATE</td>
</tr>
<tr>
<td>HWS</td>
<td>START</td>
<td>System</td>
<td>CEX.applid.STA.SYSTEM</td>
<td>UPDATE</td>
<td></td>
</tr>
<tr>
<td>SETUIDC ON</td>
<td>HWS</td>
<td>START</td>
<td>ACEE cache</td>
<td>CEX.applid.STA.UIDCACHE</td>
<td>UPDATE</td>
</tr>
<tr>
<td>STOPIA</td>
<td>HWS</td>
<td>STOP</td>
<td>Alias</td>
<td>CEX.applid.STO.ALIAS</td>
<td>UPDATE</td>
</tr>
<tr>
<td>STOPCLNT</td>
<td>HWS</td>
<td>CANCEL</td>
<td>Client session</td>
<td>CEX.applid.STO.CLIENT</td>
<td>UPDATE</td>
</tr>
<tr>
<td>STOPDS</td>
<td>HWS</td>
<td>STOP</td>
<td>Datastore</td>
<td>CEX.applid.STO.DS</td>
<td>UPDATE</td>
</tr>
<tr>
<td>CEX</td>
<td>DISABLE</td>
<td>User exit</td>
<td>CEX.applid.STO.EXIT</td>
<td>UPDATE</td>
<td></td>
</tr>
<tr>
<td>STOPIP</td>
<td>HWS</td>
<td>STOP</td>
<td>IMSplex</td>
<td>CEX.applid.STO.IMSPLEX</td>
<td>UPDATE</td>
</tr>
<tr>
<td>STOLINK</td>
<td>HWS</td>
<td>STOP</td>
<td>MSC logical link</td>
<td>CEX.applid.STO.LINK</td>
<td>UPDATE</td>
</tr>
<tr>
<td>STOMSC</td>
<td>HWS</td>
<td>STOP</td>
<td>MSC physical link</td>
<td>CEX.applid.STO.MSC</td>
<td>UPDATE</td>
</tr>
<tr>
<td>SETAUTO OFF</td>
<td>HWS</td>
<td>STOP</td>
<td>ODBM routing</td>
<td>CEX.applid.STO.OAUTO</td>
<td>UPDATE</td>
</tr>
<tr>
<td>STOPOD</td>
<td>HWS</td>
<td>STOP</td>
<td>ODBM</td>
<td>CEX.applid.STO.ODBM</td>
<td>UPDATE</td>
</tr>
<tr>
<td>STOPPORT</td>
<td>HWS</td>
<td>STOP</td>
<td>Port</td>
<td>CEX.applid.STO.PORT</td>
<td>UPDATE</td>
</tr>
<tr>
<td>CEX</td>
<td>STOP</td>
<td>Publisher client</td>
<td>CEX.applid.STO.PUB</td>
<td>UPDATE</td>
<td></td>
</tr>
<tr>
<td>SETPWMC OFF</td>
<td>HWS</td>
<td>STOP</td>
<td>Mixed-case passwords</td>
<td>CEX.applid.STO.PWMC</td>
<td>UPDATE</td>
</tr>
<tr>
<td>SETRACF OFF</td>
<td>HWS</td>
<td>STOP</td>
<td>RACF</td>
<td>CEX.applid.STO.RACF</td>
<td>UPDATE</td>
</tr>
<tr>
<td>STORMT</td>
<td>HWS</td>
<td>STOP</td>
<td>Remote IMS Connect instance</td>
<td>CEX.applid.STO.RMT</td>
<td>UPDATE</td>
</tr>
<tr>
<td>SETRRS OFF</td>
<td>HWS</td>
<td>STOP</td>
<td>RRS</td>
<td>CEX.applid.STO.RRS</td>
<td>UPDATE</td>
</tr>
<tr>
<td>STOSCLN</td>
<td>HWS</td>
<td>STOP</td>
<td>Send client</td>
<td>CEX.applid.STO.SCLN</td>
<td>UPDATE</td>
</tr>
<tr>
<td>CLOSEHWS</td>
<td>HWS</td>
<td>STOP</td>
<td>System</td>
<td>CEX.applid.STO.SYSTEM</td>
<td>UPDATE</td>
</tr>
</tbody>
</table>

Chapter 11. Command access in IMS Connect Extensions  239
Table 10. Resource name and RACF authority for IMS Connect and IMS Connect Extensions commands (continued)

<table>
<thead>
<tr>
<th>Command</th>
<th>Type</th>
<th>Verb</th>
<th>Object</th>
<th>Resource name</th>
<th>Authority</th>
</tr>
</thead>
<tbody>
<tr>
<td>SETUIDC OFF</td>
<td>HWS</td>
<td>STOP</td>
<td>ACEE cache</td>
<td>CEX.&lt;applid&gt;.STUIDCACHE</td>
<td>UPDATE</td>
</tr>
<tr>
<td>CEX TRACE</td>
<td>HWS</td>
<td>TRACE</td>
<td>Port on/off</td>
<td>CEX.&lt;applid&gt;.TRC.PORT</td>
<td>UPDATE</td>
</tr>
<tr>
<td>RECORDER</td>
<td>HWS</td>
<td>TRACId</td>
<td>Recorder on/off</td>
<td>CEX.&lt;applid&gt;.TRC.RECORDER</td>
<td>UPDATE</td>
</tr>
<tr>
<td>CEX UPDATE</td>
<td>CEX</td>
<td>UPDATE</td>
<td>Datastore</td>
<td>CEX.&lt;applid&gt;.UPD.DS</td>
<td>UPDATE</td>
</tr>
<tr>
<td>CEX UPDATE</td>
<td>CEX</td>
<td>UPDATE</td>
<td>System</td>
<td>CEX.&lt;applid&gt;.UPD.SYSTEM</td>
<td>UPDATE</td>
</tr>
</tbody>
</table>

Example: typical access control configuration

IMS Connect Extensions provides a sample job, CEXRACF, that implements IMS Connect Extensions resource profiles in RACF.

The following is the SYSIN deck for this job:

```bash
/* ------------------------ */
/* 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 */
ADDGROUPEXPER
/* Operations Supervisor Group */
ADDGROUPEXOPSUP
/* Administrator Group */
ADDGROUPEXADMIN
/* ---------- */
/* 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 IKJTSOxx, and add CEXPTGEN to the following lists:

```plaintext
AUTHPGM NAMES(
  CEXPTGEN
)
```

and:

```plaintext
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. Managing IMS Connect workloads

These topics describe how to configure IMS Connect Extensions for optimal management of IMS Connect workloads.

Topics:

- Chapter 12, “OTMA workload routing in IMS Connect,” on page 245
- Chapter 13, “OTMA operations management and tuning,” on page 273
- Chapter 14, “ODBM workload routing in IMS Connect,” on page 279
- Chapter 15, “Enhancing IMS Connect security,” on page 301
- Chapter 16, “Customizing IMS Connect message translation,” on page 311
Chapter 12. OTMA workload routing in IMS Connect

OTMA rules-based routing in IMS Connect Extensions allows you to route and distribute OTMA workload received by IMS Connect between available IMS data stores.

OTMA rules-based routing in IMS Connect Extensions 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, OTMA workload can be dynamically redirected to pools of suitable data stores according to OTMA 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.

OTMA rules-based routing in IMS Connect Extensions can provide operational adaptability, optimized performance, workload balancing, and higher availability.

**Operational adaptability and transactional continuity**

IMS Connect Extensions host command environment for REXX and configuration options allow you to automate responses to changing IMS data store 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 data stores 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 IMS data stores 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 preferentially selecting the local data store (primary) for a given destination ID. When the local data store is not available, the workload can be automatically redirected to remote IMS data stores. Alternatively, messages can be distributed to a pool of IMS data stores and each of these data stores can be configured with a capacity weight rating (CWR) to give preference to IMS systems with greater capacity. OTMA rules-based routing allows you to more easily introduce parallelism in OTMA by employing multiple data stores that point to the same IMS system. This type of parallelism can significantly improve performance.

**IMS data store workload balancing**

IMS Connect Extensions can redirect incoming requests among data stores
according to their processing capacity. Data stores with greater capacity receive more transactions. You specify the relative weighting of each data store in the data store definition.

**High availability**
When a data store is not available, IMS Connect Extensions dynamically removes it from the candidate list and routes messages to other available data stores based on their capacity weight ratings (CWR).

---

**Principles of OTMA rules-based routing**

IMS Connect directs messages between its distributed clients and IMS resources. It does this by passing request messages to the IMS data store specified in the IRM header and receiving response messages that are returned to the originating TCP/IP client. OTMA workload routing in IMS Connect Extensions is the process where OTMA routing rules are defined that dynamically alter the incoming target IMS data store specified in an IRM message to route it to one or more IMS data stores selected from a list of candidates. If desired, you can also specify additional qualifying rules that route certain transaction codes to a different list of target IMS data stores.

OTMA rules-based routing works in the following ways:

**The data store ID in the IRM message (IRM_IMSDestId) is used to find a matching OTMA routing rule**
Rules are matched in IMS Connect Extensions based on the value supplied in the IRM_IMSDestId field of the incoming message. Because the field simply contains a string, the data store ID does not have to be a physical data store defined to the IMS Connect system. Instead, it can be a generic value that might represent 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 IMS data store name selected from the list of data stores referenced by the OTMA routing rule.

**OTMA routing lists are used to define the list of candidate IMS data stores**
For each type of IRM received by IMS Connect, you can specify a target OTMA routing list and an optional fallback OTMA routing list. Messages are routed to IMS data stores in the target OTMA routing list first. The fallback list contains data stores that can be used if the data stores in the target list are unavailable.

You can specify different OTMA routing lists for each message type. For example, you could configure one list of candidates for transactional (send-receive) messages and another list of candidates for asynchronous messages.

**OTMA 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 data stores. If you specify a qualifying rule without a matching master rule in the repository, an implied routing rule will be generated internally at run time.
IMS Connect Extensions checks for the name of the active OTMA routing plan

Only one OTMA routing plan can be active on a system at a time. Routing rules that are assigned to other routing plans are not considered.

A routing rule that is not assigned to a routing plan will always be in effect, except when it is overridden by a routing rule that contains a more specific message matching condition.

Define sets of rules for all systems, or develop customized rules for each

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.

You can also have one master rule for a given DestID, as well as optional rules that specify different target IMS data stores for a specified list of transaction codes.

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.

Use CEXCTLIN control options to set the default routing behavior for an IMS Connect

The control input data set allows you to set several behaviors for OTMA routing. The input data set is specified through an optional CEXCTLIN DD statement in the IMS Connect startup job. These options take effect when IMS Connect Extensions restarts. Use these options to control the behavior of IMS Connect Extensions when an IMS data store is experiencing a flood condition, when there are no valid IMS data store destinations, and when IMS Connect Extensions receives a message that does not match any of its current routing rules.

Developing OTMA routing rules for IMS Connect

To route OTMA workloads in IMS Connect, you’ll need to create one or more OTMA routing rules that depend on several supporting definitions in the IMS Connect Extensions repository. The definitions you build will depend on the routing scenario you would like to create.

OTMA routing rules in IMS Connect Extensions are flexible in that they allow you to configure your systems in a variety of ways. Whilst building a set of OTMA routing rules will ultimately depend on an understanding of the nature of your workload, you will essentially use the following procedures to configure your rules:

1. **Create an OTMA routing rule to route workload to one or more IMS data stores.**

   Use this procedure first to create your initial routing rule and its supporting definitions for a single IMS Connect, a group of IMS Connect systems, or all systems. This procedure will help you achieve the following routing outcomes:
• Route OTMA workloads dynamically to a pool of IMS data stores based on the IMS data store destination ID in the request message.
• Redirect OTMA workload intended for a specific IMS data store to a different data store.
• Redirect OTMA workload for specific message types or specific transactions only.

To create an initial routing rule, see "Routing OTMA workload to one or more IMS data stores." on page 248.

2. Optional: Configure your routing rule with fallback IMS data stores.
   After you have created your initial routing rule, use this optional procedure to configure the rule to route workload to a list of fallback IMS data stores if primary target data stores in the rule are unavailable. This can be useful as a backup for your primary data stores, or when you need to perform maintenance of the primary IMS data stores.

   To configure a rule with fallback data stores, see "Routing OTMA workload to a list of fallback IMS data stores” on page 252.

3. Optional: Set the relative capacities of each of your IMS data stores.
   After you have created your initial routing rule, use this optional procedure to balance workload across multiple IMS data stores based on the data stores processing capacity. This procedure will help you set the initial capacity weight rating (CWR) for an IMS data store which allows you to reduce (or increase) the processing capacity of a data store, or set a custom configuration that balances workload accordingly. Once an initial value has been set, you can dynamically adjust the capacity weight rating using the Operations Console for IBM Explorer for z/OS.

   To set capacity weight ratings, see "Setting the capacity of an IMS data store for OTMA workloads” on page 254.

4. Recommended: Group your routing rules by assigning them to a routing plan.
   After you have created your initial routing rule, use this recommended procedure to establish multiple rule sets bound together by a routing plan. Use a routing plan to create sets of rules that work together. The routing plan can then be activated in the following ways:
   • Manually by an operator via the IMS Connect Extensions ISPF dialog or Operations Console for IBM Explorer for z/OS.
   • Automatically using the IMS Connect Extensions host command environment for REXX and the automation tool of your choice.

   To create routing plans, see "Assigning OTMA routing rules to a plan” on page 257.

Routing rules will not take effect until they are activated. To activate your routing rules, see "Activating OTMA routing rules in IMS Connect” on page 259.

Important: You can also control the default routing behavior of IMS Connect Extensions when a request is received but there is no matching OTMA routing rule. For more information, see "Controlling OTMA routing behavior via CEXCTLIN control options” on page 265.

Routing OTMA workload to one or more IMS data stores
You can create an OTMA routing rule to perform dynamic routing of IMS request messages (IRM) containing a specific IMS destination ID (IRM_IMSDESTID) to any IMS data store defined in an OTMA routing list. This allows you to distribute
workload across multiple IMS data stores or to redirect workload to a single data store without changing your client application.

**Before you begin**

Create IMS Connect Extensions repository definitions for the IMS Connect systems and IMS data stores you wish to use in your routing rule. For more information, see “Defining IMS Connect systems” on page 320 and “Defining IMS data stores” on page 339. If you wish to apply your rule to several IMS Connect systems at once, create an IMS Connect system group. For more information, see “Defining system groups” on page 355.

**About this task**

A simple routing rule that directs OTMA workload to a list of IMS data stores for processing is a common first step to implementing OTMA rules-based routing with IMS Connect Extensions.

In this scenario, the OTMA routing rule requires three key ingredients:

- The name of the IMS Connect system (or systems) that require the routing rule.
- The conditions required to invoke the rule; in this case the name of an IMS data store specified in the client request. The IMS data store name is specified by the client in the IMS destination ID field (IRM_IMSDESTID) of the IMS request message (IRM).
- A list of IMS data stores that may be used as routing candidates. In IMS Connect Extensions, this list referred to as an OTMA routing list.

If, for example, you have an IMS Connect system that receives requests that are normally processed exclusively on IMS data store DSLOC01 but you would like to add additional data stores without changing your application, you can create a routing rule (for example, RULE1) that will route this request to an IMS data store contained within a target a list of IMS data stores (GROUP1). The target IMS data store list could be configured to contain your original data store DSLOC01 and several additional data stores, or it might list an entirely different set of IMS data stores as routing candidates (ignoring IMS data store DSLOC01 altogether).
Because incoming messages ar
are keyed to OTMA
routin
rules by the IMS
destination ID (IRM_IMSDestID) set by the client application, it doesn’t matter if
the incoming IMS destination ID is a rea
 IMS data stor
the OTMA
routin
allows you to dir
t the traffic in any way you like. This approach offers a

great deal of flexibilit
and enables client applications to send “logical” IMS
destination IDs rather than physical ones as needs require. Furthermore, this
arrangement allows you to drain workload from an IMS data store and shut it
down without disrupting ongoing operations. For more information, see
“Suspending OTMA workload routing to IMS data stores using drain” on page
273.

Tip: To get a summary of IMS data store names contained within IMS request
messages received by IMS Connect, use IBM IMS Performance Analyzer for z/OS
to generate an IMS Connect form-based transaction transit summary report from
the IMS Connect Extensions journal that contains the ORIGDS (Original IMS data
store) form field. To see how IMS Connect Extensions changes the routing
destination using OTMA rules-based routing, add the TARGDS (Target IMS data
store) field to the report form as well. For more information, see Creating List and
Summary Report Forms in the Guided Tour section of the IBM IMS Performance
Analyzer for z/OS User's Guide.

The following procedure shows you how to create and set up the appropriate
routing rule and data store list as described in the diagram above.

Procedure
1. Create an OTMA routing list definition that defines a collection of IMS data
stores to be used as routing targets in an OTMA routing rule. To do this,
complete the following steps:
   a. From the IMS Connect Extensions primary menu, select option 1
      Definitions.
   b. Select option 10 OTMA Routing Lists.
   c. Type NEW and press Enter.
   d. Type a name and press Enter.
e. Assign one or more IMS data stores to the list. To do this, press the the Prompt function key (F4) on the field beneath the **Datastore** column. To insert new rows for additional data stores, use line action 1. For a complete list of options, use line action / For a complete list of options, see “Defining OTMA routing lists” on page 361.

**Remember:** You must define an IMS data store in IMS Connect and IMS Connect Extensions before you can use it. To define the IMS data store in IMS Connect Extensions, see “Defining IMS data stores” on page 339.

f. When you have finished making changes press the Exit function key (F3). The new routing list is displayed in the OTMA Routing Lists panel.

2. Create an OTMA routing rule definition for an IMS Connect system (or group of systems) that defines the conditions required to route your chosen IRM messages to the OTMA routing list you defined in step 1 on page 250

a. From the IMS Connect Extensions primary menu, select option 1 **Definitions**.

b. Select option 9 **OTMA Routing Rules**.

c. Type **NEW** and press Enter.

d. Type a name and press Enter. The OTMA Routing Rule panel is displayed.

e. Specify the IMS Connect system or systems to which this rule applies in the **Apply rule to** section. Enter 1 **Systems**, 2 **Group**, or 3 **All systems**. Press the Prompt function key (F4) in either the **Systems** or **Group** fields to select from a list of previously defined systems or groups.

f. Specify the IMS destination ID (IRM_IMSDESTID) contained within the incoming IRM message that you wish to route in the **Original Datastore** field. For example, to create a rule that applies to messages that contain an IMS destination ID of DSLOC01, enter DSLOC01.

g. Select one or more message types to which this rule applies by typing Y next to the type followed by the name of the OTMA routing list in the **Target** column. Most users should select **Send-Receive Transactions**, **Send-Only requests**, and **Resume TPIPE requests**. For a complete list of options, see “Defining OTMA routing rules” on page 356.
h. When you have finished making changes, press the Exit function key (F3). The new routing rule is displayed in the OTMA Routing Rules panel.

**What to do next**

Activate the routing rule in IMS Connect. For more information, see “Activating OTMA routing rules in IMS Connect” on page 259.

**Routing OTMA workload to a list of fallback IMS data stores**

An OTMA routing rule can be configured to route workload to a list of fallback IMS data stores if primary target data stores in the rule are unavailable.

**Before you begin**

Create an OTMA routing rule that routes workload to multiple IMS data stores. See “Routing OTMA workload to one or more IMS data stores” on page 248.

**About this task**

Consider the following scenario where an OTMA routing rule RULE2 has been configured to route all workload with an IMS destination ID (IRM_IMSDestID) of IML to an OTMA routing list named GROUP1 containing IMS data store DSLOCAL. Under normal operations, we would prefer all this type of processing to be directed to the data store DSLOCAL. However, should this data store become unavailable, we would like the processing to be redirected to a data store in GROUP2.
The following procedure shows you how to add a fallback OTMA routing list to an existing OTMA routing rule.

**Procedure**

1. Create an OTMA routing list definition for your fallback IMS data stores. For assistance with this task, see “Routing OTMA workload to one or more IMS data stores” on page 248.

   **Tip:** An IMS data store definition can be assigned to more than one OTMA routing list.
2. Modify your OTMA routing rule definition to add the fallback OTMA routing list you defined in step 1 on page 253. In the example below, the rule is triggered on IMS Connect IMSCON1 for any request that contains a destination ID of IMSL.

![Figure 122. OTMA routing list GROUP2 containing IMS data stores DSRMT01, DSRMT02, and DSRMT03](image)

What to do next

Activate the routing rule in IMS Connect. For more information, see “Activating OTMA routing rules in IMS Connect” on page 259.

Setting the capacity of an IMS data store for OTMA workloads

Workload balancing allows IMS Connect Extensions to redirect incoming transaction requests to IMS data stores according to their capacity weight rating (CWR), thereby balancing the transaction workload across multiple IMS systems and ensuring both availability and responsiveness for the remote client.
Before you begin

Create an OTMA routing rule that routes workload to multiple IMS data stores. See “Routing OTMA workload to one or more IMS data stores” on page 248.

About this task

Workload balancing is accomplished using a weighted rotate algorithm that considers the processing capacity of an IMS data store. In IMS Connect Extensions, this processing capacity is realized by assigning a capacity weight rating (CWR) to each IMS data store. The capacity weight rating for an IMS data store is set in the IMS data store definition stored in the IMS Connect Extensions repository.

An IMS data store’s capacity weight rating is relative to other IMS data stores that may be available for processing the incoming message. For example, if IMS data store IMSP1 has a capacity weight rating of 5 and data store IMSP2 has a rating of 15, IMSP2 should, on average, receive three times as many messages as IMSP1. This ratio of 5 to 15 means that IMSP1 receives an average of 25 percent of the workload and IMSP2 receives 75 percent.

Basic rotation in which each data store is deemed to have the same capacity can be achieved by assigning each data store the same capacity weight rating (for example, by assigning a value of 1 to each data store in an OTMA routing list), or by deactivating workload balancing in the IMS Connect system definition. You can keep an IMS data store in reserve by assigning it a capacity of zero (0), making it ineligible as a routing candidate.

IMS Connect Extensions will not route messages to an IMS data store that is unavailable, is experiencing a severe flood condition, or have a capacity weight rating of zero. In this situation, IMS Connect Extensions will distribute the workload across the remaining eligible data stores according to their capacity weight ratings.

Figure 124. IMS Connect and IMS Connect Extensions distributing an OTMA workload over two IMS data stores according to each data store’s capacity weight rating
You can also use capacity weight ratings to automatically redistribute workload in response to changing periods of demand with the IMS Connect Extensions host command environment for REXX. The member CEXRXC61 in the SCEXSAMP library shows how to adjust capacity weights using a REXX exec. See “REXX automation samples for the IMS Connect Extensions host command environment” on page 223.

**Procedure**

1. Set a capacity weight rating (CWR) for every IMS data store definition associated with an OTMA routing rule. To do this, complete the following steps:
   a. From the IMS Connect Extensions primary menu, select option 1 Definitions.
   b. Select option 3 Datastores.
   c. Select the IMS data store using line action $.
   d. Enter a value in the **Capacity weight rating** field. IMS data stores with greater capacity weight rating (CWR) will receive more transactions. See “Defining IMS data stores” on page 339.

   **Note:** A value of zero indicates that the data store will be held in reserve and receive no workload.

   ![Figure 125. Setting the capacity weight rating (CWR) for an IMS data store using the Datastore panel in the IMS Connect Extensions ISPF dialog](image)

   e. When you have finished making changes press the Exit function key (F3).

   Repeat this process for each IMS data store as required.

2. Activate workload balancing in the IMS Connect system definition. To do this, complete the following steps:
   a. From the IMS Connect Extensions primary menu, select option 1 Definitions.
b. Select option 1 System Definitions.

c. To edit the IMS Connect system definition you are interested in, enter line action S next to the name of the system.

d. Enter a slash (/) next to the Activate Workload Balancing option.

Note: If you have not already done so, you must also select the Activate IMS Connect Extensions, Activate Advanced Features, and Activate OTMA rules options.

e. Press the Exit function key (F3) to save and exit the panel.

What to do next

Activate the routing rule in IMS Connect. For more information, see “Activating OTMA routing rules in IMS Connect” on page 259.

Related tasks:

“Managing IMS data stores” on page 204

Use the Datastores and Datastore Groups tab in Status Monitor to perform management tasks on IMS data stores from the IMS Connect Extensions Operations Console. Use this facility to start and stop IMS data stores, suspend and resume routing to a data store, and to dynamically update a data stores capacity weight rating (CWR) in response to periods of changing demand.

Related reference:

“Defining IMS data stores” on page 339

The Datastores panel allows you to define the IMS data stores defined and controlled by IMS Connect in the IMS Connect Extensions repository. To access this ISPF panel, select option 1.3 Datastores from the IMS Connect Extensions primary menu.

Assigning OTMA routing rules to a plan

Routing plans provide a way to group OTMA routing rules that are intended to operate at the same time. You can use routing plans to create different routing configurations that can be dynamically activated as required.
Before you begin

Before you can activate OTMA routing rules in IMS Connect, you must first create some rules. See “Routing OTMA workload to one or more IMS data stores” on page 248.

About this task

A routing rule that is not assigned to a routing plan will always be in effect, except when it is overridden by a routing rule that contains a more specific message matching condition.

Procedure

1. Create a routing plan definition. To do this, complete the following steps:
   a. From the IMS Connect Extensions primary menu, select option 1 Definitions.
   b. Select option 16 Routing Plans.
   c. Type NEW and press Enter.
   d. Type a name and press Enter. The Routing Plan panel is displayed.
   e. Enter a description for the routing plan.
   f. Press the Exit function key (F3) to save and exit the panel.

2. Assign the required OTMA routing rules to the plan. To do this, complete the following steps for each rule you wish to assign:
   a. From the IMS Connect Extensions primary menu, select option 1 Definitions.
   b. Select option 9 OTMA Routing Rules.
   c. Select an OTMA routing rule using line action S.
   d. Use the the Prompt function key (F4) to select the routing plan name in the Routing Plan field.

Figure 127. Creating a routing plan in the IMS Connect Extensions ISPF dialog
Press the Exit function key (F3) to save and exit the panel.

What to do next

Activate the routing plan and its associated rules in IMS Connect. For more information, see “Activating OTMA routing rules in IMS Connect.”

Activating OTMA routing rules in IMS Connect

OTMA routing rules defined in the IMS Connect Extensions repository only take effect in IMS Connect once you have activated OTMA rules-based routing.

Before you begin

Before you can activate OTMA routing rules in IMS Connect, you must first create some rules. See “Routing OTMA workload to one or more IMS data stores” on page 248. If you wish to use a routing plan to organize your routing rules, see “Assigning OTMA routing rules to a plan” on page 257.

Procedure

1. If you have not already done so, activate OTMA routing rules in the IMS Connect system definition. To do this, complete the following steps:
   a. From the IMS Connect Extensions primary menu, select option 1 Definitions.
   b. Select option 1 System Definitions.
   c. To edit the IMS Connect system definition you are interested in, enter line action $ next to the name of the system.
   d. Enter a slash (/) next to the Activate OTMA rules option.

   Note: If you have not already done so, you must also select the Activate Advanced Features and Activate IMS Connect Extensions options.
Press the Exit function key (F3) to save and exit the panel.

2. To apply the changes you made in step 1 on page 259 refresh the IMS Connect system. To do this, complete the followings steps:

a. From the IMS Connect Extensions primary option menu, select option 2 Operations.

b. Set View to option 2 Systems.

c. Enter line action CX next to the desired system. The Commands panel is displayed.

d. Select option 2 Refresh. The Refresh Commands panel is displayed.

e. Select option 1 System Definition and then press Enter.

f. Select the command displayed on the Command Preprocessor panel by entering line action / next to the command.

g. Press Enter to submit the request. The result for each command is displayed in the Response column. A response of 0000 indicates that the command was processed successfully.

3. To ensure your IMS Connect system has the latest IMS data store definitions from your repository, perform a refresh for each IMS data store associated with the IMS Connect To do this, complete the followings steps:

a. Press the Exit function key (F3) until you to return to the Commands panel.
b. Use the Prompt function key (F4) in the field to the right of the **Datastore** field to select an IMS data store defined in the IMS Connect Extensions repository.

c. Select option **2 Datastore** and then press Enter.

d. Select the command displayed on the Command Preprocessor panel by entering line action / next to the command.

e. Press Enter to submit the request. The result for each command is displayed in the **Response** column. A response of 0000 indicates that the command was processed successfully.

f. Repeat this process for each IMS data store as required.

4. Optional: If you have assigned one or more routing rules to a routing plan, activate the routing plan. To do this, complete the followings steps:

   a. Press the Exit function key (F3) until you to return to the Commands panel.

   b. Select option **4 Set**.

   c. Select option **7 Routing Plan**.

   d. Use the the Prompt function key (F4) in the field to the right of the **Activate OTMA Routing Plan** field to select a plan from a list of routing plans defined in the IMS Connect Extensions repository.

   e. Enter / to the left of the **Activate OTMA Routing Plan** field.

f. Press Enter. The Command Processor panel is displayed.

g. Select the command displayed on the Command Preprocessor panel by entering line action / next to the command.

h. Press Enter to submit the request. The result for each command is displayed in the **Response** column. A response of 0000 indicates that the command was processed successfully.

5. Refresh your OTMA routing rules. To do this, complete the followings steps:
a. Press the Exit function key (F3) until you return to the Commands panel.
b. Select option 2 Refresh.
c. Select option 7 OTMA Routing Rules and then press Enter.
d. Select the command displayed on the Command Preprocessor panel by entering line action / next to the command.

e. Press Enter to submit the request. The result for each command is displayed in the Response column. A response of 0000 indicates that the command was processed successfully.

Note: Routing rules are also refreshed when IMS Connect is restarted.

6. Browse the message log for an IMS Connect system to review the rules that are now in-effect. See “Reviewing the active OTMA routing rules in IMS Connect.”

What to do next

To switch to a new routing plan, simply repeat steps 4 on page 261 and 5 on page 261. You only need to refresh the IMS Connect system and IMS data store definitions when you first create the definitions or when the IMS Connect Extensions repository definition has been modified.

Related tasks:

“Activating and deactivating OTMA routing plans” on page 198

Use the Update > OTMA Routing Plan context option in the Status Monitor of the Operations Console to activate or deactivate OTMA routing plan on several IMS Connect systems at once.

Related reference:

“SET command” on page 453

The SET host command for REXX activates or deactivates an OTMA or ODBM routing plan.

Reviewing the active OTMA routing rules in IMS Connect

When a system starts that has OTMA routing rules activated, a series of messages is written to CEXPRINT. These messages display the OTMA routing plan that is currently in effect (if any) and provides summarized information about the OTMA routing rules, OTMA routing lists, and IMS data stores that may be in use.

Note: More OTMA routing plans, OTMA routing rules, OTMA routing lists, and IMS data stores may be defined in the IMS Connect Extensions repository than are in-effect on the system. The OTMA routing rules descriptions that are written to CEXPRINT show the in-memory rules and objects that are currently in effect.
**Tip:** To see the effects of OTMA rules-based routing, use the following methods:

- Monitor data store accepted counts using the status monitor. See "Status Monitor" on page 187.
- Use the IMS Connect Extensions feed with an analytics platform to monitor the distribution of workload across IMS data stores. See Chapter 7, “Forwarding a live feed of IMS Connect events,” on page 99.
- Use IBM IMS Performance Analyzer for z/OS to generate an IMS Connect form-based transaction transit summary report from the IMS Connect Extensions journal that contains the ORIGDS (Original IMS data store) and TARGDS (Target IMS data store) form fields. See “Reporting on OTMA workload routing in IMS Connect using IMS Performance Analyzer” on page 90.

OTMA routing rules and their dependent definitions are listed using message CEX5091I. The first message that is listed is displayed below:

```
07.07.47 CEX5091I ---Begin OTMA routing rules descriptions---, refresh index=1---
```

If OTMA rules-based routing has been activated in the IMS Connect system definition, the following message is displayed:

```
04.40.16 CEX5091I OTMA Routing is active
```

If there is an active routing plan, it is listed next. In this example, OFFPEAK is the active OTMA routing plan on this system.

```
07.07.47 CEX5091I OTMA Routing plan is OFFPEAK
```

The OTMA routing rules that are in effect are listed next. In this example, the OTMA routing rule that is triggered by an IRM_IMSDestID of DEPS is shown. The target OTMA routing list for each message type is RLPLA\textsubscript{W}. The name of the OTMA routing rule is PLNRULEW.

```
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
```

The next rule in effect is for workload with an IRM_IMSDestID of PROD. The rule name is OTMARULE. For each message type, workload can be distributed to IMS data stores listed in OTMA routing list RLIST1. If the data stores in RLIST1 are unavailable, the fallback data stores in RLIST2 will be used.

```
07.07.47 CEX5091I OTMA routing rule for Destid=PROD
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
```

Finally, the OTMA routing lists are displayed in message CEX5093I. The message shows the IMS data stores listed in each OTMA routing list and the data store's capacity weight rating (CWR).
Implied OTMA routing rules

In the following example, a rule TRNRULE has been defined that matches on an IRM.IMSDestID of PRODTRN but is also qualified by transactions listed in MYTRANS. In this situation, IMS Connect Extensions generates an additional implicit rule for the PRODTRN destination ID that is not qualified by transaction. For more information on implicit rules, see "Defining OTMA routing rules" on page 356.

Note: You can see the contents of an IMS request message (IRM) using IBM IMS Problem Investigator for z/OS and the IMS Connect Extensions journal. For more information on IRM, see Format of user portion of IRM for HWSSMPL0, HWSSMPL1, and user-written message exit routines in the IMS user documentation. See also “Reporting and analysis with IMS Problem Investigator” on page 73.

Related concepts:

“Browsing the message log for an IMS Connect” on page 164

The message log for an IMS Connect system reveals activity within IMS Connect and IMS Connect Extensions. Use the message log to understand more about the status of archiving, to view error messages, or to display a history of commands that have been issued and their responses.
Controlling OTMA routing behavior via CEXCTLIN control options

You can control several aspects of default OTMA routing behavior via CEXCTLIN control options.

The following topics relate to the use of the CEXROUTE option to adjust OTMA workload routing behavior for IMS Connect Extensions.

Tip: Member CEXCTL01 in the SCEXSAMP sample library contains the complete range of sample control input data set options. For more information on CEXCTLIN control options, see Chapter 26, “Control input data set options,” on page 561.

Rejecting transactions for data stores with no OTMA routing rule

You can enforce the use of OTMA routing-rules in IMS Connect Extensions by creating a special routing rule that contains no target IMS data stores and by using a combination of CEXCTLIN control options that work with this rule to reject workloads that cannot be matched to your existing set of active routing rules.

Before you begin

If you are using CEXCTLIN control options to adjust OTMA routing behavior, you should also set up one or more OTMA routing rules. See “Routing OTMA workload to one or more IMS data stores” on page 248.

About this task

To ensure that all of your OTMA workload will pass through only those OTMA routing-rules you have defined in IMS Connect Extensions, you must create a special routing rule that contains no target IMS data stores. IMS Connect Extensions can then be configured to use this rule when IMS Connect receives an IMS destination ID for which it has no routing rule. After completing this procedure, client requests that contain destination IDs that have no matching OTMA routing rule will be rejected.

Procedure

1. Create an OTMA routing rule that will be used exclusively to reject requests and will contains no target or fallback IMS data stores. To do this, complete the following steps:
   a. From the IMS Connect Extensions primary menu, select option 1 Definitions.
b. Select option 9 **OTMA Routing Rules**.

c. Type **NEW** and press Enter.

d. Type a name (for example, “NOTARGET”) and press Enter. The OTMA Routing Rule panel is displayed.

e. Specify the IMS Connect system or systems to which this rule applies in the **Apply rule to** section. Enter **1 Systems**, **2 Group**, or **3 All systems**. Press the Prompt function key (F4) in either the **Systems** or **Group** fields to select from a list of previously defined systems or groups.

f. In the **Original Datastore** field, enter a name that does not match the name of a IMS destination ID specified in any IRM you may receive. In this example, we have specified a datastore name of “DUMMY” as our destination ID as it is unlikely to be used by a client application, but you can use any name you wish. Do not specify any target or fallback routing lists; leave these fields blank.

g. When you have finished making changes, press the Exit function key (F3). The new routing rule is displayed in the OTMA Routing Rules panel.

2. Modify the CEXCTLIN DD statement in the IMS Connect startup job to include the **CEXROUTE** control option. Specify the following parameters:

   **Note:** For more information on either of these options, see “**CEXROUTE option**” on page 562.

   **RBR_NODEST=** *destid* where *destid* is the Original Datastore name you specified in step **1 on page 265**

   The **RBR_NODEST** parameter allows you to dynamically change the IRM_IMSDestId of the incoming IRM if the IRM_IMSDestId cannot be matched with any OTMA routing rule during the initial pass. Here, we wish to redirect this workload to the new routing rule we specified in step **1 on page 265** To do this, use **RBR_NODEST=** *destid* where *destid* is the Original Datastore name you specified in the routing rule. When
IMS Connect Extensions performs a second pass over the routing rules to find a match, it finds the rule you created in step 1 on page 265.

In the example above, we used the name DUMMY as our Original Datastore name. To instruct IMS Connect Extensions to change the IRM_IMSDestID to DUMMY so it will match on this rule, use RBR_NODEST=DUMMY.

**Remember:** You can use any unused Original Datastore name you like; you don't have to use the word DUMMY.

RBR_FAILURE=REJECT

The RBR_FAILURE keyword specifies how an input message is to be handled when the request cannot be routed to any of the destinations specified in the routing rule. When RBR_FAILURE=REJECT is specified, the message will be rejected with an RSM indicating that no destination is available. In our case, the rule we specified in step 1 on page 265 has no destinations at all; with RBR_FAILURE=REJECT specified any workload that is routed to this rule will now be automatically rejected.

3. To apply these changes and to start rejecting message that do not match a rule, restart IMS Connect.

**Related concepts:**

“Suspending OTMA workload routing to IMS data stores using drain” on page 273

The drain facility in IMS Connect Extensions temporarily suspends OTMA rules-based routing of messages to a data store (or a set of data stores), 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.

**OTMA flood control processing in IMS Connect Extensions**

IMS Connect Extensions OTMA rules-based routing can be used to avoid message flood conditions in IMS.

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 described by 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.
Tip: Use the IMS Connect Exception Events report in IMS Performance Analyzer to report on OTMA flood control notifications. For more information, see OTMA flood control notification in the IBM IMS Performance Analyzer for z/OS Report Reference.

Note: 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 a data store 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 data store that is in Member Flood Warning condition.

For more information about message flood detection in IMS, see Message flood detection in the IMS user documentation.

Enabling OTMA message flood detection

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 specified in the DFSYDTx member of the IMS PROCLIB data set. See OTMA client descriptor syntax and parameters in the IMS user documentation.
- The INPUT parameter of the /START TMEM IMS type-1 command. See “Issuing IMS type-1 commands” on page 215 and the /START TMEM command topic in the IMS user documentation.
- 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 IMS type-1 command:

/START TMEMBER ALL INPUT number

Values specified in the OTMA client descriptor can be overridden by the /START TMEM IMS type-1 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.

Using CEXCTLIN options to protect target IMS systems

Use the CEXROUTE INELIGIBLE=GLOBALFLOODWARNING option to prevent OTMA rules-based routing to IMS data stores that are experiencing a Global Flood Warning. For more information on the CEXROUTE INELIGIBLE=GLOBALFLOODWARNING option, see “CEXROUTE option” on page 562. With this option set, IMS Connect Extensions OTMA rules-based routing will perform the following actions when processing OTMA routing rules to avoid a message flood condition:

1. The target OTMA routing list in the OTMA routing rule is checked for an available IMS data store. In this first pass, IMS data stores in the degraded state are excluded from the list of routing candidates.
2. If no available IMS data store is found in the target OTMA routing list, an attempt is made to find an available data store in the fallback OTMA routing list.
3. If no fallback OTMA routing list is present, or no available candidates can be found in the list, the target OTMA routing list is searched again for available data stores, this time including those in the degraded state. If the `CEXROUTE INELIGIBLE=GLOBALFLOODWARNING` option is set in the CEXCTLIN data set, any IMS data store that is in the degraded state is treated as being in the unavailable state and therefore excluded from the list of routing candidates.

4. If no candidates can once again be found in the target OTMA routing list, the fallback OTMA routing list, if present, is searched again for an available data store, this time including those in degraded state. As before, if the `CEXROUTE INELIGIBLE=GLOBALFLOODWARNING` option is set in the CEXCTLIN data set, any IMS data store that is in the degraded state is treated as being in the unavailable state and therefore excluded from the list of routing candidates.

When the `CEXROUTE INELIGIBLE=GLOBALFLOODWARNING` is specified, IMS Connect Extensions will treat a data store with a Global Flood Warning condition as if it had a Global Flood Severe condition and stop sending messages to the data store. If the Global Flood Warning condition results in no available data stores, IMS Connect Extensions routing will reject the message with an RSM indicating that there are no available data stores. This is done because if the message is returned to IMS Connect, it will be sent to the target data store and might compromise the availability of the IMS system.

IMS Connect Extensions will send messages to a data store in Flood Warning status if that is the only option available. If the data store changes to the Member Flood Severe condition, IMS Connect Extensions will not send messages to the data store. 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.

**Related concepts:**

- "Datastore Monitor" on page 141
  Use the Datastore Monitor in the IMS Connect Extensions ISPF dialog to manage and monitor your IMS data stores. Use the Datastore Monitor to view detailed information about an IMS data store, to suspend and resume routing of messages to an IMS data store using the drain feature, and to start and stop an IMS data store.

**Related tasks:**

- "Managing IMS data stores" on page 204
  Use the Datatstores and Datastore Groups tab in Status Monitor to perform management tasks on IMS data stores from the IMS Connect Extensions Operations Console. Use this facility to start and stop IMS data stores, suspend and resume routing to a data store, and to dynamically update a data stores capacity weight rating (CWR) in response to periods of changing demand.

**Pre-routing user exit**

You can use the pre-routing exit to influence the selection of IMS data stores as candidates for OTMA rules-based routing.

IMS Connect can determine whether an IMS data store 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 data stores” case

Consider the case where all IMS data stores 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 IRM_IMSDestID that is sent to IMS Connect is not a valid data store. That way if no data stores are selected and the original datastore is used as the final data store target, it will fail due to missing data store.
• Use the CEXROUTE RBR_FAILURE=REJECT option in the CEXCTLIN data set to request that the message be rejected if there are no data stores selected by routing. For more information, see “CEXROUTE option” on page 562.

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:

<table>
<thead>
<tr>
<th>RC</th>
<th>Function</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>INIT</td>
<td>The exit has initialized and wishes to receive pre-routing requests.</td>
</tr>
<tr>
<td>0</td>
<td>TERM</td>
<td>The exit has completed termination and released all resources.</td>
</tr>
<tr>
<td>0</td>
<td>ROUT</td>
<td>All of the tmember names provided to the exit are acceptable.</td>
</tr>
<tr>
<td>RC</td>
<td>Function</td>
<td>Meaning</td>
</tr>
<tr>
<td>----</td>
<td>----------</td>
<td>---------</td>
</tr>
</tbody>
</table>
| 4  | ROUT     | All of the tmember names are acceptable; do not call the exit again for this specific transaction.
|    |          | 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. |
|    |          | **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. Return and reason codes will be reported in a CEX5620E message. |
| 16 | TERM     | The exit encountered issues during termination. 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. 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.

**Related information:**

"CEX5620E” on page 524
The pre-routing exit `exit_status`
Chapter 13. OTMA operations management and tuning

After you have configured OTMA rules-based routing in IMS Connect Extensions, review the following topics for additional ways in which you can manage and tune your IMS Connect systems.

Suspending OTMA workload routing to IMS data stores using drain

The drain facility in IMS Connect Extensions temporarily suspends OTMA rules-based routing of messages to a data store (or a set of data stores), 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 /CHECKPOINT (or /CHE) IMS type-1 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, you might issue a command to stop the IMS data store first (for example, using the IMS Connect STOPDS WTO command), but that can result in transaction responses being trapped within the IMS system as they would not have the data store 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 data store pipe before issuing the /CHE command.

The drain facility in IMS Connect Extensions allows the routing status of one or more IMS data stores to be changed so that the data store may be “drained” of in-progress transactions prior to IMS shutdown. The drain process works for any form of OTMA routing in IMS Connect Extensions.

**Important:** The drain facility only works on messages handled by IMS Connect Extensions OTMA rules-based routing. It does not work on messages that circumvent a routing rule or with exits customized with their own routing mechanisms. To ensure that all of your workload is handled by IMS Connect Extensions, see “Rejecting transactions for data stores with no OTMA routing rule” on page 265.

You can drain IMS data stores in the following ways:

- Using the Datastore Monitor panel in the IMS Connect Extensions ISPF dialog. For more information, see “Datastore Monitor” on page 141.
- Using the Datastores tab in the IMS Connect Extensions Operations Console for z/OS Explorer. With the Operations Console you can sort IMS data stores across multiple IMS Connect systems by XCF group to provide a IMS view of your data stores. By selecting all the data stores for an IMS system you can effectively drain the IMS system. For more information on the Operations Console, see “Managing IMS data stores” on page 204.
Using the IMS Connect Extensions host command environment for REXX and the `ROUTE ACTION=DRAIN` and `QUERY TYPE=PENDING RESPONSES` commands. Sample members CEXRXC21, CEXRXC22, and CEXRXC23 in the SCEXSAMP library demonstrate the correct usage of these commands. For more information, see "REXX automation samples for the IMS Connect Extensions host command environment" on page 223.

Related concepts:

"Datastore Monitor" on page 141

Use the Datastore Monitor in the IMS Connect Extensions ISPF dialog to manage and monitor your IMS data stores. Use the Datastore Monitor to view detailed information about an IMS data store, to suspend and resume routing of messages to an IMS data store using the drain feature, and to start and stop an IMS data store.

Related tasks:

"Managing IMS data stores" on page 204

Use the Datastores and Datastore Groups tab in Status Monitor to perform management tasks on IMS data stores from the IMS Connect Extensions Operations Console. Use this facility to start and stop IMS data stores, suspend and resume routing to a data store, and to dynamically update a data stores capacity weight rating (CWR) in response to periods of changing demand.

**Dynamically adding IMS data stores as targets for OTMA routing**

IMS Connect Extensions supports rules-based routing of OTMA messages to dynamically added IMS data stores without restarting IMS Connect.

**Before you begin**

If you plan to add an IMS data store dynamically to IMS Connect, it is preferable to add the data store 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 data store might be defined to IMS Connect but not available for OTMA routing in IMS Connect Extensions:

1. The data store was specified in the IMS Connect configuration member. IMS Connect and IMS Connect Extensions were restarted, but the data store has not been defined in the IMS Connect Extensions repository.

   In this scenario, add the IMS data store definition to the IMS Connect Extensions repository and then use the `ADD` 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` host command. See "REXX automation samples for the IMS Connect Extensions host command environment" on page 223.

2. The data store was dynamically added to IMS Connect. IMS V13 introduced an IMS type-2 command that allows you to add IMS Connect IMS data store definitions during runtime processing without having to restart your IMS Connect instance. For more information, visit the `CREATE IMSCON TYPE(DATASTORE) command` topic in the IMS user documentation.

   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 data store definition in IMS Connect Extensions before creating the
data store itself using the *CREATE IMSCON* command. This allows you to refresh the in-memory routing rules to use the new data store for OTMA rules-based routing.

The amount of room reserved for dynamically adding entries to the table is controlled by the AUTOADD_DATASTORE option on the CEXCTLIN DD statement. Reserve enough space to add the maximum number of data stores needed for the life of the IMS Connect region. For more information, see “AUTOADD_DATASTORE option” on page 561.

**Procedure**

1. Add the IMS data store definition in the IMS Connect Extensions repository. See “Defining IMS data stores” on page 339.
2. Issue the *CREATE IMSCON* IMS type-2 command to add an IMS data store definition to IMS Connect. This command adds the data store definition during runtime processing and does not require an IMS Connect restart.
3. Add the data store definition to one or more OTMA routing lists in IMS Connect Extensions. See “Defining OTMA routing lists” on page 361.
4. Refresh the in-memory OTMA routing rule definitions in IMS Connect. See “Refreshing in-memory definitions from the IMS Connect Extensions repository” on page 150.

**What to do next**

Messages can now be routed to the data store. The status of the data store can now be seen in the ISPF Operations dialog and Operations Console for z/OS Explorer.

**Related concepts:**

“Datostore Monitor” on page 141

Use the Datostore Monitor in the IMS Connect Extensions ISPF dialog to manage and monitor your IMS data stores. Use the Datostore Monitor to view detailed information about an IMS data store, to suspend and resume routing of messages to an IMS data store using the drain feature, and to start and stop an IMS data store.

**Related tasks:**

“Managing IMS data stores” on page 204

Use the Datostores and Datostore Groups tab in Status Monitor to perform management tasks on IMS data stores from the IMS Connect Extensions Operations Console. Use this facility to start and stop IMS data stores, suspend and resume routing to a data store, and to dynamically update a data stores capacity weight rating (CWR) in response to periods of changing demand.

**Related reference:**

“ADD command” on page 429

The ADD host command for REXX adds an entry to the in-memory IMS data store table for a data store that is defined to IMS Connect but was not defined to IMS Connect Extensions at startup.

“REFRESH command” on page 449

The REFRESH host command for REXX rebuilds the in-memory copy of the selected definitions in the target IMS Connect system from the definitions stored in your IMS Connect Extensions repository.
Rebalancing sessions across IMS Connect systems

The Session Message Limit option in the IMS Connect system definition allows you to specify the maximum number of input messages for a persistent session before the session is automatically closed. In an IMS sysplex distributor environment, if you design your applications to automatically reconnect when the session is closed, new persistent sessions will be routed to the IMS Connect with the lowest session totals. This feature is particularly useful when an IMS Connect system is being brought back online to quickly rebalance your persistent sessions across IMS Connect systems.

About this task

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 fail, 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 IMS Connect system definition allow you to specify the maximum number of input messages for a persistent session. When the 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.

Procedure

To implement automatic session rebalancing for an IMS Connect system, perform the following steps:

1. Code your remote clients to detect when a socket session is being closed by IMS and automatically request a new socket.

2. Configure the IMS Connect system definition to set a session message limit. To do this, complete the following steps:
   a. From the IMS Connect Extensions primary menu, select option 1.1 System Definitions
   b. Enter a slash (/) next to the Activate Session Message Limit option.
   c. Specify the maximum number of input messages in the Limit threshold field.
      Tip: The Message Count field on the Active Sessions panel displays the number of input messages received so far by the session. See “Customizing the Active Sessions panel” on page 138.
      For more information, see “Defining IMS Connect systems” on page 320.

3. Refresh the in-memory copy of the IMS Connect system definitions with the ones in the IMS Connect Extensions repository. See “Refreshing in-memory definitions from the IMS Connect Extensions repository” on page 150.

What to do next

To rebalance all sessions according to a schedule, create a job that uses the DRAIN host command for REXX that specifies TOKEN=* and the member CEXRXC26 in the
SCEXSAMP library shows how to drain all sessions using a REXX exec.

Related reference:

“DRAIN command” on page 434

The DRAIN host command for REXX causes eligible IMS Connect sessions to be marked for closure at the next inbound message. Sessions are closed normally and not canceled.

“Defining IMS Connect systems” on page 320

Use the System Definitions panel to create a new definition for an IMS Connect and to specify 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. To access this ISPF panel, select option 1.1 System Definitions from the IMS Connect Extensions primary menu.

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 for 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. From the primary menu in the IMS Connect Extensions ISPF dialog, select option 1.7 Transactions.
2. Enter S next to the transaction code for which you want to enable additional transaction processing options.
3. 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.

4. Press the Exit function key (F3) to save the definition.

5. Use the **Refresh** command to reload the transaction definition, or restart IMS Connect.

**Related reference:**

“Defining transactions” on page 349

The Transactions panel allows you to define transactions to be routed by IMS Connect Extensions through the legacy transaction routing feature. To access this ISPF panel, select option 1.7 Transactions from the IMS Connect Extensions primary menu.
Chapter 14. ODBM workload routing in IMS Connect

ODBM rules-based routing in IMS Connect Extensions provides advanced routing in IMS Connect for TCP/IP client requests that use Distributed Relational Database Architecture™ (DRDA) to access IMS data via IMS Open Database.

ODBM rules-based routing provides an additional level of abstraction on top of standard ODBM routing. It allows you to redirect DRDA requests received by the client to alternate IMS data stores using the PSB name and alias values supplied in the request as well as route workload to fallback data stores when IMS systems are shut down. Use ODBM rules-based routing if you need to direct more workload to one data store over another, or if you wish to ensure open database requests for specific databases are sent to the correct IMS. This abstraction gives you great flexibility in managing client workload without having to change either the client, your ODBM configuration, or your IMS systems.

Tip: For general information about setting up an ODBM instance in your environment, see Enabling open access to IMS data in the IMS user documentation.

Principles of ODBM rules-based routing

A DRDA request message identifies the target database for a given interaction by supplying the alias name for an IMS data store. When IMS Connect, receives the request, it uses this IMS data store alias to determine which ODBM instance will handle the request. IMS Connect Extensions extends this basic message passing capability with ODBM routing rules that can intercept this request and redirect it to alternative IMS data stores as desired.

ODBM rules-based routing works in the following ways:

The alias in the DRDA request message is used to find a matching ODBM routing rule

Rules are matched in IMS Connect Extensions based on the alias supplied in the DRDA request message in the distributed data management (DDM) architecture ACCRDB command (X'2001') in the RDBNAM parameter. For more information, refer to the topic ACCRDB command (X'2001') in the IMS user documentation. Because the field simply contains a string, the alias supplied in the request does not have to be a IMS data store alias defined to ODBM. Instead, it can be a generic value that might represent some workload grouping such as an application, business group, or cost center. IMS Connect Extensions locates the ODBM routing rule whose key field (alias) matches that generic string (RDBNAM parameter) specified in the message. IMS Connect Extensions processes the routing rule and replaces the generic alias supplied in the request with a real IMS data store alias selected from the list of routing targets referenced by the ODBM routing rule. In IMS Connect Extensions, these targets are represented by ODBM target definitions in the IMS Connect Extensions repository.

ODBM target definitions are used to specify a routing pathway between an ODBM instance and an IMS data store

An IMS Connect Extensions ODBM target definition specifies a routing pathway between an ODBM instance and an IMS data store (identified by its alias). In ODBM, IMS data store aliases are defined in the CSLDCxxx member of the IMS PROCLIB data set.
ODBM routing lists are used to define the list of ODBM targets

An ODBM routing list contains a list of ODBM targets that may be used for routing. For each ODBM routing rule, you can specify a target ODBM routing list and an optional fallback ODBM routing list. Messages are routed (via ODBM) to IMS data stores in the target ODBM routing list first. The fallback list contains data stores that can be used if the data stores in the target list are unavailable.

ODBM routing rules can be further qualified by adding conditions for one or more PSB names

To do this, you can define a “master” ODBM routing rule for each incoming alias name, and then specify additional qualifying rules that route requests with particular PSB names to a different list of ODBM targets. If you specify a qualifying rule without a matching master rule in the repository, an implied routing rule will be generated internally at run time.

IMS Connect Extensions checks for the name of the active ODBM routing plan

Only one ODBM routing plan can be active on a system at a time. Routing rules that are assigned to other routing plans are not considered.

A routing rule that is not assigned to a routing plan will always be in effect, except when it is overridden by a routing rule that contains a more specific message matching condition.

Define sets of rules for all systems, or develop customized rules for each

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.

You can also have one master rule for a given alias, as well as optional rules that specify different ODBM routing lists for select PSB names.

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.

Use CEXCTLIN control options to set the default routing behavior for an IMS Connect

The control input data set allows you to set several behaviors for ODBM routing. The input data set is specified through an optional CEXCTLIN DD statement in the IMS Connect startup job. These options take effect when IMS Connect Extensions restarts. Use these options to control the behavior of IMS Connect Extensions when the DRDA request does not match any of its current routing rules.
Developing ODBM routing rules for IMS Connect

To route ODBM workloads in IMS Connect, you’ll need to create one or more ODBM routing rules that depend on several supporting definitions in the IMS Connect Extensions repository. The definitions you build will depend on the routing scenario you would like to create.

Important: To route DRDA request traffic, you must first configure the ODBMROUT exit member in the BPE configuration member to use the IMS Connect Extensions routing exit, CEXROUT0. For more information, see “Configuring the ODBMROUT exit member” on page 31.

ODBM routing rules in IMS Connect Extensions are flexible in that they allow you to configure your systems in a variety of ways. Whilst building a set of ODBM routing rules will ultimately depend on an understanding of the nature of your workload, you will essentially use the following procedures to configure your rules:

1. **Create an ODBM routing rule to route workload to one or more IMS data stores via ODBM.**

   Use this procedure first to create your initial routing rule and its supporting definitions for a single IMS Connect, a group of IMS Connect systems, or all systems. This procedure will help you achieve the following routing outcomes:
   * Route ODBM workloads dynamically to an IMS data store selected from a pool of IMS data stores. Workloads are delivered via ODBM and are based on the alias name of an IMS data store supplied in the request message.
   * Redirect ODBM workload intended for a specific IMS data store to a different data store.

   To perform this routing, the procedure will explain how to create ODBM target definitions in the IMS Connect Extensions repository. ODBM target definitions specify a routing pathway between an ODBM instance and an IMS data store.

   To create your first ODBM routing rule, see “Routing ODBM workload to one or more IMS data stores” on page 282.

2. **Optional: Configure your routing rule with fallback IMS data stores.**

   After you have created your initial routing rule, use this optional procedure to configure the rule to route workload to a list of fallback IMS data stores if primary target data stores in the rule are unavailable. This can be useful as a backup for your primary data stores, or when you need to perform maintenance of the primary IMS data stores.

   To configure a rule with fallback IMS data stores, see “Routing ODBM workload to a list of fallback IMS data stores” on page 287.

3. **Optional: Route workload according to PSB name.**

   After you have created your initial routing rule, use this optional procedure to route specific ODBM workloads to alternate IMS data stores based on the PSB name received in the DRDA request. This may be used to ensure requests only run on IMS data stores where the database exists.

   To qualify an ODBM routing rule by PSB name, see “Routing ODBM workload according to PSB name” on page 288.

4. **Optional: Set the relative capacities of each of your ODBM targets.**

   After you have created your initial routing rule, use this optional procedure to balance workload across multiple IMS data stores based on the processing capacity defined in an ODBM target. This procedure will help you set the initial capacity weight rating (CWR) for an ODBM target which allows you to
reduce (or increase) the processing capacity of an IMS data store accordingly, or set a custom configuration that balances workload accordingly. Once an initial value has been set, you can dynamically adjust the capacity weight rating using the Operations Console for IBM Explorer for z/OS.

To set capacity weight ratings, see “Setting the capacity of an ODBM target” on page 290.

5. **Recommended:** Group your routing rules by assigning them to a routing plan.

After you have created your initial routing rule, use this recommended procedure to establish multiple rule sets bound together by a routing plan. Use a routing plan to create sets of rules that work together. The routing plan can then be activated in the following ways:

- Manually by an operator via the IMS Connect Extensions ISPF dialog or Operations Console for IBM Explorer for z/OS.
- Automatically using the IMS Connect Extensions host command environment for REXX and the automation tool of your choice.

To create routing plans, see “Assigning ODBM routing rules to a plan” on page 293.

Routing rules will not take effect until they are activated. To activate your routing rules, see “Activating ODBM routing rules in IMS Connect” on page 294.

**Important:** You can also control the default routing behavior of IMS Connect Extensions when a request is received but there is no matching ODBM routing rule. For more information, see “Controlling ODBM routing behavior via CEXCTLIN control options” on page 299.

**Related tasks:**
- “Configuring the ODBMROUTE exit member” on page 31

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

### Routing ODBM workload to one or more IMS data stores

You can create ODBM routing rules to perform distributed routing of DRDA messages containing a specific IMS data store alias to lists of IMS data store aliases via ODBM. This allows you to distribute workload across multiple IMS data stores or to redirect workload to an alternate data store without changing your client application.

**Before you begin**

Create IMS Connect Extensions repository definitions for the IMS Connect systems you wish to use in your routing rule. For more information, see “Defining IMS Connect systems” on page 320. If you wish to apply your rule to several IMS Connect systems at once, create an IMS system group. For more information, see “Defining system groups” on page 355.

**About this task**

A simple routing rule that directs ODBM workload to a list of IMS data stores for processing is a common first step to implementing ODBM rules-based routing with IMS Connect Extensions.

In this scenario, the ODBM routing rule requires the following key ingredients:
- The name of the IMS Connect system (or systems) that require the routing rule.
- The conditions required to invoke the rule; in this case the alias name of an IMS data store specified in the client request. In ODBM, application programs use alias names to access IMS data stores.
- One or more ODBM target definitions to use as routing candidates. An IMS Connect Extensions ODBM target definition specify a routing pathway between an ODBM instance and an IMS data store. To define an ODBM target, you will need the following information:
  - The ODBMID of the ODBM instance you will use to route workload.
  - The alias names of one or more IMS data stores associated with that ODBM instance.
- An ODBM routing list that contains the ODBM target definitions.

Tip: To see a list of ODBM names and IMS aliases for an active IMS Connect system, use the Status Monitor. See “Systems Overview” on page 126.

The following diagram shows how these elements work together to route ODBM workloads across IMS. In this example, a request is received by IMS Connect that is intercepted by ODBM routing rule RULE1. RULE1 has been configured to capture workloads directed at alias ANY1 and redirect them to targets contained within ODBM routing list GROUP1. In this scenario, GROUP1 contains three ODBM targets (ODT1, ODT2, and ODT3) which are linked to three IMS data store aliases (IMSA, IMSB, and IMSC respectively). When a message is received by IMS Connect, IMS Connect Extensions will route it to one of the ODBM targets.

![Figure 135. IMS Connect and IMS Connect Extensions performing dynamic routing of ODBM workloads to multiple IMS data stores via ODBM targets](image)

Because incoming messages are keyed to ODBM routing rules by the alias name, it doesn't actually matter if the incoming alias is a real IMS data store alias or not; the ODBM routing list and targets within allow you to direct the traffic in any way you like. This approach offers a great deal of flexibility and enables client applications to send “logical” aliases rather than physical ones as needs require. Furthermore, this arrangement allows you to easily take ODBMs and IMS data stores offline without disrupting ongoing operations. For more information, see “Suspending OTMA workload routing to IMS data stores using drain” on page 273.
The following procedure shows you how to create and set up the appropriate routing rule, ODBM targets, and ODBM target list as described in the diagram above.

**Procedure**

1. Create ODBM target definitions that identify a pathway between an ODBM instance and an IMS data store alias. To do this, complete the following steps:
   a. From the IMS Connect Extensions primary menu, select option 1 Definitions.
   b. Select option 11 ODBM Targets.
   c. Type **NEW** and press Enter.
   d. Type a name and press Enter.
   e. On the ODBM Targets panel, enter the following information:

   **ODBM Name**
   In the **ODBM Name** field, enter the ODBMID of the ODBM instance you wish to use.

   **Tip:** Use the Status Monitor to see a list of ODBMs for an active IMS Connect. See “Systems Overview” on page 126.

   The ODBMID is the name defined in the CSLDIxxx member of the IMS PROCLIB data set followed by the characters OD and any blank spaces that ODBM needs to add to make the ODBMID eight characters in length. For more information, see CSLDIxxx member of the IMS PROCLIB data set in the IMS user documentation.

   **IMS Alias**
   In the **IMS Alias** field, enter the 4-character alias name for the IMS data store you wish to target.

   **Tip:** Use the Status Monitor to see a list of aliases in an active IMS Connect. See “Systems Overview” on page 126.

   Application programs must use an alias name to access IMS data stores and do not need to know the actual IMSID of the IMS data store. Alias names for IMS data stores are defined in the CSLDCxxx member of the IMS PROCLIB data set. For more information, see CSLDCxxx member of the IMS PROCLIB data set in the IMS user documentation.

   **Note:** For additional options on this panel, see “Defining ODBM targets” on page 362.
f. When you have finished making changes press the Exit function key (F3).

The new ODBM target is displayed in the ODBM Targets panel.

Repeat this process for each new ODBM target as required.

2. Create an ODBM routing list definition that defines a collection of ODBM targets that may be used when routing workloads. To do this, complete the following steps:
   a. From the IMS Connect Extensions primary menu, select option 1 Definitions.
   b. Select option 13 ODBM Routing Lists.
   c. Type NEW and press Enter.
   d. Type a name and press Enter.
   e. Assign one or more ODBM targets to the list. To do this, press the the Prompt function key (F4) on the field beneath the Target column. To insert new rows for additional data stores, use line action i. For a complete list of options, use line action /. For a complete list of options, see "Defining ODBM routing lists" on page 367.

f. When you have finished making changes press the Exit function key (F3).

The new routing list is displayed in the ODBM Routing List panel.

3. Create an ODBM routing rule definition for an IMS Connect system (or group of systems) that defines the conditions required to route your chosen DRDA messages to the ODBM routing list you defined in step 2.
   a. From the IMS Connect Extensions primary menu, select option 1 Definitions.
b. Select option 12 **ODBM Routing Rules**.

c. Type **NEW** and press Enter.

d. Type a name and press Enter. The ODBM Routing Rule panel is displayed.

e. Specify the IMS Connect system or systems to which this rule applies in the **Apply rule to** section. Enter **1 Systems**, **2 Group**, or **3 All systems**. Press the Prompt function key (F4) in either the **Systems** or **Group** fields to select from a list of previously defined systems or groups.

f. Specify the alias name supplied in the client DRDA request in the **Input Alias** field. For example, to create a rule that applies to messages that contain an alias of ANY1, enter **ANY1**.

**Tip:** The alias name is supplied in the DRDA request message in the distributed data management (DDM) architecture ACCRDB command (X'2001') in the RDBSNAM parameter. For more information, refer to the topic **ACCRDB command (X'2001')** in the IMS user documentation. To see what alias is being used by your client applications, use IBM IMS Problem Investigator for z/OS to format and view IMS Connect Extensions journal record X'5B' DRDA Request. See "Reporting and analysis with IMS Problem Investigator" on page 73.

g. Type **Y** in the field next to the **DRDA Requests** option and then enter the name of the ODBM routing list you defined in step 2 on page 285 in the **Target** column.

h. When you have finished making changes, press the Exit function key (F3). The new routing list is displayed in the ODBM Routing Rules panel.

**Note:** For additional options, see “Defining ODBM routing rules” on page 364.

**What to do next**

Activate the routing rule in IMS Connect. For more information, see “Activating ODBM routing rules in IMS Connect” on page 294.
Routing ODBM workload to a list of fallback IMS data stores

An ODBM routing rule can be configured to route workload to a list of fallback IMS data stores via ODBM if primary target data stores in the rule are unavailable.

Before you begin

Create an ODBM routing rule that routes workload to multiple IMS data stores. See “Routing ODBM workload to one or more IMS data stores” on page 282.

About this task

Consider the following scenario where an ODBM routing rule has been configured to perform “pass-through” routing of all ODBM workload with an incoming alias name of IMSA to the same IMS data store, IMSA. We can provide a backup IMS data store to IMSA by configuring the routing rule to use a fallback ODBM routing list. The fallback ODBM routing list contains one or more ODBM targets that can be used to process this workload if routing to IMSA is suspended, the data store is shutdown, or the data store is otherwise unavailable.

The following procedure shows you how to add a fallback ODBM routing list to an existing ODBM routing rule.

Procedure

1. Create an ODBM routing list definition for one or more fallback IMS data stores. For assistance with this task, see “Routing OTMA workload to one or more IMS data stores” on page 248.
Modify your ODBM routing rule definition to add the fallback ODBM routing list you defined in step 1 on page 287. In the example below, the rule is triggered on IMS Connect group PRODGRP for any request that contains an alias of IMSA.

```
File Menu Settings Help
EDIT ODBM Routing Rule
Command ==>                      Scroll ==> PAGE
Name ... : RULE2
Description ... ODBM Routing Rule for IMSA

Apply rule to:                  Rule is active when:
  1. System ... + Routing Plan ...
  2. Group ... PRODGRP + Rules with no plan are always active
  3. All systems

Condition:
  Input Alias ... IMSA
  PSB name list ...

---- Routing lists ----
/ Request types                  Target + Fallback +
Y DRDA Requests                  GROUP1   GROUP2

************************************************************ Bottom of data ************************************************************
```

Figure 141. Specifying the IMSBLIST ODBM routing list for fallback routing in the ODBM Routing Rule panel in the IMS Connect Extensions ISPF dialog

What to do next

Activate the routing rule in IMS Connect. For more information, see “Activating OTMA routing rules in IMS Connect” on page 259.

Routing ODBM workload according to PSB name

An ODBM routing rule can be configured to route workload according to the PSB name received in the DRDA request.

About this task

Consider the following scenario with two ODBM routing rules that redirect requests to different IMS data stores depending on the PSB name specified in the incoming client request.

- The first rule, RULE1, applies when the incoming DRDA request contains an IMS data store alias name of IMSA. The rule is not qualified by PSB name, meaning that the contents of the PSBNAME field in the request message can be
any value. The rule routes this request to ODBM routing list GROUP1 which contains ODBM target ODT1. ODT1 identifies the ODBM instance used to route the request as well as the IMS data store alias IMSA. This rule acts as a general rule for any message received with an alias of IMSA.

- The second rule, PSBDRULE, applies when the incoming DRDA request contains a PSB name of PSBC and an IMS data store alias of IMSA. In this scenario, we have decide that this message should be rerouted to IMSB. To do this, the rule routes this request to ODBM routing list GROUP2 which contains ODBM target ODT2. ODT2 identifies the ODBM instance used to route the request as well as the IMS data store alias IMSB. This rule overrides RULE1 as it has a more specific condition specified on the PSB name.

The following procedure shows you how to create ODBM routing rules that apply to specific PSB names specified in the client request.

**Procedure**

1. Create a PSB name list definition containing a list of PSB names to apply to your ODBM routing rule. The list should contain only those PSB names that require conditional routing. To do this, complete the following steps:
   a. From the IMS Connect Extensions primary menu, select option 1 Definitions.
   b. Select option 14 PSB Name Lists.
   c. Type NEW and press Enter.
   d. Type a name and press Enter.
   e. Type one or more PSB names in the list. To do this, type the name in in the field beneath the PSB name column and enter an optional description in the Description column. To insert new rows for additional PSB names, use line action 1. For a complete list of options, use line action /

Figure 142. IMS Connect and IMS Connect Extensions routing ODBM workload based on PSB name

The following procedure shows you how to create ODBM routing rules that apply to specific PSB names specified in the client request.

**Procedure**

1. Create a PSB name list definition containing a list of PSB names to apply to your ODBM routing rule. The list should contain only those PSB names that require conditional routing. To do this, complete the following steps:
   a. From the IMS Connect Extensions primary menu, select option 1 Definitions.
   b. Select option 14 PSB Name Lists.
   c. Type NEW and press Enter.
   d. Type a name and press Enter.
   e. Type one or more PSB names in the list. To do this, type the name in in the field beneath the PSB name column and enter an optional description in the Description column. To insert new rows for additional PSB names, use line action 1. For a complete list of options, use line action /. For a more
information on this panel, see “Defining PSB name lists” on page 368.

f. When you have finished making changes press the Exit function key (F3). The new PSB name list is displayed in the PSB Name Lists panel.

2. Create an ODBM routing rule definition that uses the PSB name list you defined in step 1 on page 289. For assistance with this task, see “Routing ODBM workload to one or more IMS data stores” on page 282. In the example below, the rule is triggered on IMS Connect system HWSOPS6 for any request that contains an alias of IMSA and a PSB name found within PSB name list PSBCLIST.

![Figure 143. A PSB name list definition PSBCLIST that contains a PSB name of PSBC](image)

```
Figure 143. A PSB name list definition PSBCLIST that contains a PSB name of PSBC
```

```
Figure 144. Creating an ODBM routing rule qualified on a list of PSB names in the IMS Connect Extensions ISPF dialog
```

**What to do next**

Activate the routing rule in IMS Connect. For more information, see “Activating OTMA routing rules in IMS Connect” on page 259.

**Setting the capacity of an ODBM target**

Workload balancing allows IMS Connect Extensions to redirect incoming transaction requests to ODBM targets according to their capacity weight rating (CWR), thereby balancing the transaction workload across multiple IMS systems and ensuring both availability and responsiveness for the remote client.
Before you begin

Create an ODBM routing rule that routes workload to multiple ODBM targets. See “Routing ODBM workload to one or more IMS data stores” on page 282.

About this task

Workload balancing is accomplished using a weighted rotate algorithm that considers the processing capacity of an ODBM target. In IMS Connect Extensions, this processing capacity is realized by assigning a capacity weight rating (CWR) to each ODBM target. The capacity weight rating for an ODBM target is set in the ODBM target definition stored in the IMS Connect Extensions repository.

An ODBM target’s capacity weight rating is relative to other ODBM targets that may be available for processing the incoming message. For example, if ODBM target ODBMT1 has a capacity weight rating of 10 and ODBM targets ODBMT2 and ODBMT3 each have a capacity weight rating of 20, ODBMT1 will, on average, receive half the workload received by ODBMT2 or by ODBMT3. This ratio of 10:20:20 means that ODBMT1 receives an average of 20 percent of the workload and ODBMT2 and ODBMT3 each receive 40 percent.

Basic rotation in which each ODBM target is deemed to have the same capacity can be achieved by assigning each target the same capacity weight rating (for example, by assigning a value of 1 to each ODBM target in an ODBM routing list), or by deactivating workload balancing in the IMS Connect system definition. You can keep an ODBM target in reserve by assigning it a capacity of zero (0), making it ineligible as a routing candidate.

IMS Connect Extensions will not route messages to an ODBM target that is unavailable or have a capacity weight rating of zero. In this situation, IMS Connect Extensions will distribute the workload across the remaining eligible data stores according to their capacity weight ratings.

You can also use capacity weight ratings to automatically redistribute workload in response to changing periods of demand with the IMS Connect Extensions host
command environment for REXX. The member CEXRX61 in the SCEXSAMP library shows how to adjust capacity weights using a REXX exec. See "REXX automation samples for the IMS Connect Extensions host command environment" on page 223.

Procedure

1. Set a capacity weight rating (CWR) for every ODBM target definition associated with an ODBM routing rule. To do this, complete the following steps:
   a. From the IMS Connect Extensions primary menu, select option 1 Definitions.
   b. Select option 11 ODBM Targets.
   c. Select the ODBM target using line action S.
   d. Enter a value in the Capacity weight rating field. ODBM targets with greater capacity weight rating (CWR) will receive more transactions. See “Defining ODBM targets” on page 362.

   Note: A value of zero indicates that the ODBM target will be held in reserve and receive no workload.

   e. When you have finished making changes press the Exit function key (F3).

   Repeat this process for each ODBM target as required.

2. Activate workload balancing in the IMS Connect system definition. To do this, complete the following steps:
   a. From the IMS Connect Extensions primary menu, select option 1 Definitions.
   b. Select option 1 System Definitions.
   c. To edit the IMS Connect system definition you are interested in, enter line action S next to the name of the system.
   d. Enter a slash (/) next to the Activate Workload Balancing option.

   Note: If you have not already done so, you must also select the Activate IMS Connect Extensions, Activate Advanced Features, and Activate ODBM rules options.
Press the Exit function key (F3) to save and exit the panel.

3. Refresh the in-memory definitions you have modified. See "Refreshing in-memory definitions from the IMS Connect Extensions repository" on page 150.

What to do next

Activate the routing rule in IMS Connect. For more information, see "Activating ODBM routing rules in IMS Connect" on page 294.

Assigning ODBM routing rules to a plan

Routing plans provide a way to group ODBM routing rules that are intended to operate at the same time. You can use routing plans to create different routing configurations that can be dynamically activated as required.

Before you begin

Before you can activate ODBM routing rules in IMS Connect, you must first create some rules. See "Routing OTMA workload to one or more IMS data stores" on page 248.

About this task

A routing rule that is not assigned to a routing plan will always be in effect, except when it is overridden by a routing rule that contains a more specific message matching condition.

Procedure

1. Create a routing plan definition. To do this, complete the following steps:
   a. From the IMS Connect Extensions primary menu, select option 1 Definitions.
   b. Select option 16 Routing Plans.
   c. Type NEW and press Enter.
   d. Type a name and press Enter. The Routing Plan panel is displayed.
e. Enter a description for the routing plan.

f. Press the Exit function key (F3) to save and exit the panel.

2. Assign the required ODBM routing rules to the plan. To do this, complete the following steps for each rule you wish to assign:
   
a. From the IMS Connect Extensions primary menu, select option 1 Definitions.
   
b. Select option 12 ODBM Routing Rules.
   
c. Select an ODBM routing rule using line action $.
   
d. Use the the Prompt function key (F4) to select the routing plan name in the Routing Plan field.

   e. Press the Exit function key (F3) to save and exit the panel.

What to do next

Activate the routing plan in IMS Connect. For more information, see "Activating ODBM routing rules in IMS Connect."

Activating ODBM routing rules in IMS Connect

ODBM routing rules defined in the IMS Connect Extensions repository only take effect in IMS Connect once you have activated ODBM rules-based routing.

Before you begin

Before you can activate ODBM routing rules in IMS Connect, you must first create some rules. See "Routing ODBM workload to one or more IMS data stores" on page 282. If you wish to use a routing plan to organize your routing rules, see "Assigning ODBM routing rules to a plan" on page 293.
Procedure

1. If you have not already done so, activate ODBM routing rules in the IMS Connect system definition. To do this, complete the following steps:
   a. From the IMS Connect Extensions primary menu, select option 1 Definitions.
   b. Select option 1 System Definitions.
   c. To edit the IMS Connect system definition you are interested in, enter line action $ next to the name of the system.
   d. Enter a slash (/) next to the Activate ODBM rules option.

   Note: If you have not already done so, you must also select the Activate Advanced Features and Activate IMS Connect Extensions options.

2. To apply the changes you made in step 1 refresh the IMS Connect system. To do this, complete the followings steps:
   a. From the IMS Connect Extensions primary option menu, select option 2 Operations.
   b. Set View to option 2 Systems.
   c. Enter line action $X next to the desired system. The Commands panel is displayed.
   d. Select option 2 Refresh.
   e. Select option 1 System Definition and then press Enter.
   f. Select the command displayed on the Command Preprocessor panel by entering line action / next to the command.
3. Optional: If you have assigned one or more routing rules to a routing plan, activate the routing plan. To do this, complete the followings steps:
   a. Press the Exit function key (F3) until you to return to the Commands panel.
   b. Select option 4 Set.
   c. Select option 7 Routing Plan.
   d. Use the the Prompt function key (F4) in the field to the right of the **Activate ODBM Routing Plan** field to select a plan from a list of routing plans defined in the IMS Connect Extensions repository.
   e. Enter / to the left of the **Activate ODBM Routing Plan** field.

4. Refresh your ODBM routing rules. To do this, complete the followings steps:
   a. Press the Exit function key (F3) until you to return to the Commands panel.
   b. Select option 2 Refresh.
   c. Select option 8 ODBM Routing Rules and then press Enter.
   d. Select the command displayed on the Command Preprocessor panel by entering line action / next to the command.
e. Press Enter to submit the request. The result for each command is displayed in the Response column. A response of 0000 indicates that the command was processed successfully.

Note: Routing rules are also refreshed when IMS Connect is restarted.

5. Browse the message log for an IMS Connect system to review the rules that are now in-effect. See “Reviewing the active ODBM routing rules in IMS Connect.”

What to do next

To switch to a new routing plan, simply repeat steps 3 on page 296 and 4 on page 296. You only need to refresh the IMS Connect system when you first create a definition or when the IMS Connect Extensions repository definition has been modified.

Related tasks:

“Activating and deactivating ODBM routing plans” on page 197

Use the Update > ODBM Routing Plan context option in the Status Monitor of the Operations Console to activate or deactivate ODBM routing plan on several IMS Connect systems at once.

Related reference:

“SET command” on page 453

The SET host command for REXX activates or deactivates an OTMA or ODBM routing plan.

Reviewing the active ODBM routing rules in IMS Connect

When a system starts that has ODBM routing rules activated, a series of messages is written to CEXPRINT. These messages display the ODBM routing plan that is currently in effect (if any) and provides summarized information about the ODBM routing rules, ODBM routing lists, and ODBM targets that may be in use.

Note: More ODBM routing plans, ODBM routing rules, ODBM routing lists, and ODBM targets are likely to be defined in the IMS Connect Extensions repository than are in-effect on the system. The ODBM routing rules descriptions that are written to CEXPRINT show the in-memory rules and objects that are currently in effect.

ODBM routing rules and their dependent definitions are listed using message CEX5191I. The first message that is listed is displayed below:

04.40.16 CEX5191I --Begin ODBM routing rules descriptions---, refresh index=1---

If ODBM rules-based routing has been activated in the IMS Connect system definition, the following message is displayed:
04.40.16 CEX5191I ODBM Routing is active

If there is an active routing plan, it is listed next. In this example, PEAK is the active ODBM routing plan on this system.

04.40.16 CEX5191I ODBM Routing plan is PEAK

The ODBM routing rules that are in effect are listed next. In this example, the ODBM routing rule that is triggered by an alias name of DETS is shown. The request may contain any PSB name, which is indicated here using an asterisk (*). The target ODBM routing list is ODLSTB. The name of the OTMA routing rule is PLNRULEW.

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

The next rule in effect is for workload with an alias of PROD and a PSB name of AUTPSB11. The rule name is PHONRULE. Workload can be distributed to ODBM targets listed in ODBM routing list PHONLIST.

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

The final rule in effect is for workload with an alias of PROD and a PSB name prefixed with DFS. The rule name is DFSRULE. Workload can be distributed to ODBM targets listed in ODBM routing list DFSLIST with fallback targets in ODBM routing list DFSFALL.

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 DFSLIST DFSFALL
04.40.16 CEX5191I End of entries for ODBM routing rule
04.40.16 CEX5191I ---End ODBM routing rules descriptions---

Finally, the ODBM routing lists are displayed using message CEX5193I. This message shows the name of the ODBM routing list, its ODBM target (or targets), the name of the Common Services Library (CSL) Open Database Manager (ODBM) address space defined for that target, the name of the IMS alias associated with the ODBM, and listed in each and capacity weight rating (CWR) of the ODBM target.

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 DFSTARG DCA1OD DOP1 1
04.40.16 CEX5193I ODLSTB ODTRGB DCA1OD DOP2 1
04.40.16 CEX5193I PHONLIST PHONETGT DCA2OD DOP2 1
04.40.16 CEX5193I ---End ODBM routing list description---

Related concepts:
Browsing the message log for an IMS Connect on page 164

The message log for an IMS Connect system reveals activity within IMS Connect and IMS Connect Extensions. Use the message log to understand more about the status of archiving, to view error messages, or to display a history of commands that have been issued and their responses.

Related information:

- “CEX5191I” on page 507
- ODBM routing rule descriptions
- “CEX5193I” on page 507
- ODBM target list names

Controlling ODBM routing behavior via CEXCTLIN control options

You can also control aspects of default ODBM routing behavior via CEXCTLIN control options.

Before you begin

If you are using CEXCTLIN control options to adjust ODBM routing behavior, you should also set up one or more ODBM routing rules. See “Routing ODBM workload to one or more IMS data stores” on page 282.

About this task

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. For more information, see Chapter 26, “Control input data set options,” on page 561.

Tip: Member CEXCTL01 in the SCEXSAMP sample library contains the complete range of sample control input data set options. Use this sample to explore the effect of customizing different configuration options.

Procedure

1. Review the CEXCTLIN DD statement in the IMS Connect startup job and add the following options as desired:

   CEXROUTE

   The CEXROUTE option configures several routing options in IMS Connect Extensions. The following options apply to ODBM routing:

   **RBR_NOALIAS**

   Controls how ODBM rules-based routing functions if the inbound alias is not found in the active ODBM routing rules during the initial pass. Specify one of the following options:

   - **alias**
     
     If RBR_NOALIAS=destid is specified, the alias specified in the client request 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:** The ODBM routing rule that matches the value you specify for alias must not refer to a PSB name list.
PASS
If RBR_NOALIAS=PASS is specified, the DRDA request is passed through unmodified. This is the default.

ODBMDEFAULTS
The ODBMDEFAULTS option configures the default behavior for ODBM routing in IMS Connect Extensions. You can configure this option in the following way:

DEFAULTACTION
Sets the default action for ODBM routing when the routing exit cannot make a routing decision due to one 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.

The DEFAULTACTION parameter accepts the following options:

INPUTALIAS
If DEFAULTACTION=INPUTALIAS is specified, the alias received in the client request is used as the output alias and the ODBM name is set to blanks. This is the default.

Notes:
- This is the normal behavior of the HWSROUT0 exit supplied with IMS Connect.
- This is also the default 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.
  - During routing processing, a required ODBM routing rule could not be found.

BLANKALIAS
If DEFAULTACTION=BLANKALIAS is specified, the request is passed through with the ODBM name and alias both set to blanks.

2. To enable any options you selected in the previous step, restart IMS Connect.
Chapter 15. 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 Activate Security is enabled in the IMS Connect system definition, IMS Connect Extensions performs user ID and password validation by issuing a RACROUTE VERIFY request. If Activate ACEE Cache is enabled and a value specified in the Ageing interval field, the Accessor Environment Element (ACEE) created by the RACROUTE VERIFY is cached 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.

Activating ACEE caching

To activate ACEE caching, complete the following steps:

1. In the IMS Connect system definition, complete the following fields.
   a. Enter / (slash) next to the Activate Security field.
   b. Enter / (slash) next to the Activate ACEE Cache field.
   c. Enter a cache aging interval in the Ageing Interval field.

```
/ Activate Advanced Features
  - Activate Pacing
    Interval count ... 3
    Warning threshold ... 8          Reject threshold ... 0
/ Activate Session Message Limit
  Limit threshold ... 500
7 Activate Security
  / Activate ACEE Cache
    Ageing interval ... 60 (Min)
    Activate validation
      Security class ... 1. IMS Connect
      Validation type ... 1. IMS Connect
      2. IMS Connect + IP Address + Port

Figure 154. Activating the ACEE cache in the System Definition panel in the IMS Connect Extensions ISPF dialog
```

For further details, see “Defining IMS Connect systems” on page 320.

2. Refresh the IMS Connect system definitions. See “Refreshing in-memory definitions from the IMS Connect Extensions repository” on page 150.
Displaying ACEE cache usage statistics

To display the current ACEE cache usage statistics, use any of the following methods:

- Issue the **AC** line action in the Operations - Systems View ISPF dialog. See “Using the Operations dialog” on page 120.
- Use the Operations Console for z/OS Explorer. See “Viewing ACEE cache statistics” on page 199.
- Use the **QUERY** host command for REXX. See “**QUERY** commands” on page 436.

Lifecycle of a user ID in the ACEE cache

When IMS Connect Extensions receives a type 71 ENF signal from RACF 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 IMS Connect system definition. The ACEE is cleared after this time interval has passed.

ACEEs can also be purged from the cache in the following ways:

- Use the Security Commands dialog. See “Issuing security commands” on page 155.
- Use the Operations Console for z/OS Explorer. See “Clearing the ACEE cache” on page 200.
- Use the **CLEAR** host command for REXX. See “**CLEAR** command” on page 430.

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.

Modifying the size of the ACEE cache

The number of ACEEs in the cache is managed by the **SECURITY** option specified in the **CSEXCTLIN** data set. The number specified in the **CACHED_USER_MAX** parameter 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. See “**SECURITY** option” on page 568.

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.

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.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.xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.xxxx.zzzzz** 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.AB12.0000.0000.0000.0000.0000.3333.4444.05678
CEX.IPV6.ZZSEC01.AB12::0000.0000.3333.4444.05678
```

- **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 SECURITY option specified in the CEXCTLIN data set allows you to control several options that affect validation.

The VALIDATE_TRUSTED parameter determines whether or not to perform IP validation for messages that are considered trusted. A message is considered trusted if the OMUSR_TRSTUSR flag is set in the incoming message, or the message acquired the trusted attribute from an IP address rule. For more information on IP address rules, see "Creating workload rules for specific IP addresses."

The PWCase parameter determines how mixed-case passwords are handled.

For more information, see “SECURITY option” on page 568.

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.

Creating workload rules for specific IP addresses

IMS Connect Extensions allows you to create IP address rules for an IMS Connect system that can be used to treat messages coming from specific IP addresses as trusted users or to override the user ID supplied in a request message with another of your choosing.

Before you begin

To apply an IP address rule to one or more IMS Connect systems, the systems must be defined in your IMS Connect Extensions repository. See "Creating an initial configuration for IMS Connect Extensions” on page 32.

About this task

IP address rules can be applied independently of other IMS Connect Extensions security features, however it is highly recommended that you use IMS Connect Extensions access control to ensure that workloads are coming from authorized IP addresses. See "Defining IMS Connect systems” on page 320."
Procedure

1. Define one or more IP address rules in the IMS Connect Extensions repository. To do this, complete the following steps:
   a. From the primary menu in the IMS Connect Extensions ISPF dialog, select option 1.17 IP Address Rules. The IP Address Rules panel is displayed.
   b. To create a new rule, enter the NEW command. Type a name for the rule and then press Enter.
   c. Configure the rule using the IP Address Rule panel. The panel contains the following fields:
      - **Name**: The name of the IP address rule.
      - **Description**: A description for this IP address rule.

   **Apply rule to**
   Specifies the IMS Connect system or systems to which this rule will apply. Choose from the following options:
   - **1 System**
     Apply the rule to a single system. Use the the Prompt function key (F4) to select from a list of defined systems. To define an IMS Connect system, see “Defining IMS Connect systems” on page 320.
   - **2 Group**
     Apply the rule to a group of systems. Use the the Prompt function key (F4) to select from a list of defined groups. To define a group, see “Defining system groups” on page 355.
   - **3 All systems**
     Apply the rule to all IMS Connect systems.

   **Condition**
   Specifies the conditions that must be met for the rule to take effect.
   - **IP Address**
     Apply the rule to workloads originating from the specified IPv4 or IPv6 address (for example, 192.23.88.1 for an IPv4 address or 2001:0db8:85a3:0:0:0:0:0 for an IPv6 address) or from a range of IP addresses by using a trailing wildcard (*). The wildcard must appear at the end of the IP address and immediately after the final node separator character in the address (for example, 192.55.0:* for an IPv4 address or 0:0:0:0::8f00:* for an IPv6 address). A wildcard must not be mixed with an IPv6 address containing two consecutive colons (::). IPv6 addresses may not contain IPv4 dot-decimal notation.

   **Options**
   The following preconditioning options may be applied:
   - **Override User id**
     Override the user ID defined in the request message with the specified user ID. Workloads will be processed using the specified user ID, regardless of what is contained in the request.
   - **Only override blank userids**
     When this option is selected, the user ID specified in
the **Override User id** field is only used when no user ID is supplied in the request message.

**Trust user id**

Treat all messages from this IP address or IP address range as trusted users in IMS Connect. SAF password verification in IMS Connect is not performed.

---

**Figure 155. Configuring an IP address rule in the IMS Connect Extensions ISPF dialog**

*d.* When you have finished making changes, press the Exit function key (F3) to save your definition. The new definition is displayed in the IP Address Rules panel.

2. For each IMS Connect system that requires an IP address rule, you will need to activate IP address rule checking in its system definition. To do this, complete the following steps:

   a. From the Definitions panel in the IMS Connect Extensions ISPF dialog, select option 1 **System Definitions**. The System Definitions panel is displayed.

   b. Select an IMS Connect system using line action S. The System Definition panel is displayed.

   c. To activate IP address rule checking on this IMS Connect system, enter a slash (/) next to the **Activate IP Address Rules** field.

---

**Figure 156. Activating IP address rules in the IMS Connect Extensions ISPF dialog**
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. You can do this using one of the following methods:

**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. See “Defining user exits” on page 336.

**Definitions maintenance utility**

In the **DEFN EX ADD** command, specify:

```
IRM_OFFSETS=(EXIT,EXIT_NAME=name)
```

See “User exit (EX) **ADD command**” on page 386.

**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.V3R1M0.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 16. 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. From the primary menu in the IMS Connect Extensions ISPF dialog, select option 1.2 User Exits.
2. Enter S next to the exit for which you want to perform custom translation.
3. 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.

4. Press the Exit function key (F3) to save the definition.
5. 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 157. Translating messages between Windows 1252 encoding and UK EBCDIC encoding.

Related reference:
"Defining user exits" on page 336

The User Exits panel allows you to configure user message exits. To access this ISPF panel, select option 1.2 User Exits from the IMS Connect Extensions primary menu. You can define the standard IBM-supplied exits to IMS Connect Extensions using a LOAD command.
Part 5. Administering IMS Connect Extensions definitions

These topics describe how to maintain IMS Connect Extensions definitions.

Topics:

- Chapter 17, “Administering IMS Connect Extensions definitions with the ISPF dialog,” on page 315
- Chapter 18, “Definition utilities,” on page 375
Chapter 17. Administering IMS Connect Extensions definitions with the ISPF dialog

This section describes how to create IMS Connect Extensions repository definitions using the ISPF dialog.

To manage IMS Connect Extensions repository definitions, select option 1 Definitions on the primary menu of the IMS Connect Extensions ISPF dialog.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>System Definitions</td>
</tr>
<tr>
<td>2</td>
<td>User Exits</td>
</tr>
<tr>
<td>3</td>
<td>Datastores</td>
</tr>
<tr>
<td>4</td>
<td>Datastore Groups</td>
</tr>
<tr>
<td>5</td>
<td>Affinity Lists</td>
</tr>
<tr>
<td>6</td>
<td>Applications</td>
</tr>
<tr>
<td>7</td>
<td>Transactions</td>
</tr>
<tr>
<td>8</td>
<td>System Groups</td>
</tr>
<tr>
<td>9</td>
<td>OTMA Routing Rules</td>
</tr>
<tr>
<td>10</td>
<td>OTMA Routing Lists</td>
</tr>
<tr>
<td>11</td>
<td>ODBM Targets</td>
</tr>
<tr>
<td>12</td>
<td>ODBM Routing Rules</td>
</tr>
<tr>
<td>13</td>
<td>ODBM Routing Lists</td>
</tr>
<tr>
<td>14</td>
<td>PSB Name Lists</td>
</tr>
<tr>
<td>15</td>
<td>Transaction Lists</td>
</tr>
<tr>
<td>16</td>
<td>Routing Plans</td>
</tr>
<tr>
<td>17</td>
<td>IP Address Rules</td>
</tr>
<tr>
<td>S</td>
<td>Definitions Setup</td>
</tr>
</tbody>
</table>

*Figure 158. The Definitions panel in the IMS Connect Extensions 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 for OTMA rules-based routing**

OTMA rules-based routing in IMS Connect Extensions requires you to create one or more routing rules that depend on several supporting definitions. The structure of this relationship can be seen in the following diagram.
The following minimum set of definitions are required to create an OTMA routing rule for a single IMS Connect system. Use the following sequence to create your definitions:

1. IMS Connect system (option 1 System Definitions)
2. IMS data stores (option 3 Datastores)
3. OTMA routing list containing your IMS data stores (option 10 OTMA Routing Lists)
4. OTMA routing rule referencing all of the above (option 9 OTMA Routing Rules)

The following definitions are optional and may be added later as required:

- IMS Connect system group containing a collection of IMS Connect systems (option 8 System Groups)
- Transaction list (option 15 Transaction Lists)
- Routing plan (option 16 Routing Plans)

For more information, see Chapter 12, “OTMA workload routing in IMS Connect,” on page 245.

**Definitions for ODBM rules-based routing**

ODBM rules-based routing in IMS Connect Extensions requires you to create one or more routing rules that depend on several supporting definitions. The structure of this relationship can be seen in the following diagram.

---

**Figure 159. The structure of an OTMA routing rule in IMS Connect Extensions**

The following minimum set of definitions are required to create an OTMA routing rule for a single IMS Connect system. Use the following sequence to create your definitions:

1. IMS Connect system (option 1 System Definitions)
2. IMS data stores (option 3 Datastores)
3. OTMA routing list containing your IMS data stores (option 10 OTMA Routing Lists)
4. OTMA routing rule referencing all of the above (option 9 OTMA Routing Rules)

The following definitions are optional and may be added later as required:

- IMS Connect system group containing a collection of IMS Connect systems (option 8 System Groups)
- Transaction list (option 15 Transaction Lists)
- Routing plan (option 16 Routing Plans)

For more information, see Chapter 12, “OTMA workload routing in IMS Connect,” on page 245.

**Definitions for ODBM rules-based routing**

ODBM rules-based routing in IMS Connect Extensions requires you to create one or more routing rules that depend on several supporting definitions. The structure of this relationship can be seen in the following diagram.
The following minimum set of definitions are required to create an ODBM routing rule for a single IMS Connect system. Use the following sequence to create your definitions:

1. IMS Connect system (option 1 System Definitions)
2. ODBM targets (option 11 ODBM Targets)
3. ODBM routing list containing your ODBM targets (option 13 ODBM Routing Lists)
4. ODBM routing rule referencing all of the above (option 12 ODBM Routing Rules)

The following definitions are optional and may be added later as required:

- IMS Connect system group containing a collection of IMS Connect systems (option 8 System Groups)
- PSB name list (option 14 PSB Name Lists)
- Routing plan (option 16 Routing Plans)

For more information, see Chapter 14, “ODBMB workload routing in IMS Connect,” on page 279.

**Legacy OTMA routing methods**

For customers who are using transaction routing and primary datastore routing, the following definitions apply. For more information, see Chapter 31, “Legacy OTMA routing methods,” on page 607.

**Transaction routing**

- IMS Connect system (option 1 System Definitions)
- IMS Connect system group containing a collection of IMS Connect systems (option 8 System Groups)
- IMS data stores (option 3 Datastores)
- IMS data store group (option 4 Datase Group)
• Affinity List (option 5 Affinity Lists)
• Applications (option 6 Applications)
• Transactions (option 7 Transactions)

Primary datastore routing
For customers who are using the legacy “primary datastore” routing method, the following definitions apply.
• IMS Connect system (option 1 System Definitions)
• IMS Connect system group containing a collection of IMS Connect systems (option 8 System Groups)
• IMS data stores (option 3 Datastores)
• IMS data store group (option 4 Datastore Groups)
• Affinity List (option 5 Affinity Lists)

Related tasks:
“Creating an initial configuration for IMS Connect Extensions” on page 32
Use the Definitions Setup dialog to take up definitions from the IMS Connect configuration member into an IMS Connect Extensions repository. This provides you with an initial configuration and a basis from which you can start customizing IMS Connect Extensions.

Related reference:
“Refreshing in-memory definitions from the IMS Connect Extensions repository” on page 150
Use the Refresh Commands panel to reload IMS Connect Extensions in-memory definitions for the selected system from definitions stored in the IMS Connect Extensions repository. To access this ISPF panel, select option 2 Refresh on the Commands panel.

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.
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 panel to create a new definition for an IMS Connect and to specify 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. To access this ISPF panel, select option 1.1 **System Definitions** from the IMS Connect Extensions primary menu.

The System Definitions panel is displayed with a list of IMS Connect systems defined in your IMS Connect Extensions repository. Use the Filter field to list only items that match a filter pattern.

<table>
<thead>
<tr>
<th>Command</th>
<th>System Definitions</th>
<th>Row 1 from 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filter ... (Blank or pattern)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Enter "/" to select action

/  Name   Description       -----  Changed  -----  ID
-  HWS1   Production IMS Connect  2018/07/31 13:39:49  AJL
-  HWS2   Test IMS Connect      2018/07/31 13:40:04  AJL

---

*Figure 161. The System Definitions panel in the IMS Connect Extensions ISPF dialog*

To define a new IMS Connect system in the IMS Connect Extensions repository, enter **NEW** on the **Command** line. Enter the 1-8 character name of your IMS Connect system and press Enter. The System Definition panel is displayed.

**Note:** The name must be the same as the HWS ID parameter in the IMS Connect configuration member. If the name does not match the HWS ID parameter, IMS Connect Extensions will not activate. A system definition cannot be shared by multiple IMS Connect systems.
The panel displays the following information:

**Name**  The name of the IMS Connect system.

**Description**  A description of this system.

**Console settings**  The console listener settings determine how ISPF Operations dialog clients, Operations Console for z/OS Explorer 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 file.
console port must not be used by any other job or subsystem in the LPAR on which this IMS Connect system runs.

**Tip:** The console port number is reported in message "CEX5020I" on page 492.

**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. See “SECURITY option” on page 568.

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. See “Defining IMS Connect systems” on page 320.

**Activate Event Collection**

This field allows you to activate and deactivate the Event Collection feature for IMS Connect Extensions. For more information, see Part 2, “IMS Connect event collection, reporting, and forwarding,” on page 53. Event Collection provides the data for you to run performance and availability reports.
**Note:** To display activity for a system, event collection must be active and **Collection Level** must be set to a value of 1 or higher.

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

Configure IMS Connect Extensions event collection using the following options:

**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.

**Tip:** For more information on how you can use the collected event data to report on IMS Connect performance, resource usage, and transit event tracing, see “IMS Connect Extensions event collection” in the *IBM IMS Performance Analyzer for z/OS Report Reference*.

**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 reporting easier in IBM IMS Performance Analyzer for z/OS.

**Activate Publisher API**

The IMS Connect Extensions publisher API provides an interface that publishes captured data to an external application. See Chapter 20, “IMS Connect Extensions publisher API,” on page 463.

Enter / (slash) to activate the IMS Connect Extensions 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 IMS Connect Extensions publisher API clients. If you change the maximum number of clients through the IMS Connect Extensions publisher API console, then the number will revert to this value when IMS Connect Extensions is restarted.

See “Setting the maximum number of publisher API clients” on page 172.

Note: You must restart IMS Connect to apply changes to this option.

Active Data Set
Enter / (slash) to specify allocation information for the active journal data sets, edit or create the active journal data set template, and any other parameters needed at initialization time. See “Defining active journal data set processing options” on page 329.

Archive Data Set
Enter / (slash) to specify settings 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, to edit or create the Archive Journal data set template, and any other parameters needed at initialization time. See “Defining the archive journal data set template” on page 332.

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. See “Activating command access to an IMS Connect system” on page 235. 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 CEXapplid.verb.object in the SAF resource class OPERCMDS.

When Activate 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 RACF PassTickets for security authentication. Use PassTickets if you do not want user passwords to be transmitted.

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. Use the following options to configure these advanced features:

**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 defined in the **Warning threshold** and **Reject threshold** fields as well as thresholds contained within the IMS data store and data store group definitions. See “Defining IMS data stores” on page 339 and “Defining groups of IMS data stores” on page 344. Based on 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 message 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 message rejection 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.
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. See “Rebalancing sessions across IMS Connect systems” on page 276.

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**

Enter / (slash) to activate security management.

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.

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. See “Caching user credentials to improve performance” on page 301.

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
Enter / (slash) to 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. See "Validating an incoming OTMA transaction or DRDA request" on page 302.

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.

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.

Activate IP Address Rules
This field specifies whether IP address rules are active. IP address rules allow you to specify what preconditioning is to take place on workloads coming from specific IP addresses. See "Defining IP address rules" on page 372.

Routing schemes
Use the following options to activate or deactivate message routing:

Activate OTMA rules
Enter / (slash) to activate OTMA rules-based routing for this IMS Connect system.

OTMA routing rules allows IMS Connect Extensions to route IRM messages to collections of IMS data stores called OTMA routing lists. See "Defining OTMA routing rules" on page 356.

Activate ODBM rules
Enter / (slash) to activate ODBM rules-based routing for this IMS Connect system.
ODBM routing rules 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. See “Defining ODBM routing rules” on page 364.

Activate Transaction Routing
Enter / (slash) to activate transaction routing for this IMS Connect system.

Transaction routing is a legacy routing method that allows IMS Connect Extensions to select a destination IMS data store for an incoming IRM message based on the transaction code and destination ID specified by the message. See “Transaction routing” on page 607.

Routing options
Use the following options to configure message routing:

Define Applications for system
This option allows you to specify the applications that are associated with this IMS Connect system when using transaction routing or override transaction processing options.

Enter / (slash) to display the Applications panel. See “Defining applications controlled by an IMS Connect system” on page 335.

Activate Workload Balancing
Enter / (slash) to activate workload balancing for this IMS Connect system. Use this feature to balancing the workload across systems and ensuring both availability and responsiveness for remote clients.

Workload balancing allows IMS™ Connect Extensions to redirect OTMA workload to IMS data stores and ODBM workload to ODBM targets according to their capacity weight ratings (CWR). To define the capacity weight rating for an IMS data store, see “Defining IMS data stores” on page 339. To define the capacity weight rating for an ODBM target, see “Defining ODBM targets” on page 362.

Activate Statistics Collection
Enter / (slash) to activate statistics collection for this IMS Connect system.

Statistics collection must be active for information to be displayed by the IMS Connect Extensions Status Monitor. You can view these statistics using the ISPF dialog or z/OS Explorer.

Edit IMS Connect configuration
Enter / (slash) to display the IMS Connect Configuration panel. This option allows you to view or edit the IMS Connect configuration member for this IMS Connect system. See “Editing the IMS Connect configuration” on page 335.

Note: 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**.

**Related concepts:**

“Confi guring event collection and journals” on page 61

Before you can collect and report on IMS Connect events, you must configure your systems.

**Related tasks:**

“Rebalancing sessions across IMS Connect systems” on page 276

The **Session Message Limit** option in the IMS Connect system defi nition allows you to specify the maximum number of input messages for a persistent session before the session is automatically closed. In an IMS sysplex environment, if you design your applications to automatically reconnect when the session is closed, new persistent sessions will be routed to the IMS Connect with the lowest session totals. This feature is particularly useful when an IMS Connect system is being brought back online to quickly rebalance your persistent sessions across IMS Connect systems.

**Related reference:**

“Overriding IMS Connect system settings” on page 158

Use the Set System Definition panel to temporarily override IMS Connect Extensions in-memory defi nition settings for an IMS Connect system. The IMS Connect system defi nition in the IMS Connect Extensions repository remains unchanged. To access this ISPF panel, select option 1 **System De fi nition** on the Set Commands panel.

**Defining active journal data set processing options**

The Active Journal Data Set panel contains processing options for the active IMS Connect Extensions journal. To access this ISPF panel, select option 1.1 **System De fi nitions** from the IMS Connect Extensions primary menu and then select **Active Data Set**.
Use the following fields in the Active Journal Data Set panel to configure the active IMS Connect Extensions journal:

**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 Pnn where nn represents the active journal number in the range 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 IMS Connect system name specified in the System Definition panel 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. If you specify one data set you must specify REUSE in the **Journal full option** field.

**Archive manager**

Enter / (slash) next to the **Activate journal archiving** to use the Archive Manager utility to copy event records from active journal data sets when they become full to archive journal data sets on tape or DASD. To configure archive journal data sets, see "Defining the archive journal data set template” on page 332.
Journal full option
This field allows you to determine what happens if all active data sets and the optional overflow data set 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 for the active data set. For example, TAPE or SYSDA.

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**  The data set size in tracks
- **CYLS**  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 Journal full option field is taken.

Overflow size
The size of the overflow data set as a multiple of the value specified in the Primary quantity field. Enter a value in the range 1 - 99. The default value is 4 (that is, 4 times the value specified in the Primary quantity field).
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.

Note: 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. See “Creating an
archive JCL skeleton” on page 63.

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 tasks:
“Configuring active journals” on page 68
Use the Active Journal Data Set panel to set active journal options and the archive
Job Statement Information and JCL skeleton.

Related reference:
“Archive Manager utility” on page 587
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.

Defining the archive journal data set template
The Archive Journal Data Set Template panel contains options for the archive IMS
Connect Extensions journals. To access this ISPF panel, select option 1.1 System
Definitions from the IMS Connect Extensions primary menu and then select
Archive Data Set.

Tip: To activate journal archiving, see “Defining active journal data set processing
options” on page 329.
Use the following fields in the Archive Journal Data Set Template panel to configure the active IMS Connect Extensions journal:

**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.

**For use with V1R1 archive 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.

**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 tasks:**

- "Creating an archive journal template" on page 69
  Use the Archive Journal Data Set Template panel to enter the archive data set name you want to use.
Related reference:

“Archive Journal cleanup utility” on page 585

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.

Defining applications controlled by an IMS Connect system

The System Definition - Applications panel contains a list of applications controlled by the IMS Connect system. To access this ISPF panel, select option 1.1 System Definitions from the IMS Connect Extensions primary menu and then select Define applications for name where name is the name of the IMS Connect system.

```
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FINANCE</td>
<td>Finance transactions</td>
</tr>
<tr>
<td>HR</td>
<td>Human resources transactions</td>
</tr>
<tr>
<td>PAYROLL</td>
<td>Payroll transactions</td>
</tr>
<tr>
<td>SALES</td>
<td>Sales transactions</td>
</tr>
</tbody>
</table>
```

Figure 165. The System Definition - Applications panel in the IMS Connect Extensions ISPF dialog

Applications associate the IMS Connect system with the transaction codes it can route, and are used for transaction routing and other transaction level features. On this panel, you can modify the application list in the following ways:

- In the Name field, enter an application name or use the Prompt function key (F4) to select from a list of defined applications. To define an application, see “Defining applications” on page 347.
- To insert a new application name, enter line action I.
- To remove an application from the list, enter line action D next to the application you want to remove.
- To save your changes, press the Exit function key (F3).

Editing the IMS Connect configuration

The Configuration panel allows you to view and edit the IMS Connect configuration member. To access this ISPF panel, select option 1.1 System Definitions from the IMS Connect Extensions primary menu and then select Edit IMS Connect configuration.
Use the following fields in the Configuration panel to edit the IMS Connect configuration member:

**Name**  The name of the IMS Connect system.

**Description**  The description of the system 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.

**Configuration member's output mode**
Enter one of the following values:

1  View the configuration member.
2  Edit the configuration member.

---

**Defining user exits**

The User Exits panel allows you to configure user message exits. To access this ISPF panel, select option 1.2 **User Exits** from the IMS Connect Extensions primary menu. You can define the standard IBM-supplied exits to IMS Connect Extensions using a **LOAD** command.

The User Exits panel is displayed with a list of exits defined in your IMS Connect Extensions repository. Use the **Filter** field to list only items that match a filter pattern.
To define a new user exit, enter **NEW** on the **Command** line. Enter the 1-8 character name of your exit and press Enter. The User Exit panel is displayed.

<table>
<thead>
<tr>
<th>File</th>
<th>Menu</th>
<th>Settings</th>
<th>Help</th>
</tr>
</thead>
<tbody>
<tr>
<td>EDIT</td>
<td>User Exit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Command ==&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Name**: . . . : CEXSVCO1

**Description**: . SERVICE EXIT, 4 byte prefix

/ **Activate Exit Definition**

Choose how IRM offsets are determined:

1. Use default IRM offsets as referenced in the IMS Connect sample exit

2. Use IRM offsets (0 or 28-32759) as defined below:
   - User ID Offset . . . . 0
   - Password Offset . . . . 0
   - Group Offset . . . . 0
   - APPLname Offset . . . . 0

3. Use supplementary exit named below:
   - Pre-Processing Exit .

**Message options**:

/ Use a 4-byte length prefix for messages instead of 2 bytes
   - Extend length of RSM for error information

**Message ID support**:

MSG ID1 . . . 2 1. ASCII 2. EBCDIC MSG ID2 . . . 1 1. ASCII 2. EBCDIC

**Message translation**:

- Activate translation . . client CCSID server CCSID
- On error . . . 1 1. Disable user exit 2. Continue processing

*Figure 168. The User Exit panel in the IMS Connect Extensions ISPF dialog*

The panel displays the following information:

**Name** The name of the user exit.

**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 308.

Use a 4-byte length prefix for messages instead of 2 bytes
The acceptable values are:
- l 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 16, “Customizing IMS Connect message translation,” on page 311 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
"Definition list panels: common features" on page 318, 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.

---

**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.

---

**Related tasks:**

- Chapter 16, “Customizing IMS Connect message translation,” on page 311
- 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.

- “Activating the client services exit” on page 597
- This topic explains how to activate the client services exit (CEXSVCO1). This exit enables you to extend the capabilities of IMS Connect clients.

---

**Related reference:**

- “Managing user exits” on page 147
- Use the User Exit Commands panel to add, delete, enable, disable, and reload user exits without interrupting the execution of IMS Connect. To access this ISPF panel, select option 1 Exits on the Commands panel.

---

**Defining IMS data stores**

The Datastores panel allows you to define the IMS data stores defined and controlled by IMS Connect in the IMS Connect Extensions repository. To access this ISPF panel, select option 1.3 Datastores from the IMS Connect Extensions primary menu.
The Datastores panel is displayed with a list of IMS data stores defined in your IMS Connect Extensions repository. Use the Filter field to list only items that match a filter pattern.

To define a new IMS data store in the IMS Connect Extensions repository, enter NEW on the Command line. Enter the 1-8 character name of your IMS data store and press Enter. The Datastore panel is displayed.

The panel displays the following information:

**Name**  The name of the IMS data store.

**Description**  A description of this IMS data store.
Datastore Group
The name of a group that this IMS data store belongs to. Use groups for primary IMS data store routing, pacing and reporting.

This field is required if using the IMS data store as a primary for an IMS Connect system. Use the Prompt function key (F4) to get a list of defined IMS data store groups. To define a group, see “Defining groups of IMS data stores” on page 344.

Use as Primary Datastore for IMS Connect System
Set the IMS data store as a primary IMS data store for a selected IMS Connect system. Messages are routed to this IMS data store if the original IMS data store set in the message is the same as the name of the IMS data store group and the message is handled by the selected system.

Note: Within a IMS data store group, only one IMS data store 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 IMS data store.

Transaction Pacing monitors incoming transaction request rates according to thresholds assigned to this IMS data store. Based upon these thresholds, IMS Connect Extensions either issues warning messages or rejects transactions. The acceptable field values are:

/ Pacing is activated for this IMS data store. Where pacing threshold settings have been defined, these will be honored. Warning messages or transaction rejections may occur.

Note: Pacing must also be active in the system definition. See “Defining IMS Connect systems” on page 320.

blank Transaction Pacing is not active for this IMS data store

Warning Threshold
This field defines the Warning Transaction threshold for this IMS data store. 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 IMS data store level, warning thresholds use the Interval Count as defined in the system definition. See “Defining IMS Connect systems” on page 320.

The acceptable field values are any number less than 999999. A value of 0 deactivates warning processing for this IMS data store.

Reject Threshold
This field defines the Reject Transaction rate for this IMS data store. 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 IMS data store level, reject thresholds use the Interval Count as defined in the system definition. See “Defining IMS Connect systems” on page 320.
The acceptable field values are any number less than 999999. A value of 0 deactivates reject processing for this IMS data store.

**Activate Transaction Routing**
This field allows you to activate and deactivate the Transaction Routing feature for this IMS data store.

Transaction Routing allows IMS Connect Extensions to alter the target IMS data store that will process the incoming transaction request. See “Candidate datastore list” on page 611 for detailed examples of how Transaction Routing options work.

If transaction routing is inactive then all transaction affinity and data store affinity rules are bypassed and the original target data store defined on the incoming transaction request is used.

The acceptable field values are:

/ Transaction routing is active for this IMS data store. IMS Connect Extensions will alter the target IMS data store if the combinations of transaction affinity and IMS data store affinity allow.

blank Transaction routing is not active for this IMS data store.

**Route transactions to**
This field allows you to define which IMS data stores (if any) can share work with the IMS data store if transaction routing is active for this IMS data store.

The acceptable field values are:

1 All IMS data stores in this IMS Connect system can support this IMS data store.

2 One IMS data store only. Use the Prompt function key (F4) to get a list of defined IMS data stores.

3 Any IMS data store defined in a group. Use the Prompt function key (F4) to get a list of defined groups. To define a group, see “Defining groups of IMS data stores” on page 344.

4 Any IMS data store defined in an affinity list. Use the Prompt function key (F4) to get a list of defined affinity lists. To define an affinity list, see “Defining affinity lists” on page 346.

**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 615 for details.

The acceptable field values are:

/ Allow routing of Send Only requests for this IMS data store. Send Only requests with this IMS data store 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 IMS data store 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 615 for details.

The acceptable field values are:

/  Allow routing of RTPIPE requests to an affinity list of eligible IMS data stores. The routing of RTPIPE requests relies on Super Member routing. Rules-based routing, primary IMS data store 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 IMS data stores that are candidates for routing of Resume TPIPE (RTPIPE) requests when Super Member routing is activated for this IMS data store. When pairing Send Only and Resume TPIPE requests, ensure that the affinity list for Super Member routing is compatible with that for transaction routing. To define an affinity list, see “Defining affinity lists” on page 346.

Capacity Weight Rating
This field allows you to define the capacity weight rating (CWR) for this IMS data store. The rating is used by workload balancing to reroute incoming messages to alternate IMS data stores.

Acceptable field values are numbers in the range 1 - 100 and 0. A value of zero has a special meaning. It indicates that the IMS data store is not a candidate for routing.

The higher the rating number, the more workload this IMS data store can handle. Weightings are relative to other IMS data store definitions.

Examples:
1. If you have 3 IMS data stores 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 IMS data stores.
2. If you have 2 IMS data stores 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 IMS data store with the higher weighting.

Related tasks:
“Setting the capacity of an IMS data store for OTMA workloads” on page 254

Workload balancing allows IMS Connect Extensions to redirect incoming transaction requests to IMS data stores according to their capacity weight rating (CWR), thereby balancing the transaction workload across multiple IMS systems and ensuring both availability and responsiveness for the remote client.

Related reference:
“Overriding IMS data store definitions” on page 159

Use the Set Datastore panel to temporarily override IMS Connect Extensions in-memory definition settings for an IMS data store. The IMS data store definition in the IMS Connect Extensions repository remains unchanged. To access this ISPF panel, select option 3 Datastore on the Set Commands panel.

Defining groups of IMS data stores

The Datastore Groups panel allows you to view groups of IMS data stores controlled by IMS Connect Extensions. To access this ISPF panel, select option 1.4 Datastore Groups from the IMS Connect Extensions primary menu.

The Datastore Groups panel is displayed with a list of IMS data store groups defined in your IMS Connect Extensions repository. Use the Filter field to list only items that match a filter pattern.

To define a new group, enter NEW on the Command line. Enter the 1-8 character name of your list and press Enter. The Datastore Group panel is displayed.

Note: To add an IMS data store to the group, use the Datastore panel. See “Defining IMS data stores” on page 339.

Name The name of the IMS data store group.
Description
A description of this group

Activate Pacing
This field allows you to activate and deactivate the transaction pacing feature for this group.

Transaction pacing monitors incoming transaction request rates according to thresholds assigned to this 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 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 in the system definition. See “Defining IMS Connect systems” on page 320.

blank Transaction pacing is not active for this group.

The following fields apply to transaction pacing:

Warning Threshold
This field defines the Warning Transaction threshold for this 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 group level, warning thresholds use the Interval Count as defined in the system definition. See “Defining IMS Connect systems” on page 320.

The acceptable field values are any number less than 999999. A value of 0 deactivates warning processing for this group.

Reject Threshold
This field defines the reject transaction rate for this 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 group level, reject thresholds use the Interval Count as defined in the system definition. See “Defining IMS Connect systems” on page 320.

The acceptable field values are any number less than 999999. A value of 0 deactivates reject processing for this group.

Commands
You can use the following commands in the Command field:

LOCATE string
Locate the specified IMS data store definition.

NEW Create a new IMS data store definition using the current group.

REFRESH
Refresh the list of IMS data stores.

RESET Clear line actions.
**Defining affinity lists**

The Affinity Lists panel allows you to group IMS data stores into logical group for the legacy transaction routing feature. To access this ISPF panel, select option 1.5 *Affinity Lists* from the IMS Connect Extensions primary menu.

The Affinity Lists panel is displayed with a list of applications defined in your IMS Connect Extensions repository. Use the Filter field to list only items that match a filter pattern.

```
----  Row 1 from 1  ----
Command ====> Affinity Lists  Scroll ====> PAGE
Filter . . .  (Blank or pattern)
Enter "/" to select action

/ Name       Description  ----  Changed  ----  ID
    AFL1       My affinity list  2018/08/02 15:03:23  AXL

********************************************** Bottom of data **********************************************
```

*Figure 174. The Affinity Lists panel in the IMS Connect Extensions ISPF dialog*

To define a new affinity list, enter **NEW** on the Command line. Enter the 1-8 character name of your list and press Enter. The Affinity List panel is displayed. In this dialog you will assign one or more IMS data stores to the list.

```
---  Row 1 to 2 of 2  ---
Command ====> Affinity List  Scroll ====> PAGE

Name . . . : AFL1
Description . . My affinity list

Enter "/" to select action

- Datastore + Description
  - IMSA  Payroll system A
  - IMSB  Payroll system B

********************************************** Bottom of data **********************************************
```

*Figure 175. The Affinity List panel in the IMS Connect Extensions ISPF dialog*

The panel displays the following information. Enter line action / next to an IMS data store to insert additional names and to see a complete list of options:

**Name**  The name of the affinity list.

**Description**  A description of this affinity list.

**Datastore and Description**  The name of an IMS data store. Use the Prompt function key (F4) to get a list of defined IMS data stores. To define an IMS data store, see “Defining IMS data stores” on page 339.

**Related concepts:**
Transaction routing is a legacy process by which IMS Connect Extensions selects a destination IMS data store for an incoming IMS request message (IRM) based on the transaction code and destination ID specified by the message.

### Defining applications

The Applications panel 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. To access this ISPF panel, select option 1.6 **Applications** from the IMS Connect Extensions primary menu.

The Applications panel is displayed with a list of applications defined in your IMS Connect Extensions repository. Use the Filter field to list only items that match a filter pattern.

**Figure 176. The Applications panel in the IMS Connect Extensions ISPF dialog**

To define a new application, enter **NEW** on the **Command** line. Enter the 1-8 character name of your application and press Enter. The Application panel is displayed.

**Figure 177. The Application panel in the IMS Connect Extensions ISPF dialog**
The panel displays the following information:

**Name**  The name of this application.

**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 data stores that will process the incoming transaction request. See “Candidate datastore list” on page 611 for detailed examples of how transaction routing options work.

If transaction routing is inactive then all transaction affinity and data store affinity rules are bypassed and the original target data store defined on the incoming transaction request is used.

The acceptable field values are:

1. Transaction routing is active for this application. IMS Connect Extensions will alter the target IMS data store if the combinations of transaction affinity and IMS data store affinity allow.

2. Transaction routing is not active for this application

Transaction routing has the following options:

**Route transactions to**

This field allows you to define which IMS data stores (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 IMS data stores in this IMS Connect system can support this application.

2. One IMS data store only. Use the Prompt function key (F4) to get a list of defined IMS data stores. To define an IMS data store, see “Defining IMS data stores” on page 339.

3. Any IMS data store defined in a group. Use the Prompt function key (F4) to get a list of defined groups. To define a group of IMS data stores, see “Defining groups of IMS data stores” on page 344.

4. Any IMS data store defined in an affinity list. Use the Prompt function key (F4) to get a list of defined affinity lists. To define an affinity list, see “Defining affinity lists” on page 346.

**Routing Error processing**

This field defines the processing options when an IMS data store is not available to process the incoming transactions.

The acceptable field values are:

1. Use the original destination IMS data store in the message.

2. Reject the transaction.
**Defining transactions**

The Transactions panel allows you to define transactions to be routed by IMS Connect Extensions through the legacy transaction routing feature. To access this ISPF panel, select option 1.7 **Transactions** from the IMS Connect Extensions primary menu.

The Transactions panel is displayed with a list of transactions defined in your IMS Connect Extensions repository. Use the **Filter** field to list only items that match a filter pattern.

<table>
<thead>
<tr>
<th>Command</th>
<th>Transactions</th>
<th>Scroll</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filters . . . (Blank or pattern)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Enter &quot;/&quot; to select action</td>
<td></td>
<td></td>
</tr>
<tr>
<td>* Name</td>
<td>Description</td>
<td>-----</td>
</tr>
<tr>
<td>2018/07/20 13:52:38</td>
<td>AJL</td>
<td></td>
</tr>
</tbody>
</table>

| Figure 178. The Transactions panel in the IMS Connect Extensions ISPF dialog |

To define a new transaction in the IMS Connect Extensions repository, enter **NEW** on the **Command** line. Enter the 1-8 character name of your definition and press Enter. The Transaction panel is displayed.

**Tips:**

- Transaction names in IMS Connect Extensions can be specified with a prefix instead of a fully qualified transaction name. This allows you to define a set of rules for a group of similarly named transactions. For example, you could define a transaction named PAY* for all payroll transactions. You can even define a transaction named * as the default definition for all transactions.
• You can also 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.

• With transaction routing enabled, 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 named PAR+, PART and PART*, an incoming message for transaction PART will match the PART definition, but it will not match PART* or PAR*.

• You can also take-up transactions from IMS MODBLKS. See “Transaction take-up from MODBLKS” on page 353.

The panel displays the following information:

- **Name** The name of this transaction definition.
- **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.

---

**Figure 179. The Transaction panel in the IMS Connect Extensions ISPF dialog**

The panel displays the following information:

- **Name** The name of this transaction definition.
- **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 611 for detailed examples of how Transaction Routing options work.

If transaction routing is inactive then all transaction affinity and data store affinity rules are bypassed and the original target data store 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 IMS data stores (if any) can run this Transaction if Transaction Routing is active for this transaction.

The acceptable field values are:

1 All IMS data stores in this IMS Connect system can support this transaction.

2 One datastore only. Use the Prompt function key (F4) to get a list of defined IMS data stores. To define an IMS data store, see “Defining IMS data stores” on page 339.

3 Any datastore defined in a group. Use the Prompt function key (F4) to get a list of groups. To define a group of IMS data stores, see “Defining groups of IMS data stores” on page 344.

4 Any datastore defined in an affinity list. Use the Prompt function key (F4) to get a list of defined affinity lists. “Defining affinity lists” on page 346.
Routing Error processing

This field defines the processing options when an IMS data store 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 IMS data store in the message
2. Reject the transaction

Related concepts:

"Transaction routing" on page 607

Transaction routing is a legacy process by which IMS Connect Extensions selects a destination IMS data store for an incoming IMS request message (IRM) based on the transaction code and destination ID specified by the message.

Related tasks:

"Managing IMS Connect transaction options” on page 277

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.

Related reference:

"Overriding transaction definitions” on page 162

Use the Set Transaction panel to temporarily override IMS Connect Extensions in-memory definition settings for a transaction definition. The transaction definition in the IMS Connect Extensions repository remains unchanged. To access this ISPF panel, select option 6 Transaction on the Set Commands panel.

Transaction take-up from MODBLKS

Use the TAKEUP command on the Transactions panel to quickly define transactions in the IMS Connect Extensions repository using transaction definitions stores in your IMS MODBLKS libraries.

To perform a take-up from IMS MODBLKS, complete the following steps:

1. Open the Transactions panel. See “Defining transactions” on page 349.
2. On the Command line, type TAKEUP and then press Enter.

3. On the Transaction Selection panel, enter the following information in the fields provided:
**MODBLKS library**
The data set containing your IMS MODBLKS.

**Member suffix**
The MODBLKS library member suffix to be used.

**IMS release**
The IMS release.

---

When you are finished, press Enter. A list of transactions defined in the MODBLKS library that are not already defined to IMS Connect Extensions is displayed. It allows you to select transactions to add to IMS Connect Extensions. Transactions are added using options modelled on the nominated existing transaction definition.

4. Select the desired transactions using the following panel options:
   - Use line action S to select a transaction in the list.
   - Use the **Command** line to enter 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 $*$ 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 all line actions.
   - Use the **Model** field to add new transaction definitions based on an existing definition. Use the Prompt function key (F4) to view a list of defined transactions.

---

Figure 181. Specifying MODBLKS options in the Transaction selection panel

---

IBM IMS Connect Extensions for z/OS: User’s Guide
When you are finished, press Enter.

Result

The results of the MODBLK take-up are displayed on the Transactions panel:

![Image of the Transactions panel]

Figure 182. Selecting a transaction in the Transaction Selection panel

Figure 183. The Transactions panel showing the results of a transaction take-up from IMS MODBLKS

Defining system groups

IMS Connect Extensions allows you to group collections of IMS Connect systems. After defining a 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 group in the ISPF dialog, select option 1.8 System Groups from the IMS Connect Extensions primary menu.

The System Groups panel displays a list of groups defined in your IMS Connect Extensions repository. Use the Filter field to list only items that match a filter pattern.
To define a new system group, enter **NEW** on the **Command** line. Enter the 1-8 character name of your group and press Enter. The System Group panel is displayed. In this dialog you will assign one or more systems to the group.

The panel displays the following information. Enter line action / next to a system to insert additional systems and to see a complete list of options:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Changed</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>MF1</td>
<td>My group</td>
<td>2018/07/31 13:56:04</td>
<td>AJL</td>
</tr>
</tbody>
</table>

The panel displays the following information. Enter line action / next to a system to insert additional systems and to see a complete list of options:

**Name**  The name of the group.

**Description**  A description of this group.

**System and Description**

The name of an IMS Connect system defined in your IMS Connect Extensions repository. Use the Prompt function key (F4) to select from a list of defined systems. To define an IMS Connect system, see “Defining IMS Connect systems” on page 320.

To save your changes, press the Exit function key (F3). The new group is displayed in the System Groups panel.

---

**Defining OTMA routing rules**

OTMA routing rules enable you to specify routing behavior for IRM messages. They describe which OTMA routing lists to use to route messages that have a given original IMS data store identifier (IRM_IMSDESTID). To define OTMA routing rules, select option 1.9 **OTMA Routing Rules** from the IMS Connect Extensions primary menu.
The OTMA Routing Rules panel displays a list of rules defined in your IMS Connect Extensions repository. Use the Filter field to list only items that match a filter pattern.

To define a new OTMA routing rule, enter NEW on the Command line. Enter the 1-8 character name of your routing rule and press Enter. The OTMA Routing Rule panel is displayed. In this dialog you will define the routing behavior for messages with a specified DestID.

The panel displays the following information:

**Name**  The name of the OTMA routing rule.

**Description**  A description of this routing rule.

**Apply rule to**  Specifies the IMS Connect system or systems to which this rule will apply.

Choose from the following options:

**Rules is active when:**

1. System . . . . . . . . . +  **Routing Plan** . . . . WEEKLY +
2. Group . . . . . . . . . . . +  **Rules with no plan are always active**
3. All systems

**Condition:**

**Original Datastore**, SY10 (IRM.IMS DestId)
**Additional qualifier**, TRANSACTION +
**List name**, TXLLA +

**Message types**

- **Send-Receive Transactions**
  - **Send-Only requests**
    - **Resume TPIPE requests**
      - **Synchronous call-out resume TPIPE requests**
      - **Synchronous call-out responses**

**Figure 186. The OTMA Routing Rules panel in the IMS Connect Extensions ISPF dialog**

**Figure 187. The OTMA Routing Rule panel in the IMS Connect Extensions ISPF dialog**
1 System
Apply the rule to a single system. Use the the Prompt function key (F4) to select from a list of defined systems. To define an IMS Connect system, see “Defining IMS Connect systems” on page 320.

2 Group
Apply the rule to a group of systems. Use the the Prompt function key (F4) to select from a list of defined groups. To define a group, see “Defining system groups” on page 355.

3 All systems
Apply the rule to all IMS Connect systems.

Routing Plan
Optional. Assign the routing rule to a routing plan. A routing plan defines a group of routing rules that are intended to operate at the same time.

Use the Prompt function key (F4) to select from a list of defined routing plans. To define a routing plan, see “Defining routing plans” on page 370.

Note: A routing rule that is not assigned to a routing plan will always be in effect, except when it is overridden by a routing rule that contains a more specific message matching condition.

Condition
Specifies the conditions that must be met by the incoming IRM message for the rule to take effect. Select from the following options:

Original Datastore (IRM_IMSDestId)
A value that is to be matched against the string that appears in the IRM_IMSDESTID field on incoming IRM messages in order to select this OTMA routing rule for the incoming message. For more information on IRM structures for IMS Connect client messages, see Format of user portion of IRM for HWSSMPL0, HWSSMPL1, and user-written message exit routines in the IMS user documentation.

Important: The value supplied by the client in the IRM_IMSDESTID message field does not have to be the name of an IMS data store defined in the IMS Connect configuration member. In IMS Connect Extensions, the value it receives is simply used to identify matching messages to which this rule applies. As such, there are several options available when specifying a value on this panel:

• The value can be a fully qualified IMS data store name (for example, IMSA, which will only be applicable if it is a direct match to the IRM_IMSDESTID field.

• The value can be a prefix string with a trailing asterisk (for example, IMS*) which indicates that the IRM_IMSDESTID field need only match the prefix string component of the name in order to be selected.

• The value can be a single asterisk (*) which matches any IRM_IMSDESTID field value.

When multiple rules exist that contain overlapping IRM_IMSDESTID values, more specific name values in the Original Datastore field take precedence over less specific values.
such as masked name values. Only the first OTMA routing rule found to match the IRM_IMSDESTID field for a given incoming message is activated.

**Additional qualifier**

Specifies an optional condition that qualifies this rule. Use the Prompt function key (F4) to select a qualifier type. The following options are available:

- **NONE**  
  The routing rule has no additional qualifiers. The rule will match on the IRM_IMSDESTID only. The rule will take precedence over any other rules that match on the same IRM_IMSDESTID and also specify an additional qualifier.

- **TRANSACTION**  
  The routing rule will only take effect for the named transactions that match the IRM_IMSDESTID. Use the Prompt function key (F4) on the List name field to select a transaction list. To define a transaction list, see “Defining transaction lists” on page 369.

  **Note:** If you use the TRANSACTION qualifier, you may also wish to define an additional unqualified rule (Additional qualifier = NONE) for the same IRM_IMSDESTID to act as a “master” rule. If you specify a qualifying rule without a matching master rule in the repository, an implied routing rule will be generated internally at run time. Use the message log to review the set of routing rules that are in effect. Messages “CEX5091I” on page 501 and “CEX5191I” on page 507 list the routing rules that are in effect. See “Browsing the message log for an IMS Connect” on page 164.

**Message types**

Route messages to IMS data stores based on the message type.

When an incoming message is matched to the name in the Original Datastore field, IMS Connect Extensions selects a routing list based on the message type.

For each of the listed message types, choose from the following options.

- **Y**  
  Enable routing for this message type. Requests that match this routing rule are routed to a list of IMS data stores specified in the Target field. Use the optional Fallback field to specify a list of fallback IMS data stores in the event that the IMS data stores listed in the Target field are unavailable. Use the Prompt function key (F4) to select a list of defined OTMA routing lists. To define an OTMA routing list, see “Defining OTMA routing lists” on page 361.

- **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 OTMA 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. Any values in the Target and Fallback fields are ignored.

The following message types are supported:

**Send-Receive Transactions**
Standard IMS Connect transactions.

**Tip:** This message type can also be used to route IMS commands if CEXROUTE IMSCMDROUTE=YES is specified in the control input data set. See Chapter 26, “Control input data set options,” on page 561.

**Send-Only requests**
IMS Connect Send Only requests.

**Resume TPIPE requests**
IMS Connect 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 requests with Resume TPIPE requests, use the same collection of target and fallback IMS data stores for both message types and ensure all data stores 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 call-out resume TPIPE requests**
Routing for Resume TPIPE (RTPipe) messages marked as synchronous callout requests.

**Synchronous call-out responses**
Routing for synchronous callout responses.

**Tip:** Use IRM field IRM_F4 to help you determine the types of messages in your environment. To view this part of the IMS request message (IRM) use IBM IMS Problem Investigator for z/OS to browse the IMS Connect Extensions journal. For more information on IRM, see Format of user portion of IRM for HWSSMPL0, HWSSMPL1, and user-written message exit routines in the IMS user documentation. See also “Reporting and analysis with IMS Problem Investigator” on page 73.

To save the new routing rule, press the Exit function key (F3). The new routing rule is displayed 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. See “Defining IMS Connect systems” on page 320.

**Related concepts:**
Chapter 12, “OTMA workload routing in IMS Connect,” on page 245

OTMA rules-based routing in IMS Connect Extensions allows you to route and distribute OTMA workload received by IMS Connect between available IMS data stores.
Defining OTMA routing lists

The OTMA Routing Lists panel enables you to define collections of IMS data stores to use as candidates in an OTMA routing rule. To access this ISPF panel, select option 1.10 **OTMA Routing Lists** from the IMS Connect Extensions primary menu.

The OTMA Routing Lists panel displays a list of OTMA routing lists defined in your IMS Connect Extensions repository. Use the **Filter** field to list only items that match a filter pattern.

To define a new OTMA routing list, enter **NEW** on the **Command** line. Enter the 1-8 character name of your routing list and press Enter. The OTMA Routing List panel is displayed. In this dialog you will assign one or more IMS data stores to the routing list.

The panel displays the following information. Enter line action / next to an IMS data store to insert additional IMS data stores and to see a complete list of options:

- **Name**
- **Description**
- **Datastore and Description**

You can see a complete list of options by entering line action / next to an IMS data store. The panel displays the following information:

**File Menu Settings Help**

**Command ==> NEW**

OTMA Routing Lists Member NEPAYDS saved

**Scroll ==> PAGE**

Filter . . . ______ (Blank or pattern)

Enter "/" to select action

```
/  Name Description       ---- Changed ----   ID
NEPAYDS  NE Region Payroll Datastores 2018/08/01 11:22:25  VIP
```

**Figure 188. The OTMA Routing Lists panel in the IMS Connect Extensions ISPF dialog**

To define a new OTMA routing list, enter **NEW** on the **Command** line. Enter the 1-8 character name of your routing list and press Enter. The OTMA Routing List panel is displayed. In this dialog you will assign one or more IMS data stores to the routing list.

```
EDIT 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 189. The OTMA Routing List panel in the IMS Connect Extensions ISPF dialog**

The panel displays the following information. Enter line action / next to an IMS data store to insert additional IMS data stores and to see a complete list of options:

**Name**  The name of the OTMA routing list.

**Description**  A description of this OTMA routing list.

**Datastore and Description**  The name of an IMS data store defined in your IMS Connect Extensions repository. Use the Prompt function key (F4) to select from a list of defined IMS data stores. To define an IMS data store, see "Defining IMS data stores" on page 339.
To save your changes, press the Exit function key (F3). The new routing list is displayed in the OTMA Routing Lists panel.

**Related reference:**

“Defining OTMA routing rules” on page 356

OTMA routing rules enable you to specify routing behavior for IRM messages. They describe which OTMA routing lists to use to route messages that have a given original IMS data store identifier (IRM.IMSDESTID). To define OTMA routing rules, select option 1.9 **OTMA Routing Rules** from the IMS Connect Extensions primary menu.

### Defining ODBM targets

The ODBM Targets panel allows you to define ODBM target definitions for use in ODBM rules-based routing. ODBM target definitions inform IMS Connect Extensions about a routing pathway between an ODBM instance and an IMS data store referenced by its alias name. To access this ISPF panel, select option 1.11 **ODBM Targets** from the IMS Connect Extensions primary menu.

The ODBM Targets panel is displayed with a list of all ODBM targets defined in your IMS Connect Extensions repository. Use the **Filter** field to list only items that match a filter pattern.

![Figure 190. The ODBM Targets panel in the IMS Connect Extensions ISPF dialog](image)

To define a new ODBM target, enter **NEW** on the **Command** line. Enter the 1-8 character name to identify your ODBM target definition and press Enter. The ODBM Target panel is displayed.

![Figure 191. The ODBM Target panel in the IMS Connect Extensions ISPF dialog](image)

The panel displays the following information:

**Name**  The name of the ODBM target definition.
Description
A description of this ODBM target definition.

ODBM Name
The ODBMID of an ODBM instance.

ODBM creates an eight-character ODBMID that identifies the instance of ODBM within the IMSplex. The ODBMID is the name defined in the CSLDIxxx member of the IMS PROCLIB data set followed by the characters OD and any blank spaces that ODBM needs to add to make the ODBMID eight characters in length. For more information, see CSLDIxxx member of the IMS PROCLIB data set in the IMS user documentation.

Tip: To see a list of ODBM instances, use the Status Monitor. See “Systems Overview” on page 126.

IMS Alias
An optional 4-character alias name for an IMS data store.

Application programs must use an alias name to access IMS data stores and do not need to know the actual IMSID of the IMS data store. Alias names for IMS data stores are defined in the CSLDCxxx member of the IMS PROCLIB data set. For more information, see CSLDCxxx member of the IMS PROCLIB data set in the IMS user documentation.

Tip: To see a list of aliases, use the Status Monitor. See “Systems Overview” on page 126.

If you do not specify an alias, IMS Connect Extensions selects an alias associated with the ODBM specified in the ODBM Name field. In this case, workload balancing is not used and all aliases associated with the ODBM are considered to have equal capacity weight ratings for the request. This can be useful when configuring a fallback ODBM routing list that will route requests to any available IMS data store alias if the primary aliases are unavailable.

Capacity weight rating
This field allows you to define the capacity weight rating (CWR) for this ODBM target. The rating is used by workload balancing to reroute incoming messages to alternate ODBM targets.

Acceptable field values are numbers in the range 1 - 100 and 0. A value of zero 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 will apportion incoming requests equally between 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.

Related reference:
Routing lists enable you to define collections of ODBM targets to use as candidates in an ODBM routing rule. To define an ODBM routing list, use the ODBM Routing Lists panel. To access this ISPF panel, select option 1.13 ODBM Routing Lists from the IMS Connect Extensions primary menu.

### Defining ODBM routing rules

ODBDM 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. To define an ODBM routing rule, use the ODBM Routing Rules panel. To access this ISPF panel, select option 1.12 ODBM Routing Rules from the IMS Connect Extensions primary menu.

The ODBM Routing Rules panel is displayed with a list of ODBM routing rules defined in your IMS Connect Extensions repository. Use the Filter field to list only items that match a filter pattern.

![ODBDM Routing Rules Panel](image)

**Figure 192. The ODBM Routing Rules panel in the IMS Connect Extensions ISPF dialog**

To define a new ODBM routing rule, enter **NEW** on the Command line. Enter the 1-8 character name of your routing rule and press Enter. The ODBM Routing Rule panel is displayed. In this dialog you will define the routing behavior for DRDA requests with a specified input alias.
The panel displays the following information:

**Name**  The name of the ODBM routing rule.

**Description**  A description of this routing rule.

**Apply rule to**

Specifies the IMS Connect system or systems to which this rule will apply.

Choose from the following options:

1. **System**
   Apply the rule to a single system. Use the the Prompt function key (F4) to select from a list of defined systems. To define an IMS Connect system, see “Defining IMS Connect systems” on page 320.

2. **Group**
   Apply the rule to a group of systems. Use the the Prompt function key (F4) to select from a list of defined groups. To define a group, see “Defining system groups” on page 355.

3. **All systems**
   Apply the rule to all IMS Connect systems.

**Routing Plan**

Optional. Assign the routing rule to a routing plan. A routing plan defines a group of routing rules that are intended to operate at the same time.

Use the Prompt function key (F4) to select from a list of defined routing plans. To define a routing plan, see “Defining routing plans” on page 370.

**Note:** A routing rule that is not assigned to a routing plan will always be in effect, except when it is overridden by a routing rule that contains a more specific message matching condition.

**Condition**

Specifies the conditions that must be met by the incoming DRDA request for the rule to take effect. Select from the following options:

**Input Alias**

The alias name supplied by the client in the DRDA request.
PSB name list

Optional. A list of IMS program specification block (PSB) names that may be in the DRDA request. Use the Prompt function key (F4) to select from a list of defined PSB name lists. To define a PSB name list, see "Defining PSB name lists" on page 368.

If 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.

Tip: The alias name and PSB name are supplied in the DRDA request message in the distributed data management (DDM) architecture ACCRDB command (X’2001’) in the RDBNAM parameter. For more information, refer to the topic ACCRDB command (X’2001’) in the IMS documentation. To see what alias and PSB name is being used by your client applications, use IBM IMS Problem Investigator for z/OS to format and view IMS Connect Extensions journal record X’5B’ DRDA Request. See "Reporting and analysis with IMS Problem Investigator" on page 73.

Request types

Specify one of the following line actions next to the DRDA Requests field:

Y  Enable DRDA message routing. Requests are routed to a list of ODBM targets specified in the Target field. Use the optional Fallback field to specify a list of fallback ODBM targets in the event that the ODBM targets listed in the Target field are unavailable. Use the Prompt function key (F4) to select a list of defined OTMA routing lists. To define an OTMA routing list, see "Defining OTMA routing lists" on page 361.

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 ODBM 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. Any values in the Target and Fallback fields are ignored.

To save your new routing rule, press the Exit function key (F3). The new routing rule is displayed 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. See "Defining IMS Connect systems" on page 320.

Related concepts:

Chapter 14, “ODBM workload routing in IMS Connect,” on page 279
ODBM rules-based routing in IMS Connect Extensions provides advanced routing in IMS Connect for TCP/IP client requests that use Distributed Relational Database Architecture (DRDA) to access IMS data via IMS Open Database.
Defining ODBM routing lists

Routing lists enable you to define collections of ODBM targets to use as candidates in an ODBM routing rule. To define an ODBM routing list, use the ODBM Routing Lists panel. To access this ISPF panel, select option 1.13 ODBM Routing Lists from the IMS Connect Extensions primary menu.

The ODBM Routing Lists panel is displayed with a list of ODBM routing lists defined in the IMS Connect Extensions repository. Use the Filter field to list only items that match a filter pattern.

To define a new ODBM routing list, enter NEW on the Command line. Enter the 1-8 character name of your routing list and press Enter. The ODBM Routing List panel is displayed. In this dialog you will assign one or more ODBM targets to the routing list.

The panel displays the following information. Enter line action / next to an ODBM target to insert additional ODBM targets and to see a complete list of options:

- **Name**  The name of the ODBM routing list.
- **Description**  A description of this routing list.
- **Target and Description**  The name of an ODBM target defined in your IMS Connect Extensions repository. Use the Prompt function key (F4) to get a list of defined targets. To define an ODBM target, see “Defining ODBM targets” on page 362.
To save your definition, press the Exit function key (F3). The new routing list is displayed in the ODBM Routing Lists panel.

**Related reference:**
“Defining ODBM routing rules” on page 364
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. To define an ODBM routing rule, use the ODBM Routing Rules panel. To access this ISPF panel, select option 1.12 **ODBM Routing Rules** from the IMS Connect Extensions primary menu.

### Defining PSB name lists

PSB name lists enable you to define collections of PSB names to which an ODBM routing rule can be applied. To define an PSB name list, use the PSB Name Lists panel. To access this ISPF panel, select option 1.14 **PSB Name Lists** from the IMS Connect Extensions primary menu.

The PSB Name List panel displays a list of PSB name lists defined in your IMS Connect Extensions repository. Use the **Filter** field to list only items that match a filter pattern.

![Figure 196. The PSB Name Lists panel in the IMS Connect Extensions ISPF dialog](image)

To define a new PSB name list, enter **NEW** on the **Command** line. Enter the 1-8 character name of your list and press Enter. The PSB Name List panel is displayed. In this dialog you will assign one or more PSB names to the PSB name list.

![Figure 197. The PSB Name List panel in the IMS Connect Extensions ISPF dialog](image)

The panel displays the following information. Enter line action / next to a PSB name to insert additional names and to see a complete list of options:

**Name** The name of the PSB name list.
Description
A description of this PSB name list.

PSB name and Description
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.

To save your definition, press the Exit function key (F3). The new PSB name list is displayed in the PSB Name Lists panel.

Related reference:
“Defining ODBM routing rules” on page 364
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. To define an ODBM routing rule, use the ODBM Routing Rules panel. To access this ISPF panel, select option 1.12 ODBM Routing Rules from the IMS Connect Extensions primary menu.

Defining transaction lists
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. To define a transaction list, use the Transaction Lists panel. To access this ISPF panel, select option 1.15 Transaction Lists from the IMS Connect Extensions primary menu.

The Transaction Lists panel is displayed with a list of transaction lists defined in the IMS Connect Extensions repository. Use the Filter field to list only items that match a filter pattern.

Figure 198. The Transaction Lists panel in the IMS Connect Extensions ISPF dialog
To define a new transaction list, enter NEW on the Command line. Enter the 1-8 character name of your list and press Enter. The Transaction Name List panel is displayed.
The panel displays the following information. Enter line action / next to a transaction or pattern to insert additional items and to see a complete list of options:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Transaction Codes</th>
</tr>
</thead>
<tbody>
<tr>
<td>TLIST2</td>
<td>List of special handling</td>
<td>AB*, RESET, DFLT0</td>
</tr>
</tbody>
</table>

Enter "/" to select action

To save the list, press the Exit function key (F3). The new transaction list appears in the Transaction Lists panel.

**Related reference:**

“Defining OTMA routing rules” on page 356

OTMA routing rules enable you to specify routing behavior for IRM messages. They describe which OTMA routing lists to use to route messages that have a given original IMS data store identifier (IRM_IMSDESTID). To define OTMA routing rules, select option 1.9 OTMA Routing Rules from the IMS Connect Extensions primary menu.

---

**Defining routing plans**

Routing plans are used in IMS Connect Extensions to group routing rules that are intended to operate at the same time. To define a routing plan, use the Routing Plans panel. To access this ISPF panel, select option 1.16 Routing Plans from the IMS Connect Extensions primary menu.

The Routing Plans panel is displayed with a list of routing plans defined in the IMS Connect Extensions repository. Use the Filter field to list only items that match a filter pattern.
To define a new routing plan, enter NEW on the Command line. Enter the 1-8 character name of your routing plan and press Enter. The Routing Plan panel is displayed.

The panel displays the following information:

- **Name**: The name of the routing plan.
- **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.

**Notes:**
- 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.
- To activate a routing plan, use the Set Commands dialog or SET IMS Connect Extensions host command for REXX and then refresh the routing rules.

**Related tasks:**
- “Activating OTMA routing rules in IMS Connect” on page 259
- “Activating ODBM routing rules in IMS Connect” on page 294

**Related reference:**
- “Defining OTMA routing rules” on page 356

OTMA routing rules enable you to specify routing behavior for IRM messages. They describe which OTMA routing lists to use to route messages that have a given original IMS data store identifier (IRM_IMSDESTID). To define OTMA routing rules, select option 1.9 OTMA Routing Rules from the IMS Connect Extensions primary menu.
ODOBM 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. To define an ODBM routing rule, use the ODBM Routing Rules panel. To access this ISPF panel, select option 1.12 ODBM Routing Rules from the IMS Connect Extensions primary menu.

Defining IP address rules

IP address rules allow you to specify what preconditioning is to take place on workloads coming from specific IP addresses. Use IP address rules to treat messages coming from an IP address as a trusted user or to assign a specific user ID to the request. To define an IP address rule, use the IP Address Rules panel. To access this ISPF panel, select option 1.17 IP Address Rules from the IMS Connect Extensions primary menu.

The IP Address Rules panel is displayed with a list of IP address rules defined in your IMS Connect Extensions repository. Use the Filter field to list only items that match a filter pattern.

```
File Menu Settings Help
Command ==> IP Address Rules Row 1 from 2 Scroll ==> PAGE
Filter ... (Blank or pattern)
Enter "/*" to select action

/ Name  IP Address Userid Type Definition
  SRVR001 193.23.88.122 VIP001 SYSTEM HWSOPGS1
  SRVR002 193.23.89.* VIP002 SYSTEM HWSOPGS1
```

Figure 202. The IP Address Rules panel in the IMS Connect Extensions ISPF dialog

To define a new IP address rule, enter NEW on the Command line. Enter the 1-8 character name of your IP address rule and press Enter. The IP Address Rule panel is displayed. In this dialog you will define the preconditioning that is to take place for a given IP address.
The panel displays the following information:

**Name**  
The name of the IP address rule.

**Description**  
A description for this IP address rule.

**Apply rule to**  
Specifies the IMS Connect system or systems to which this rule will apply. Choose from the following options:

1. **System**  
   Apply the rule to a single system. Use the the Prompt function key (F4) to select from a list of defined systems. To define an IMS Connect system, see “Defining IMS Connect systems” on page 320.

2. **Group**  
   Apply the rule to a group of systems. Use the the Prompt function key (F4) to select from a list of defined groups. To define a group, see “Defining system groups” on page 355.

3. **All systems**  
   Apply the rule to all IMS Connect systems.

**Condition**  
Specifies the conditions that must be met for the rule to take effect.

**IP Address**  
Apply the rule to workloads originating from the specified IPv4 or IPv6 address (for example, 192.23.88.1 for an IPv4 address or 2001:0db8:85a3:0000:0000:8a2e:0370:7334 for an IPv6 address) or from a range of IP addresses by using a trailing wildcard (*). The wildcard must appear at the end of the IP address and immediately after the final node separator character in the address (for example, 192.55.* for an IPv4 address or ::0:::0:ffff:c017:* for an IPv6 address). A wildcard must not be mixed with an IPv6 address containing two consecutive colons (:). IPv6 addresses may not contain IPv4 dot-decimal notation.

**Options**  
The following preconditioning options may be applied:
Override User id

Override the user ID defined in the request message with the specified user ID. Workloads will be processed using the specified user ID, regardless of what is contained in the request.

Only override blank userids

When this option is selected, the user ID specified in the Override User id field is only used when no user ID is supplied in the request message.

Trust user id

Treat all messages from this IP address or IP address range as trusted users in IMS Connect. SAF password verification in IMS Connect is not performed.

To save your new IP address rule, press the Exit function key (F3). The new IP address rule is displayed in the IP Address Rules panel.

Before IP address rules can take effect, you must select the Activate IP Address Rules option in the system definition. See “Defining IMS Connect systems” on page 320.

Related tasks:

“Creating workload rules for specific IP addresses” on page 305

IMS Connect Extensions allows you to create IP address rules for an IMS Connect system that can be used to treat messages coming from specific IP addresses as trusted users or to override the user ID supplied in a request message with another of your choosing.
Chapter 18. 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 tasks:**

“Creating an initial configuration for IMS Connect Extensions” on page 32

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

**Definition takeup utility**

The definition takeup utility produces commands that can be used by the definition maintenance utility to populate the IMS Connect Extensions repository. It does this by reading the specified IMS Connect configuration member to create commands that can be used to create IMS Connect system definitions, IMS data store definitions, and IMS Connect user exit definitions.
Tip: Use the definition takeup utility to create an initial configuration. To create additional supporting definitions such as routing lists, routing rules, and ODBM targets, use the ISPF dialog. See Chapter 17, “Administering IMS Connect Extensions definitions with the ISPF dialog,” on page 315.

The definition 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**

The following JCL generates ADD commands from a configuration member that can be used by the definition maintenance utility:

```plaintext
//STEP EXEC PGM=FUNEXEC,PARM='CEXDFDST'
//STEPLIB DD DISP=SHR,DSN=funpre.SFUNLINK
//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=cepre.SCEXSAMP(CEXARCH1)

TAKEUP CONFIG=CONFPS
MEMBER=HWSCFG1, CEX_PORT=1235,

more batch commands
/*

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 definition takeup jobs because the utility will identify duplicate IMS data stores 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 definition 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 Common Services Library (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.

Syntax

```
TAKEUP CONFIG=ddname, MEMBER=hwscfg, HOST=localhost
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 takeover 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.

**Syntax**

```
SET CEXOUT=ddname
OPTION=(WARNING=ABORT)
OPTION=(WARNING=IGNORE)
```

**Parameters**

**CEXOUT=ddname**

Set the output of the command to the data set specified by ddname.

**OPTION**

Set global options.

**WARNING**

Determines whether the utility continues processing when a warning-level control card error has been detected. Choose from the following options:

**ABORT**

Execution of the utility ends. This is the default.

**IGNORE**

The utility continues to execute.

**Example**

```
SET OPTION=(WARNING=IGNORE)
```
Definition maintenance utility

The definition maintenance utility allows you to add definitions to the IMS Connect Extensions repository, or remove definitions that are no longer required. Use the definition maintenance utility with the commands produced by the definition takeup utility to create an initial set of definitions for your IMS Connect Extensions repository.

The definition 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

The following example JCL adds definitions to an IMS Connect Extensions repository:

```
//STEP EXEC PGM=FUNEXEC,PARM='CEXDFDSI'
//STEPLIB DD DISP=SHR,DSN=funpre.SFUNLINK
//DD DISP=SHR,DSN=cexpre.SCEXLINK
//REPOSITORY 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 Common Services Library (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

The definition maintenance utility ADD command adds an IMS Connect Extensions definition to an IMS Connect Extensions repository, optionally replacing an existing definition.

Syntax

```
ADD DEFN=( *,name )
```

Parameters for definition type

```
,REPLACE=NO
```

```
,REPLACE=YES
```

```
,AFL
```

```
,APP
```

```
,DS
```

```
,DSG
```

```
,EX
```

```
,GRP
```

```
,HWS
```

```
,IPR
```

```
,ORL
```

```
,ORR
```

```
,ORT
```

```
,PLN
```

```
,PSB
```

```
,RTG
```

```
,RTL
```

```
,_TXL
```

```
,TAXN
```

```
,DESC='description'
```

```
```

```
```

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 381
- “Application (APP) ADD command” on page 382
- “Datastore (DS) ADD command” on page 383
- “Datastore group (DSG) ADD command” on page 385
- “User exit (EX) ADD command” on page 386
Affinity list (AFL) ADD command

The affinity list (AFL) ADD command adds an affinity list to an IMS Connect Extensions repository. An affinity list is a list of data stores that you can use to set potential candidates for message routing. You only need this definition if you use routing.

Syntax

```
ADD DEFN=(AFL,name),DESC='description',REPLACE=NO,REPLACE=YES
```

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.
NO  All ADD commands directed at the repository fail.

\textbf{DSLIST}=\textit{(datastores,...)}

A comma-delimited list of datastores.

\textbf{Application (APP) ADD command}

The application (APP) ADD command adds an application to an IMS Connect Extensions 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.

\textbf{Syntax}

\begin{verbatim}
ADD DEFN=(APP, name) [DESC='description'],
  REPLACE=NO, REPLACE=YES
, TRANSACTION_ROUTING=INACTIVE
, TRANSACTION_ROUTING=ACTIVE,
  ROUTING_RULE=(\textit{DS}, name)
\end{verbatim}

\textbf{Notes:}

1. You must specify a routing rule if you activate routing.

\textbf{Parameters}

\textbf{DEFN=(APP, name)}

Define an application called \textit{name}. You can use any 8-character name for your applications.

\textbf{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.

\textbf{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.

\textbf{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.

\textbf{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 IMS data stores. IMS Connect Extensions will only route messages to those data stores that are candidate data stores 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 data stores

DS,name
   A data store called “name”

DSG,name
   A data store 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 data store 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. This is the default.

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 datastore (DS) ADD command adds an IMS data store definition to an IMS Connect Extensions repository.

Syntax

```
ADD DEFN=(DS,name) [ ,DESC=description ] [ ,REPLACE=NO | YES ] [ ,DSG=group_name ]
               [ ,PACING=(INACTIVE, WARN=0, REJECT=0) ]
               [ ,PACING=(ACTIVE, WARN=n, REJECT=n) ]
               [ ,TRANSACTION_ROUTING=INACTIVE | ACTIVE ]
               [ ,ROUTING_RULE=(ALL, DS,name, DSG,name, AFL,name) ]
```

(1)
Notes:
1. You must specify ROUTING_RULE if you activate transaction routing.

**Parameters**

**DEFN**=(*DS*,*name*)
Define a data store 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.

**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 data store 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 data store. 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_RULE**=(candidate_datastores)
This parameter specifies a transaction routing rule:

A transaction routing rule consists of a list of candidate IMS data stores. IMS Connect Extensions will only route messages to those data stores that are candidate data stores 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 data stores

**DS, name**
A data store called “name”

**DSG, name**
A data store 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 615 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 rating for this data store. 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 data stores, that IMS Connect Extensions will route messages to this data store. 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 data stores 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 rating of 8, than to DS1, with a capacity weight of 4. No transactions will be routed to DS3.

**Datastore group (DSG) ADD command**
The datastore group (DSG) **ADD** command adds a datastore group to an IMS Connect Extensions 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.

**Syntax**

```
ADD DEFN=(DSG, name), DESC='description', REPLACE=NO
```

---

Chapter 18. Definition utilities 385
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 user exit (EX) ADD command adds a user exit to an IMS Connect Extensions
repository. You must define all exits for which you want to use IMS Connect
Extensions functions.

Syntax

```
ADD-DEFN=(EX,name)

,DESC='description'

,REPLACE=NO | YES

,EXIT=ACTIVE | INACTIVE

,IRM_OFFSETS=DEFAULT | CUSTOM

,PREFIX_LENGTH=4 | 2

,EXTED_RSM=NO | YES

,MSID_STRING1=EBDIC | ASCII

,MSID_STRING2=EBDIC | ASCII

,MSG_TRANSLATION=(INACTIVE)
```

IBM IMS Connect Extensions for z/OS: User's Guide
Notes:
1. You must specify a 4-byte length prefix if you use an extended RSM.

Custom IRM offset values:

- USER_ID=0
- USER_ID=decimal_offset
- PASSWORD=0
- PASSWORD=decimal_offset
- GROUP=0
- GROUP=decimal_offset
- APPL=0
- APPL=decimal_offset

Message translation options:

- CLIENT_CCSID=decimal_ccsid
- SERVER_CCSID=decimal_ccsid
- WHEN_ERROR=DISABLE
- WHEN_ERROR=CONTINUE

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 system group (GRP) ADD command adds a system group to an IMS Connect Extensions 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.

Syntax

```
ADD DEFN=(GRP,name) ,DESC='description',REPLACE=NO
```

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 IMS Connect system (HWS) ADD command adds an IMS Connect system definition to an IMS Connect Extensions repository.

Syntax

```
ADD DEFN=(HWS,name) ,DESC='description',REPLACE=NO ,REPLACE=YES ,CEX_PORT=port_number
```

Chapter 18. Definition utilities 389
Active data set configuration:

- **DSN_template**
  - **NUM_ACTIVES**=3
  - **ARCHIVE**=YES
  - **WHEN_FULL**=REUSE
- **ARCHIVE**=NO
  - **WHEN_FULL**=WAIT
  - **WHEN_FULL**=DISCARD
  - **WHEN_FULL**=END
  - **MAN_CLASS**=class

- **STATS**=ACTIVE
- **STATS**=INACTIVE

- **PACING**=(ACTIVE, **COUNT**=3, **WARN**=0, **REJECT**=0)
  - **PACING**=(ACTIVE, **COUNT**=n, **WARN**=n, **REJECT**=n)

- **SESSION_MESSAGE_LIMIT**=ACTIVE, **THRESHOLD**=(value)
Archive data set configuration:

```
archive dsn=template
max_active=3
max_volumes=1
max_size=nnnnn
archive_cleanup=7
archive_cleanup=num_of_days
```

Commands configuration:

```
inactive
  control_access=no
active
  control_access=(no
    security_applid=applid
    gen_passticket=yes)
```

Security configuration:

```
inactive
  acee_cache=inactive
  age_interval=60
active
  acee_cache=active
  age_interval=nnnn
```

1. Act_val=inactive
2. Act_val=active,security_class=SAF_class,val_type=
   IMSCONNECT
   IMSCONNECT_IPADDRESS_PORT
   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
```
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. An active data set definition is not required if event collection is set to INACTIVE. See “Active data set configuration” on page 394.

**ARCHIVE_DS=(journal_name_template)**
Specifies the data set name prefix and other attributes for archive journal data sets. An archive data set definition is not required if the active data set definition sets archiving to INACTIVE. See “Archive data set configuration” on page 396.

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. See “SECURITY option” on page 568.

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. See “Defining IMS Connect systems” on page 320.

**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 IMS data stores or data store 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.

IP_ADDRESS_RULES=ACTIVE | INACTIVE
Specify whether to activate IP address rules. IP address rules allow you to specify what preconditioning is to take place on workloads coming from specific IP addresses.

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 Pnn, 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 585.

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 11, “Command access in IMS Connect Extensions,” on page 235 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.

**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.

**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.

**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.
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.

**IP address rule (IPR) ADD command**

The IP address rule (IPR) **ADD** command adds an IP address rule to an IMS Connect Extensions repository.

**Purpose**

IP address rules allow you to specify what preconditioning is to take place on workloads coming from specific IP addresses. Use IP address rules to treat messages coming from an IP address as a trusted user or to assign a specific user ID to the request.

**Syntax**

```
ADD DEFN=(IPR, name), DESC='description', REPLACE=NO, REPLACE=YES
```

```
,APPLY_RULE_TO=(ALL), CONDITION=(IP_ADDRESS=ip_address)
```

```
,OVERWRITE_UID=(INACTIVE)
```

```
,OVERWRITE_UID=(ACTIVE, USERID=user_id)
```

```
,TRUST_UID=INACTIVE
```

```
,TRUST_UID=ACTIVE
```

**Parameters**

**DEFN=(IPR, 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”.

IP_ADDRESS

Apply the rule to workloads originating from the specified IP address or from a range of IP addresses using a trailing wildcard (*). Both Internet Protocol version 4 (IPv4) and version 6 (IPv6) are supported.

OVERWRITE_UID

When set to ACTIVE, override the user ID defined in the request message with the value specified by the USERID=user_id parameter. Workloads will be processed using the specified user ID, regardless of what is contained in the request.

If BLANK_ONLY=ACTIVE is specified, the user ID specified in the USERID parameter is only used when no user ID is supplied in the request message.

TRUST_UID

When set to ACTIVE, treat all messages from this IP address or IP address range as trusted users in IMS Connect. RACF checking in IMS Connect is not performed.

ODBM routing list (ORL) ADD command

The ODBM routing list (ORL) ADD command adds an ODBM routing list to an IMS Connect Extensions repository.

Syntax

```
ADD DEFN=(ORL,name),DESC='description',REPLACE=NO
```

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 ODBM routing rule (ORR) ADD command adds an ODBM routing rule to a IMS Connect Extensions 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.

Syntax

```
ADD DEFN=(ORR,name),DESC='description',REPLACE=NO

,APPLY_RULE_TO=(ALL)

,CONDITION=(

,INPUT_ALIAS=alias,PSBNAME_LIST=PSB_namelist)

,ORDA_REQUEST=(Routing options)

,REPLACE=YES

REPLACE=NO | YES
```

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.
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 a routing plan will always be in effect,
   except when it is overridden by a routing rule that contains a more specific
   message matching condition.

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.”

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 an ODBM routing list of ODBM targets to use as candidates for
   routing DRDA requests.

FALLBACK
   Specifies an ODBM 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 ODBM target (ORT) ADD command adds an ODBM target to an IMS Connect Extensions 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.

**Syntax**

```
ADD DEFN=(ORT, name), DESC='description', REPLACE=NO
```

```
ADD DEFN=(ORT, name), DESC='description', REPLACE=YES
```

```
ADD DEFN=(ORT, name), TARGET_ODBM_NAME=(ODBM_name), TARGET_IMS_ALIAS=(IMS_alias)
```

```
ADD DEFN=(ORT, name), WEIGHTING=1
```

```
ADD DEFN=(ORT, name), WEIGHTING=capacity_weight
```

```
ADD DEFN=(ORT, name), 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.

**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 PSB name list (PSB) **ADD** command adds a PSB name list to an IMS Connect Extensions 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.

**Syntax**

```
ADD DEFN=(PSB, name) , DESC='description' , REPLACE=NO
```

**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 routing plan (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.

Syntax

```
ADD DEFN=(PLN,name) [DESC='description'] [REPLACE=NO | 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 OTMA routing rule (RTG) ADD command adds an OTMA routing rule to an IMS Connect Extensions repository.

Purpose

A routing rule allows you to specify rules-based routing behavior for IRM messages. It describes which IMS data stores 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 IMS data stores for a specified list of transaction codes.

Syntax

```
ADD DEFN=(RTG,name) [DESC='description'] [REPLACE=NO | YES] [APPLY_RULE_TO=(ALL) | (SYSTEM,system) | (GROUP,group) | PLAN_NAME=plan_name]
```

Chapter 18. Definition utilities  405
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=route_plan
    Assigns this routing rule to a routing plan.

        A routing rule that is not assigned to a routing plan will always be in effect, except when it is overridden by a routing rule that contains a more specific message matching condition.

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 an IMS data store defined in the IMS Connect configuration member. It is simply used as a string to identify matching messages to which this rule applies.

A destid value can be fully qualified, which means that it will only be applicable if it is a direct match to the IRM_IMSDestID field. However, wildcard masking is also supported for the destid value, where a prefix string with a trailing asterisk (*) indicates that the IRM_IMSDestID field need only match the prefix string component of the destid value in order to be selected.

An asterisk-only value is also supported and is considered to match any IRM_IMSDestID field value.

More specific destid values take precedence over less specific, masked values and only the first OTMA routing rule (RTG) found to match the IRM_IMSDestID field for a given incoming message will apply.

Qualifier is an optional condition that qualifies this rule. If you specify a qualifying rule without a matching master rule in the repository, 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 408.

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 408.

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 408.

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” on page 408.

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” on page 408.
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 an OTMA routing list of IMS data stores to use as candidates for routing messages of this type.

**FALLBACK**
Specifies an OTMA routing list of IMS data stores to use as candidates if none of the data stores in the target routing list are available.

**OTMA routing list (RTL) ADD command**
The OTMA routing list (RTL) **ADD** command adds an OTMA routing list to an IMS Connect Extensions repository. A routing list allows you to define collections of data stores to use as candidates for rules-based routing of IRM messages.

**Syntax**

```
ADD DEFN=(RTL,name) [DESC='description'] [REPLACE=NO|YES] DLIST=(dsname)
```

**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 data stores 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 transaction list (TXL) ADD command adds a transaction list to an IMS Connect Extensions 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.

**Syntax**

```plaintext
ADD DEFN=(TXL,name) ,DESC='description' ,REPLACE=NO ,REPLACE=YES
```

**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 transaction code (TXN) ADD command adds a transaction code to an IMS Connect Extensions 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.
Syntax

```
ADD DEFN=(TXN, name*) , DESC='description', REPLACE=NO , APP=application_name

, OVERRIDE_APP_RULES=NO , OVERRIDE_TRANSACTION_TIMER=NO

, OVERRIDE_APP_RULES=YES , OVERRIDE_TRANSACTION_TIMER=(YES, MSGTO=xx, ACKTO=xx)

, OVERRIDE_TRANSACTION_EXPIRATION=NO

, OVERRIDE_TRANSACTION_EXPIRATION=(YES, F1_TRNEXP=ON, OFF)

, OVERRIDE_CLIENTID_CANCELLATION=NO

, OVERRIDE_CLIENTID_CANCELLATION=(YES, F3_CANCELLATION=ON, OFF)

, ALTERNATE_TRANSACTION_CODE=NO

, ALTERNATE_TRANSACTION_CODE=(YES, OFFSET=nnnn, LENGTH=n)

, TRANSACTION_ROUTING=INACTIVE

, TRANSACTION_ROUTING=ACTIVE , ROUTING_RULE=(ALL, DS, name, DSG, name, AFL, name)
```

Notes:

1. 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_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 IMS data stores. IMS Connect Extensions will only route messages to those data stores that are candidate data stores 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 data stores

- **DS,name**
  A data store called “name”

- **DSG,name**
  A data store 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 data store 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 data store. This is the default.

- **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 data stores 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 IMS Connect Extensions repository.

Syntax

```
DELETE DEFN=(*,*)
DEFN=(type, name_mask)
```

Parameters

```
DEFN=(type, name_mask)
```

Specifies the type and name of the definition you want to delete.

The `type` parameter can be one of the following values:

- `*` All types.
- `AFL` Affinity list definitions.
- `APL` Application definitions.
- `DS` IMS data store definitions.
- `DSG` Data store group definitions.
- `EX` User exit definitions.
- `GRP` IMS Connect system group definitions.
- `HWS` IMS Connect system definitions.
- `IPR` IP address rule definitions.
- `ORL` ODBM routing list definitions.
- `ORR` ODBM routing rule definitions.
- `ORT` ODBM target definitions.
- `PLN` Routing plan definitions.
- `PSB` PSB name list definitions.
- `RTG` OTMA routing rule definitions.
- `RTL` OTMA routing list definitions.
TXL   Transaction list definitions.
TXN   Transaction definitions.

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.

Syntax

```
    SET OPTION=(<REPLACE=NO|YES>, <WARNING=ABORT|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 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:

```jcl
//STEP EXEC PGM=FUNEXEC,PARM='CEXDFDSO'
//STEPLIB DD DISP=SHR,DSN=funpre.SFUNLINK
// REPOSITORY DD DISP=SHR,DSN=ces.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 a configuration file for import into the IMS Connect Extensions Operations Console

The following example JCL demonstrates how to use the extract utility to generates a list of all system and group definitions in an IMS Connect Extensions repository for import into the Operations Console. The presence of the CEXCSVFI DD statement indicates that the “extract to CSV file” function is required.

```jcl
//STEP EXEC PGM=FUNEXEC,PARM='CEXDFDSO'
//STEPLIB DD DISP=SHR,DSN=funpre.SFUNLINK
// REPOSITORY DD DISP=SHR,DSN=ces.definition.repository
```
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 Common Services Library (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.
**Syntax**

```
TRANSFORM NAME=name DEFN=(*,*)
```

- **DEFN=(*,*)**
  - Specifies the type and name of the definition you want to transform.
  - The `type` parameter can be one of the following values:
    - `*`: All types.
    - `AFL`: Affinity list definitions.
    - `APL`: Application definitions.
    - `DS`: IMS data store definitions.
    - `DSG`: Datastore group definitions.
    - `EX`: Exchange definitions.
    - `GRP`: Group definitions.
    - `HWS`: Hardware definitions.
    - `IPR`: IPR definitions.
    - `ORL`: ORL definitions.
    - `ORR`: ORR definitions.
    - `ORT`: ORT definitions.
    - `PLN`: Planning definitions.
    - `PSB`: PSB definitions.
    - `RTG`: RTG definitions.
    - `RTL`: RTL definitions.
    - `TXL`: TXL definitions.
    - `TXN`: TXN definitions.

- **TO=mask**, **OVERRIDES=(DESC='description', PORT=nnnnnn, HOST=name_or_IP_addr, PRIMARY_FOR=hws_name, APP=application, APPLIST=(application), DSG=datstore_group, DSLIST=(datstore), HWSLIST=(system))**

**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.

  The `type` parameter can be one of the following values:
  - `*`: All types.
  - `AFL`: Affinity list definitions.
  - `APL`: Application definitions.
  - `DS`: IMS data store definitions.
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 data store 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 data store 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=\textit{datastore\_group}  
The datastore group of a datastore.

\textbf{DSLIST}=(\textit{datastore},...)  
The datastores in an affinity list.

\textbf{HWSLIST}=(\textit{hws},...)  
The IMS Connect systems in a system group.

**EXTRACT command**

The \texttt{EXTRACT} command extracts matching definitions from the repository; optionally applying transform rules to those definitions.

**Syntax**

```plaintext
EXTRACT DEFN=(\textit{type},\textit{name\_mask})
```

**Parameters**

\texttt{DEFN=(\textit{type},\textit{name\_mask})}

Specifies the type and name of the definition you want to extract.

The \texttt{type} parameter can be one of the following values:

* All types.
  * AFL Affinity list definitions.
  * APL Application definitions.
  * DS IMS data store definitions.
  * DSG Data store group definitions.
  * EX User exit definitions.
  * GRP IMS Connect system group definitions.
  * HWS IMS Connect system definitions.
  * IPR IP address rule definitions.
  * ORL ODBM routing list definitions.
  * PLN
  * PSB
  * RTG
  * RTL
  * TXL
  * TXN
```

Chapter 18. Definition utilities 419
ORR  ODBM routing rule definitions.
ORT  ODBM target definitions.
PLN  Routing plan definitions.
PSB  PSB name list definitions.
RTG  OTMA routing rule definitions.
RTL  OTMA routing list definitions.
TXL  Transaction list definitions.
TXN  Transaction definitions.

You can use a name mask to extract definitions containing a matching substring. For example, to extract all definitions prefixed with DEV enter:
```
DEFN=(*,DEV*)
```

**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.

**Syntax**
```
SET [REPOSTRY=ddname,] [CEXOUT=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 *
//REPOSTRY DD DISP=SHR,DSN=REPOSTRY.ONE
//CEXOUT DD UNIT=SYSDA,SPACE=(CYL,(5,2)),DISP=(,PASS),DSN=OUT.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=(*,*)
SET REPOSTRY=REP2,CEXOUT=OUT2
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 the **CEXOUT** DD statement.

2 The **SET** command changes **CEXOUT** and **REPOSTRY**. 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.
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
```

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 an IMS Connect Extensions repository.

**Syntax**
Parameters

\texttt{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:

\texttt{DEFN=(*,DEVT*)}

You can specify multiple \texttt{DEFN} statements separated by commas. For example:

\texttt{EXPORT DEFN=(ACD,*)}, \texttt{DEFN=(AFL,*)}, \texttt{DEFN=(TXN,*)}

To specify a definition type, use one of the following:

- \texttt{ACD} Active journal definitions
- \texttt{AFL} Affinity list definitions
- \texttt{APP} Application definitions
- \texttt{ARC} Archive data set records
- \texttt{ARD} Archive journal definitions
- \texttt{CKP} Journal checkpoint records
- \texttt{DS} IMS data store definitions
- \texttt{DSG} IMS data store group definitions
- \texttt{EX} User message exit definitions
- \texttt{GRP} IMS Connect system group definitions
- \texttt{HWS} IMS Connect system definitions
- \texttt{IPR} IP address rule definitions
ORL  ODBM routing list definitions
ORR  ODBM routing rule definitions
ORT  ODBM target definitions
PLN  Routing plan definitions
PSB  PSB name list definitions
RTG  OTMA routing rule definitions
RTL  OTMA routing list definitions
TXL  Transaction list definitions
TXN  Transaction definitions

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 an IMS Connect Extensions repository.

**Syntax**

```
IMPORT DEFN=(type,name_mask)
```

**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 DEV T enter:

```
DEFN=(*,DEV T*)
```

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 list definitions
- **APP**: Application definitions
- **ARC**: Archive data set records
- **ARD**: Archive journal definitions
- **CKP**: Journal checkpoint records
- **DS**: IMS data store definitions
- **DSG**: IMS data store group definitions
- **EX**: User message exit definitions
- **GRP**: IMS Connect system group definitions
- **HWS**: IMS Connect system definitions
- **IPR**: IP address rule definitions
- **ORL**: ODBM routing list definitions
- **ORR**: ODBM routing rule definitions
- **ORT**: ODBM target definitions
- **PLN**: Routing plan definitions
- **PSB**: PSB name list definitions
- **RTG**: OTMA routing rule definitions
- **RTL**: OTMA routing list definitions
- **TXL**: Transaction list definitions
- **TXN**: Transaction definitions

**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.
Syntax

►►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.

Syntax

►►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`.
Part 6. Application programming interfaces to IMS Connect Extensions

These topics provide information on the application programming interfaces (APIs) supplied with IMS Connect Extensions.

Topics:

- Chapter 19, “IMS Connect Extensions host command environment for REXX,” on page 429
- Chapter 20, “IMS Connect Extensions publisher API,” on page 463
Chapter 19. IMS Connect Extensions host command environment for REXX

Use the services of the IMS Connect Extensions host command environment for REXX to write REXX execs that can interact with and control IMS Connect Extensions features. You can use these features to query system statistics, suspend and resume routing to an IMS data store, activate a new routing plan, switch IMS Connect Extensions active journal data sets, start a trace, submit IMS Connect commands, and move toward increased automation of your IMS environment.

Related concepts:
Chapter 10, “Automating IMS Connect operations with REXX,” on page 221

The IMS Connect Extensions host command environment for REXX enables IMS Connect Extensions commands to be embedded in REXX programs. This facility allows you to automate IMS Connect operational tasks and to integrate IMS functions from other REXX command interfaces such as the IMS REXX SPOC API and the SDSF REXX programming interface.

ADD command

The ADD host command for REXX adds an entry to the in-memory IMS data store table for a data store that is defined to IMS Connect but was not defined to IMS Connect Extensions at startup.

When an IMS data store is to be dynamically added to IMS Connect, it is preferable to add the IMS data store definition in IMS Connect Extensions first. However, if the IMS CREATE IMSCON type-2 command has already been issued to add data store definitions during runtime processing, you can use the ADD host command for REXX to add the data store definition. This makes the data store available to IMS Connect Extensions without having to restart IMS Connect.

Notes:

- If the new IMS data store has been defined in one or more routing rules, refresh the OTMA routing rules using the REFRESH command or Refresh Commands panel. This makes the dynamically added data store available for OTMA rules-based routing and visible in the Status Monitor. See “Refreshing in-memory definitions from the IMS Connect Extensions repository” on page 150.

- If the data store credentials are still cached from when the data store was originally added to IMS Connect, they are used. If the cached data store credentials are not available, the details supplied in the ADD host command, including the values specified in the TMEMBER and optional SMEMBER fields are used.

Syntax

```
ADD—DS=datastore,—MEMBER=member_name,—TMEMBER=tmember_name,
```
Parameters

DS    The name of the data store that is to be added.

MEMBER
    The XCF member name that identifies IMS Connect in the XCF group that is specified by the GROUP parameter.

TMEMBER
    The XCF member name of the IMS that this IMS Connect communicates with in the XCF group.

SMEMBER
    Optional. The name of the OTMA super member to which this data store belongs.

APPLNAME
    Optional. The TCP/IP APPL name defined to RACF in the PTKTDATA statement.

GROUP
    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 host command.

Related tasks:

“Dynamically adding IMS data stores as targets for OTMA routing” on page 274

IMS Connect Extensions supports rules-based routing of OTMA messages to dynamically added IMS data stores without restarting IMS Connect.

CLEAR command

The CLEAR host command for REXX clears one or more users from the ACEE cache.
Syntax

CLEAR  TYPE=ACEE, USERID=user_id, STEM=CEX, 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.

Related concepts:

"Caching user credentials to improve performance" on page 301

When Activate Security is enabled in the IMS Connect system definition, IMS Connect Extensions performs user ID and password validation by issuing a RACROUTE VERIFY request. If Activate ACEE Cache is enabled and a value specified in the Ageing interval field, the Accessor Environment Element (ACEE) created by the RACROUTE VERIFY is cached 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.

CONNECT command

The CONNECT host command for REXX 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 connection ID (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=localhost, HOST=host_name,
Parameters

**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.

When the HOST parameter is not specified, the host name defaults to localhost.

**Note:** The following deprecated 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 on the host specified on the HOST parameter.

**PASSTIK or USERID and PASSWORD**
Select an authentication option. Authentication via username and password combination or via a PassTicket is required in the following situation:

- **Activate Access Control** is enabled in the system definition. See “Defining IMS Connect systems” on page 320.
- The exec will issue host commands that update the IMS Connect Extensions repository or in-memory definition, such as the ADD command, REFRESH command, or UPDATE command, or the exec will issue host commands that affect the operation of IMS or IMS Connect, such as the CLEAR command, DRAIN command, ROUTE command, or SWITCH command.

**PASSTIK=YES**
When no option is specified, PASSTIK=YES is the default. If **Activate Access Control** is enabled in the system definition, attempt to generate a PassTicket for the user ID associated with the job step. If **Activate Access Control** is not enabled, PASSTIK=NO is used.

**Note:** PASSWORD=PASSTIK is accepted as a synonym for PASSTIK=YES.

**USERID and PASSWORD**
Connect using the specified user ID and password.

**PASSTIK=NO**
Do not generate a PassTicket.

**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).

Example
Here is an example of a control card with the CONNECT command:

```
address CEX "CONNECT HOST=SRV2,PORT=4199,HWSID=HWSOPGS1,","PASSTIK=YES,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 an IMS Connect Extensions console listener” on page 483.

Related reference:
“Example REXX for a single system ACEE cache statistics query” on page 229
This sample REXX exec shows how to connect to an IMS Connect system, run a query to obtain ACEE cache statistics, and then display the results.

“Example REXX query of two systems for messages with responses pending” on page 231
This sample REXX exec shows how to combine the results of a query run on two IMS Connect systems that returns the number of messages that are pending a response from IMS.

DELAY command
The DELAY host command for REXX pauses command execution for a specified number of seconds.

Syntax

```
DELAY TIME=ssss
```

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 the DELAY command to poll at set intervals while checking the progress of an IMS data store drain request issued by the ROUTE 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 command**

The **DRAIN** host command for REXX 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**

```
DRAIN TYPE=SESSION CLIENTID=client_id TOKEN=svt_token, STEM=CEX
Connor=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.

**Tip:** This corresponds to the Event Key that is displayed on the Active Sessions ISPF panel. See **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

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRNCNT</td>
<td>Count of sessions marked for draining</td>
</tr>
</tbody>
</table>

Example

```
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.

**Related tasks:**

"Rebalancing sessions across IMS Connect systems" on page 276

The **Session Message Limit** option in the IMS Connect system definition allows you to specify the maximum number of input messages for a persistent session before the session is automatically closed. In an IMS sysplex distributor environment, if you design your applications to automatically reconnect when the session is closed, new persistent sessions will be routed to the IMS Connect with the lowest session totals. This feature is particularly useful when an IMS Connect system is being brought back online to quickly rebalance your persistent sessions across IMS Connect systems.

---

**OPTION command**

The **OPTION** host command for REXX is used to specify run-time options for the REXX host command environment. Use this command to adjust message verbosity and for connection management.

**Syntax**

```
OPTION [MSGLVL=QUIET] [TIMEZONE=CLIENT] [TIMEZONE=GMT] [TIMEZONE=SERVER] [CONID=conid] [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 MSGLEVEL=VERBOSE,TIMEZONE=GMT,CONID=SRV3
QUERY TYPE=SESSIONS,PERSISTENT=Y,SESTYPE=OTMA
```

Related reference:

- “Example REXX for a single system ACEE cache statistics query” on page 229
- “Example REXX query of two systems for messages with responses pending” on page 231

QUERY commands

The QUERY host commands for REXX reports on statistics and settings for IMS Connect Extensions resources.

Related tasks:

- “Exporting data to a CSV file” on page 185

Use the Export option to save the data that is currently displayed in the IMS Connect Extensions Operations Console to a file in CSV (comma-separated values) format.

QUERY TYPE=ACEE_CACHE command

The QUERY TYPE=ACEE_CACHE host command for REXX retrieves information about the caching of security credentials in IMS Connect to improve performance.
Syntax

```
\textbf{QUERY} \texttt{TYPE=ACEE\_CACHE}, \texttt{STEM=CEX}, \texttt{STEM=stem}, \texttt{CONID=conid}
```

Parameters

\textbf{TYPE=ACEE\_CACHE}
Retrieves 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.

\textbf{STEM}
The common stem for the REXX variables that are set by this command. The default stem is CEX.

\textbf{CONID}
A unique connection ID that was established by a previous \texttt{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 \texttt{CONNECT} command or by an \texttt{OPTION} command using the CONID keyword.

Response variables

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACEEPCT</td>
<td>Percentage of the ACEE cache used</td>
</tr>
<tr>
<td>DELETES</td>
<td>Number of elements deleted from the ACEE cache</td>
</tr>
<tr>
<td>ELEMLEN</td>
<td>Length of each element in the ACEE cache</td>
</tr>
<tr>
<td>ELEMNUM</td>
<td>Number of elements in the ACEE cache</td>
</tr>
<tr>
<td>GETNEXTS</td>
<td>Number of get next requests to the ACEE cache</td>
</tr>
<tr>
<td>GETS</td>
<td>Number of get requests to the ACEE cache</td>
</tr>
<tr>
<td>INITSIZE</td>
<td>Initial size of the ACEE cache</td>
</tr>
<tr>
<td>INSERTS</td>
<td>Number of elements inserted in the ACEE cache</td>
</tr>
<tr>
<td>STATS</td>
<td>Number of stat requests to the ACEE cache</td>
</tr>
<tr>
<td>XPNDCNT</td>
<td>The number of times the ACEE cache has been expanded</td>
</tr>
<tr>
<td>XPNDMAX</td>
<td>The maximum number of times the ACEE cache can be expanded</td>
</tr>
<tr>
<td>XPNDSIZE</td>
<td>The increment by which the ACEE cache can be expanded</td>
</tr>
</tbody>
</table>

Example

The CEXRXC44 member in the SCEXSAMP library contains a detailed example of how to report ACEE cache statistics for a system.

Related reference:

"Example REXX for a single system ACEE cache statistics query" on page 229
This sample REXX exec shows how to connect to an IMS Connect system, run a query to obtain ACEE cache statistics, and then display the results.
**QUERY TYPE=JOURNAL command**

For IMS Connect Extensions V3.1 systems and higher, the **QUERY TYPE=JOURNAL** host command for REXX returns information about the active IMS Connect Extensions journal.

**Syntax**

```
QUERY TYPE=JOURNAL,STEM=CEX
QUERY TYPE=JOURNAL,STEM=STEM,CONID=CONID
```

**Parameters**

- **TYPE=JOURNAL**
  Returns information about the active IMS Connect Extensions journal.

- **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**

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JRNDSN</td>
<td>Active journal data set name</td>
</tr>
<tr>
<td>JRNMAX</td>
<td>Maximum journal size in tracks</td>
</tr>
<tr>
<td>JRNTRK</td>
<td>Current track number being written</td>
</tr>
<tr>
<td>JRNTPCT</td>
<td>Percentage of journal used</td>
</tr>
</tbody>
</table>

**Example**

The CEXRXC03 member in the SCEXSAMP library contains a detailed example of how to check the active journal utilization for an IMS Connect system using the **QUERY** command and then issue a **SWITCH** command to switch the active journal if the observed active journal utilization falls within a specified range.

The CEXRXC46 member in the SCEXSAMP library contains a detailed example of how to query an IMS Connect system and return utilization details for the active journal data set via the REXXOUT data set.

**QUERY TYPE=PENDING_RESPONSES command**

The **QUERY TYPE=PENDING_RESPONSES** host command for REXX returns the number of messages that are pending a response from IMS (that is, messages that are in a Waiting for Datastore state).

**Syntax**
Parameters

**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.

**Tip:** Use this query in an automated script to drain (suspend) and then safely shut down an IMS system when the number of pending inbound and outbound messages falls to zero. See “REXX automation samples for the IMS Connect Extensions host command environment” on page 223.

**DATASTORE**

A single IMS data store that is to be referenced by **QUERY** command processing. See “Defining IMS data stores” on page 339.

**DSLIST**

An OTMA routing list comprising the IMS data stores that are to be referenced by **QUERY** command processing. See “Defining OTMA routing lists” on page 361.

**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

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DST.0</td>
<td>The number of data stores. Subsequent fields are qualified with a numeric in the range 1 to DST.0</td>
</tr>
<tr>
<td>DSTNAME.n</td>
<td>The data store names</td>
</tr>
<tr>
<td>PENDING.n</td>
<td>Pending response counts</td>
</tr>
<tr>
<td>STATUS.n</td>
<td>Status of the data stores</td>
</tr>
</tbody>
</table>

Example

The CEXRXC21, CEXRXC22, and CEXRXC23 members in the SCEXSAMP library contain detailed examples of how to drain one or more IMS data stores and then shut down and restart the IMS system.

Related reference:
This sample REXX exec shows how to combine the results of a query run on two IMS Connect systems that returns the number of messages that are pending a response from IMS.

**QUERY TYPE=PLAN command**

The `QUERY TYPE=PLAN` host command for REXX returns the names of the currently active OTMA routing plan and ODBM routing plan, or blank if they are inactive.

**Syntax**

```
QUERY TYPE=PLAN
,STEM=CEX
,STEM=stem
,CONID=conid
```

**Parameters**

- **TYPE=PLAN**
  Returns the names of the currently active OTMA routing plan and ODBM routing plan, or blank if they are inactive.
  
  **Tip:** You can use the REXX variable containing the plan name and then issue a `SET` command to change it.

- **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**

<table>
<thead>
<tr>
<th>Variable name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OTMAPLAN</td>
<td>Routing plan used for OTMA messages (blank if no OTMA routing plan is inactive)</td>
</tr>
<tr>
<td>ODBMPLAN</td>
<td>Routing plan used for DRDA requests (blank if no ODBM routing plan is inactive)</td>
</tr>
</tbody>
</table>

**Examples**

The CEXRXC31 member in the SCEXSAMP library contains an example of how to set an OTMA routing plan or ODBM routing plan.

**Related reference:**

"Defining routing plans" on page 370

Routing plans are used in IMS Connect Extensions to group routing rules that are intended to operate at the same time. To define a routing plan, use the Routing Plans panel. To access this ISPF panel, select option 1.16 **Routing Plans** from the IMS Connect Extensions primary menu.
QUERY TYPE=SESSIONS command

The QUERY TYPE=SESSIONS host command for REXX returns active session attributes in REXX stem variables.

Syntax

```
QUERY TYPE=SESSIONS
```

Parameters

**TYPE=SESSIONS**
Returns active session attributes in REXX stem variables.

**PERSISTENT**
Specify PERSISTENT=YES 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 ssss. WAITTIME=0 returns all matching sessions, whatever their value for wait time.

**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**

**Tip:** You can use the **OPTION** command to specify a timezone offset for displaying time fields.

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SESSION.0</td>
<td>Count of session data returned</td>
</tr>
<tr>
<td>SESSION._ALT_TXNUSED</td>
<td>Alternate transaction is used</td>
</tr>
<tr>
<td>SESSION._ALT_TXNCODE</td>
<td>Alternate transaction code</td>
</tr>
<tr>
<td>SESSION._ALT_TXN_LENGTH</td>
<td>Alternate transaction length</td>
</tr>
<tr>
<td>SESSION.CLIENTFAMILY</td>
<td>IPv4 or IPv6</td>
</tr>
<tr>
<td>SESSION.CLIENTID</td>
<td>Client ID</td>
</tr>
<tr>
<td>SESSION.CLIENTIPADDRESS</td>
<td>Client IP address</td>
</tr>
<tr>
<td>SESSION.CLIENTPORT</td>
<td>Client port number</td>
</tr>
<tr>
<td>SESSION.COMMITMODE</td>
<td>Commit mode</td>
</tr>
<tr>
<td>SESSION.EXITDEFINED</td>
<td>Exit defined flag, value is YES or NO</td>
</tr>
<tr>
<td>SESSION.EXITNAME</td>
<td>Name of the exit that processed the session</td>
</tr>
<tr>
<td>SESSION.ICLIENTID</td>
<td>Identifier of the initial connecting client</td>
</tr>
<tr>
<td>SESSION.INCONVERSATION</td>
<td>Conversation in progress</td>
</tr>
<tr>
<td>SESSION.INPUT_ALIAS</td>
<td>Routing exit input alias</td>
</tr>
<tr>
<td>SESSION.IRMTIMER</td>
<td>Response reason code</td>
</tr>
<tr>
<td>SESSION.LASTTRACE_TIME</td>
<td>Time last event occurred on the session</td>
</tr>
<tr>
<td>SESSION.LTERM</td>
<td>LTERM override name</td>
</tr>
<tr>
<td>SESSION.MSCCONNUID</td>
<td>MSC connection user ID</td>
</tr>
<tr>
<td>SESSION.MSCGENIMSID</td>
<td>MSC generic IMS identifier</td>
</tr>
<tr>
<td>SESSION.MSCLCLIMSID</td>
<td>MSC local IMS identifier</td>
</tr>
<tr>
<td>SESSION.MSCLCLPLKID</td>
<td>MSC local physical link</td>
</tr>
<tr>
<td>SESSION.MSCPARTNERID</td>
<td>MSC partner</td>
</tr>
<tr>
<td>SESSION.MSCR</td>
<td>MSC return code</td>
</tr>
<tr>
<td>SESSION.MSCREMOTEIMSID</td>
<td>MSC remote IMS identifier</td>
</tr>
<tr>
<td>SESSION.MSCRMTPLKID</td>
<td>MSC remote physical link</td>
</tr>
<tr>
<td>SESSION.MSCRSN</td>
<td>MSC reason code</td>
</tr>
<tr>
<td>SESSION.ODBMNAME</td>
<td>Routing exit ODBM name</td>
</tr>
<tr>
<td>SESSION.ORIGINALDS</td>
<td>Datastore specified in the message as the target</td>
</tr>
<tr>
<td>SESSION.OUTPUT_ALIAS</td>
<td>Routing exit output alias</td>
</tr>
<tr>
<td>SESSION.OUTPUTGROUP</td>
<td>Security exit output group</td>
</tr>
<tr>
<td>SESSION.OUTPUTUSERID</td>
<td>Security exit output user ID</td>
</tr>
<tr>
<td>SESSION.PORT</td>
<td>Port</td>
</tr>
<tr>
<td>SESSION.PREDICTEDSTATUS</td>
<td>Description of session status</td>
</tr>
<tr>
<td>SESSION.PSBNAME</td>
<td>Routing exit PSB name</td>
</tr>
<tr>
<td>SESSION.READEXITRC</td>
<td>Exit return code</td>
</tr>
<tr>
<td>SESSION.READEXITRSN</td>
<td>Exit reason code</td>
</tr>
<tr>
<td>SESSION.REROUTENAME</td>
<td>Reroute name</td>
</tr>
<tr>
<td>Field name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>SESSION.ROUTINGEXITRC.n</td>
<td>Routing exit return code</td>
</tr>
<tr>
<td>SESSION.ROUTINGEXITRSN.n</td>
<td>Routing exit reason code</td>
</tr>
<tr>
<td>SESSION.ROUTINGSERVRC.n</td>
<td>Routing server return code</td>
</tr>
<tr>
<td>SESSION.ROUTINGSERVRSN.n</td>
<td>Routing server reason code</td>
</tr>
<tr>
<td>SESSION.RTIPIPEACTIVE.n</td>
<td>RTPIPE is active</td>
</tr>
<tr>
<td>SESSION.SECURITYEXITRC.n</td>
<td>Security exit return code</td>
</tr>
<tr>
<td>SESSION.SECURITYEXITRSN.n</td>
<td>Security exit reason code</td>
</tr>
<tr>
<td>SESSION.SECURITYSERVRC.n</td>
<td>Security server return code</td>
</tr>
<tr>
<td>SESSION.SECURITYSERVRSN.n</td>
<td>Security server reason code</td>
</tr>
<tr>
<td>SESSION.SESMSGCNT.n</td>
<td>Message count for the session</td>
</tr>
<tr>
<td>SESSION.SESSIONTYPE.n</td>
<td>Session type</td>
</tr>
<tr>
<td>SESSION.SOCKET.n</td>
<td>Socket number</td>
</tr>
<tr>
<td>SESSION.SOCKETTYPE.n</td>
<td>Socket type</td>
</tr>
<tr>
<td>SESSION.STARTTIME.n</td>
<td>Session start time</td>
</tr>
<tr>
<td>SESSION.SVTTOKEN.n</td>
<td>Session token</td>
</tr>
<tr>
<td>SESSION.SYNCHLEVEL.n</td>
<td>Synch level</td>
</tr>
<tr>
<td>SESSION.TARGETDS.n</td>
<td>Datastore assigned as the target by IMS Connect Extensions</td>
</tr>
<tr>
<td>SESSION.TRANSACTION.n</td>
<td>Transaction code</td>
</tr>
<tr>
<td>SESSION.TRIGGERTYPE.n</td>
<td>Trigger type</td>
</tr>
<tr>
<td>SESSION.USERID.n</td>
<td>User ID</td>
</tr>
<tr>
<td>SESSION.WAITTIME.n</td>
<td>Time since last event</td>
</tr>
<tr>
<td>SESSION.WRITEEXITRC.n</td>
<td>Write exit return code</td>
</tr>
<tr>
<td>SESSION.WRITEEXITRSN.n</td>
<td>Write exit reason code</td>
</tr>
<tr>
<td>SESSION.XMLADAPTERNAME.n</td>
<td>XML adapter name</td>
</tr>
<tr>
<td>SESSION.XMLMAPNAME.n</td>
<td>XML map name</td>
</tr>
</tbody>
</table>

**Examples**

The CEXRXC26 member uses `QUERY TYPE=SESSIONS` to report the progress of an IMS data store drain.

The CEXRXC42 member in the SCEXSAMP library uses `QUERY TYPE=SESSIONS` to return details of persistent sessions on a system.

**QUERY TYPE=SOCKETS command**

For IMS Connect Extensions V3.1 systems and higher, the `QUERY TYPE=SOCKETS` host command for REXX retrieves information about IMS Connect socket usage.

**Syntax**

```plaintext
QUERY TYPE=SOCKETS [STEM=CEX] [STEM=stem] [CONID=conid]
```
Parameters

TYPE=SOCKETS
For IMS Connect Extensions V3.1 systems and higher, retrieves information about IMS Connect socket usage.

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

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAXSOC</td>
<td>The number of sockets that the IMS Connect can open at once as defined by the MAXSOC parameter in the TCPIP statement in the HWSCFGxx member of the IMS PROCLIB data set</td>
</tr>
<tr>
<td>WARNSOC</td>
<td>Socket warning level percentage as defined by the WARNSOC parameter in the TCPIP statement in the HWSCFGxx member of the IMS PROCLIB data set</td>
</tr>
<tr>
<td>WARNINC</td>
<td>Socket warning level incremental percentage as defined by the WARNINC parameter in the TCPIP statement in the HWSCFGxx member of the IMS PROCLIB data set</td>
</tr>
<tr>
<td>AODBCNT</td>
<td>Number of ODBM sessions</td>
</tr>
<tr>
<td>ASESCNT</td>
<td>Total number of sessions</td>
</tr>
<tr>
<td>PORTS</td>
<td>Number of IMS Connect ports in use</td>
</tr>
<tr>
<td>SOCKETS</td>
<td>Number of IMS Connect ports in use</td>
</tr>
<tr>
<td>SOCPCT</td>
<td>Number of sockets in use as a percentage of MAXSOC</td>
</tr>
</tbody>
</table>

Examples

The CEXRXC47 member in the SCEXSAMP library contains a detailed example of how to query socket utilization for an IMS Connect system.

The CEXRXC81 member in the SCEXSAMP library contains a detailed example of how to identify and optionally attempt to resolve a situation where an IMS Connect system's socket usage is approaching the system's MAXSOC value by canceling idle persistent OTMA sessions using the SHELL command.

QUERY TYPE=STATUS_MONITOR command

The QUERY TYPE=STATUS_MONITOR host command for REXX retrieves IMS Connect and IMS data store status and statistics. Retrieve information on TCP/IP ports, IMS Connect, IMS data stores, user exits, ODBM targets, IMS aliases, MSC link statistics, and remote IMS Connect system statistics.

Syntax
Notes:

1. Queries using SMTYPE=PORT can specify a value in the SMINTRVL parameter of 1 (20 seconds) or 2 (1 minute) only.

Parameters

TYPE=STATUS_MONITOR
Returns status monitor statistics in REXX stem variables.

SMTYPE
Use this field to select the types of statistics displayed:

PORT  Return TCP/IP port (and local adaptor) statistics
IMS_CONNECT  Return IMS Connect (HWS) system statistics
DATASTORE  Return IMS data store statistics
DATASTORE_GROUP  Return data store group statistics
EXIT  Return user exit statistics
ODBM  Return Open Database Manager (ODBM) target statistics
ALIAS  Return IMS alias statistics
MSC  Return IMS Multiple Systems Coupling (MSC) link statistics
REMOTE_CONNECT  Return remote IMS Connect system statistics

SMINTRVL
Event statistics are collected every 20 seconds. To analyze activity in the last hour or part thereof, specify the desired interval:

1 20 seconds. The last (most recent) 20 second interval.
2 Minute. The last minute (sum of the last three 20 second intervals).
3 1-15 minutes. The last quarter hour (sum of the most recent 15 minutes).
4 16-30 minutes. The second-last quarter hour (sum of 16-30 minutes ago).
5 31-45 minutes. The third-last quarter hour (sum of 31-45 minutes ago).
6 46-60 minutes. The fourth-last quarter hour (sum of 46-60 minutes ago).

7 Hour. Sum of activity in the last hour.

Note: Queries using SMTYPE=PORT can use an interval of 1 (20 seconds) or 2 (1 minute) only.

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

Responses contain the following information:

- A response header
- Variables specific to the value specified in the SMTYPE parameter
- Statistics for SMTYPE requests for type PORT, IMS_CONNECT, DATASTORE, DATASTORE_GROUP, and EXIT

Response header

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STATMON.0</td>
<td>Count of data element returned</td>
</tr>
<tr>
<td>STATMON.TIME.n</td>
<td>Statistics time</td>
</tr>
<tr>
<td>STATMON.INTERVAL.n</td>
<td>Statistics interval corresponding to value selected using the SMINTRVL parameter (e.g. last 20 seconds, last minute, minutes 1-15, minutes 16-30, minutes 31-45, minutes 46-60, last hour)</td>
</tr>
<tr>
<td>STATMON.DURATION.n</td>
<td>Statistics duration (in seconds)</td>
</tr>
<tr>
<td>STATMON.SYSTEM.n</td>
<td>System</td>
</tr>
<tr>
<td>STATMON.NAME.n</td>
<td>Name</td>
</tr>
<tr>
<td>STATMON.TYPE.n</td>
<td>Type</td>
</tr>
<tr>
<td>STATMON.STATUS.n</td>
<td>Status</td>
</tr>
</tbody>
</table>

Variables specific to SMTYPE=PORT

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STATMON.KEEPALIVE.n</td>
<td>KeepAlive value</td>
</tr>
</tbody>
</table>

Variables specific to SMTYPE=IMS_CONNECT

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STATMON.HWSSTARTTIME.n</td>
<td>HWS start time</td>
</tr>
<tr>
<td>STATMON.OTMAPLAN.n</td>
<td>OTMA routing plan name</td>
</tr>
<tr>
<td>STATMON.ODBMPLAN.n</td>
<td>ODBM routing plan name</td>
</tr>
<tr>
<td>Field name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>STATMON.MSGLIMIT,n</td>
<td>Session message limit status</td>
</tr>
<tr>
<td>STATMON.LIMITTHREAD,n</td>
<td>Session message limit threshold</td>
</tr>
<tr>
<td>STATMON.COLLECTLEV,n</td>
<td>Event collection level</td>
</tr>
<tr>
<td>STATMON.PREEXIT,n</td>
<td>Pre-exit status</td>
</tr>
<tr>
<td>STATMON.JRNPCT,n</td>
<td>For IMS Connect Extensions V3.1 systems or higher, the percentage of the</td>
</tr>
<tr>
<td></td>
<td>active IMS Connect Extensions journal that is used.</td>
</tr>
<tr>
<td>STATMON.SOCPCT,n</td>
<td>For IMS Connect Extensions V3.1 systems or higher, the current number of</td>
</tr>
<tr>
<td></td>
<td>sockets used as a percentage of MAXSOC.</td>
</tr>
<tr>
<td>STATMON.SUPERMEMBER,n</td>
<td>The OTMA super member name defined on the SMEMBER parameter defined in your</td>
</tr>
<tr>
<td></td>
<td>system configuration</td>
</tr>
</tbody>
</table>

### Variables specific to SMTYPE=DATASTORE

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STATMON.CWR,n</td>
<td>Datastore capacity weight rating</td>
</tr>
<tr>
<td>STATMON.WAITINGREPLY,n</td>
<td>Datastore pending response count</td>
</tr>
<tr>
<td>STATMON.CONNECTSTATUS,n</td>
<td>Datastore IMS Connect status. This field shows if the Cross-System Coupling</td>
</tr>
<tr>
<td></td>
<td>Facility (XCF) connection is Active, Inactive, or Discon (disconnected).</td>
</tr>
<tr>
<td>STATMON.IMSSTATUS,n</td>
<td>Datastore IMS (OTMA) status</td>
</tr>
<tr>
<td>STATMON_ROUTINGSTATUS,n</td>
<td>Datastore routing status</td>
</tr>
<tr>
<td>STATMON.MEMBER,n</td>
<td>Datastore XCF member</td>
</tr>
<tr>
<td>STATMON.TMEMBER,n</td>
<td>Datastore XCF Tmember</td>
</tr>
<tr>
<td>STATMON.XCFGROUP,n</td>
<td>Datastore XCF group</td>
</tr>
<tr>
<td>STATMON.SUPERMEMBER,n</td>
<td>The OTMA super member name defined on the SMEMBER parameter defined in your</td>
</tr>
<tr>
<td></td>
<td>system configuration</td>
</tr>
</tbody>
</table>

### Variables specific to SMTYPE=ALIAS

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STATMON.ALIAS,n</td>
<td>IMSAlias name</td>
</tr>
</tbody>
</table>

### Statistics for SMTYPE requests for type PORT, IMS_CONNECT, DATASTORE, DATASTORE_GROUP, and EXIT

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STATMON.ACCEPTCNT,n</td>
<td>Accepted count</td>
</tr>
<tr>
<td>STATMON.ACKCNT,n</td>
<td>ACK count</td>
</tr>
<tr>
<td>STATMON.DEALLOCCNT,n</td>
<td>DEALLOC count</td>
</tr>
<tr>
<td>STATMON.IGNRCNT,n</td>
<td>Ignored count</td>
</tr>
<tr>
<td>STATMON.INPUTCNT,n</td>
<td>Input count</td>
</tr>
<tr>
<td>Field name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>STATMON.INPUTBYTES</td>
<td>Input cumulative bytes</td>
</tr>
<tr>
<td>STATMON.INPUTMAX</td>
<td>Input maximum length</td>
</tr>
<tr>
<td>STATMON.INPUTMIN</td>
<td>Input minimum length</td>
</tr>
<tr>
<td>STATMON.NACKCNT</td>
<td>NAK count</td>
</tr>
<tr>
<td>STATMON.REJECTCNT</td>
<td>Rejected count</td>
</tr>
<tr>
<td>STATMON.REJEXERCNT</td>
<td>Rejected by EXER exit count</td>
</tr>
<tr>
<td>STATMON.REJOTHRCNT</td>
<td>Rejected other count</td>
</tr>
<tr>
<td>STATMON.REJPACECNT</td>
<td>Rejected by pacing count</td>
</tr>
<tr>
<td>STATMON.REJRUTCNT</td>
<td>Rejected by routing count</td>
</tr>
<tr>
<td>STATMON.REJSECYCNT</td>
<td>Rejected by security count</td>
</tr>
<tr>
<td>STATMON.REQUESTCNT</td>
<td>Request count</td>
</tr>
<tr>
<td>STATMON.RETURNCNT</td>
<td>Returned count</td>
</tr>
<tr>
<td>STATMON.ROUTECTNT</td>
<td>Routed count</td>
</tr>
<tr>
<td>STATMON.RTIPECNT</td>
<td>RPIPE count</td>
</tr>
<tr>
<td>STATMON.RTIPIPES</td>
<td>RPIPE cumulative bytes</td>
</tr>
<tr>
<td>STATMON.RTIPIPEMAX</td>
<td>RPIPE maximum length</td>
</tr>
<tr>
<td>STATMON.RTIPEMIN</td>
<td>RPIPE minimum length</td>
</tr>
<tr>
<td>STATMON.SENDONLYCNT</td>
<td>Send only count</td>
</tr>
<tr>
<td>STATMON.SENDONLYBYTES</td>
<td>Send only cumulative bytes</td>
</tr>
<tr>
<td>STATMON.SENDONLYMAX</td>
<td>Send only maximum length</td>
</tr>
<tr>
<td>STATMON.SENDONLYMIN</td>
<td>Send only minimum length</td>
</tr>
<tr>
<td>STATMON.SENDERRCNT</td>
<td>Sent error count</td>
</tr>
<tr>
<td>STATMON.SENDERRBYTES</td>
<td>Sent error cumulative bytes</td>
</tr>
<tr>
<td>STATMON.SENDERRMAX</td>
<td>Sent error maximum length</td>
</tr>
<tr>
<td>STATMON.SENDERRMIN</td>
<td>Sent error minimum length</td>
</tr>
<tr>
<td>STATMON.SENTOKCNT</td>
<td>Send OK count</td>
</tr>
<tr>
<td>STATMON.SENTOKBYTES</td>
<td>Sent OK cumulative bytes</td>
</tr>
<tr>
<td>STATMON.SENTOKMAX</td>
<td>Sent OK maximum length</td>
</tr>
<tr>
<td>STATMON.SENTOKMIN</td>
<td>Sent OK minimum length</td>
</tr>
</tbody>
</table>

**Examples**

The CEXRXC43 member in the SCEXSAMP library uses `QUERY` command with `TYPE=STATUS_MONITOR` to return statistics.

**QUERY TYPE=TRACE command**

The `QUERY TYPE=TRACE` host command for REXX reports whether tracing is active or inactive on the system and, if it is active, the current tracing options.
Syntax

```
QUERY TYPE=TRACE
   STEM=CEX
   STEM=stem
   CONID=conid
```

Parameters

**TYPE=TRACE**
Reports whether tracing is active or inactive on the system and, if it is active, the current tracing options.

**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

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTIVE</td>
<td>Indicates whether tracing is active</td>
</tr>
<tr>
<td>CONDNAME</td>
<td>The specific condition for which tracing is requested</td>
</tr>
<tr>
<td>CONDTYPE</td>
<td>The type of condition for which tracing is requested</td>
</tr>
<tr>
<td>LEVEL</td>
<td>Tracing level</td>
</tr>
<tr>
<td>PERSISTS</td>
<td>Indicates whether tracing has been configured to continue after a system restart, or until it is explicitly deactivated.</td>
</tr>
<tr>
<td>PORT</td>
<td>Tracing port</td>
</tr>
</tbody>
</table>

Examples

The CEXRXC45 member in the SCEXSAMP library contains a detailed example of how to report the tracing settings for a system.

**REFRESH command**

The REFRESH host command for REXX rebuilds the in-memory copy of the selected definitions in the target IMS Connect system from the definitions stored in your IMS Connect Extensions repository.

Syntax

Use the REFRESH host command in the following situations:
- When you have made changes to your IMS Connect Extensions repository definitions and you would like to apply those changes to your running IMS Connect systems.
• When you have made changes to your IMS Connect systems in-memory values (see “Setting in-memory IMS Connect Extensions definitions” on page 156) and you would like to revert to the values stored in IMS Connect Extensions repository definitions.

```
REFRESH-DEFN=(AFL,affinity_list)
  APP,application
  DS,datastore
  DSG,datastore_group
  EX,exit
  HWS
  IPR
  JRN
  ORR
  RTG
  SAF
  TXN,transaction

CONID=conid
```

**Parameters**

**DEFN**  Refresh values from definitions in the IMS Connect Extensions repository. Select from the following options:

- **AFL**  Refresh selected fields in the affinity list definition specified on the `affinity_list` parameter. For a list of fields, see “Refreshing in-memory definitions from the IMS Connect Extensions repository” on page 150.

- **APP**  Refresh selected fields in the application definition specified on the `application` parameter. For a list of fields, see “Refreshing in-memory definitions from the IMS Connect Extensions repository” on page 150.

- **DS**  Refresh selected fields in the IMS data store definition specified on the `datastore` parameter. For a list of fields, see “Refreshing in-memory definitions from the IMS Connect Extensions repository” on page 150.

- **DSG**  Refresh selected fields in the data store group definition specified on the `datastore_group` parameter. For a list of fields, see “Refreshing in-memory definitions from the IMS Connect Extensions repository” on page 150.

- **EX**  Refresh the user exit definition specified on the `exit` parameter. See “Managing user exits” on page 147.

- **HWS**  Refresh selected fields in all IMS Connect system definitions. For a list of fields, see “Refreshing in-memory definitions from the IMS Connect Extensions repository” on page 150.

- **IPR**  Refresh all IP address rules definitions. See “Creating workload rules for specific IP addresses” on page 305.

- **JRN**  Refresh IMS Connect Extensions journal options. See “Managing the IMS Connect Extensions journal” on page 163.
ORR  Refresh all ODBM routing rules based on the current routing rule and routing list definitions in the IMS Connect Extensions repository. See "Refreshing in-memory definitions from the IMS Connect Extensions repository" on page 150.

RTG  Refresh all OTMA routing rules based on the current routing rule, routing list, and ODBM target and PSB name list definitions in the IMS Connect Extensions repository. See "Refreshing in-memory definitions from the IMS Connect Extensions repository" on page 150.

SAF  Refresh SAF class rules. See "Issuing security commands" on page 155.

TXN  Refresh selected fields in all transaction definitions. For a list of fields, see "Refreshing in-memory definitions from the IMS Connect Extensions repository" on page 150.

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

To refresh all OTMA routing rules, use the following command:
address CEX "REFRESH DEFN=(RTG)"

To refresh all ODBM routing rules, use the following command:
address CEX "REFRESH DEFN=(ORR)"

To refresh all IMS Connect system definitions, use the following command:
address CEX "REFRESH DEFN=(HWS)"

To refresh an IMS data store definition named IMD4, use the following command:
address CEX "REFRESH DEFN=(DS,IMD4)"

The member CEXRC31 in the SCEXSAMP library shows an example of the REFRESH host command being used after activating a routing plan.

Related reference:
"Refreshing in-memory definitions from the IMS Connect Extensions repository" on page 150

Use the Refresh Commands panel to reload IMS Connect Extensions in-memory definitions for the selected system from definitions stored in the IMS Connect Extensions repository. To access this ISPF panel, select option 2 Refresh on the Commands panel.
ROUTE command

The ROUTE host command for REXX suspends message routing to an IMS data store to drain it of in-progress transactions, or resumes routing to the data store as required. The ROUTE command 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

```
ROUTE ACTION=DRAIN, DATASTORE=datastore, DLIST=OTMA_routing_list, AUTORESUME
```

```
ROUTE ACTION=RESUME, STEM=CEX, STEM=stem, CONID=conid
```

Notes:
1. Applies to ACTION=DRAIN only.

Parameters

ACTION
The routing action to perform. Select one of the following options:

DRAIN
Changes the routing status of the IMS data store or stores specified in either the DATASTORE or DLIST parameters from Normal to Suspended, and withdraws those data stores as a candidate for routing.

**Important:** The drain facility only works on messages handled by IMS Connect Extensions OTMA rules-based routing. It does not work on messages that circumvent a routing rule or with exits customized with their own routing mechanisms. To ensure that all of your workload is handled by IMS Connect Extensions, see “Rejecting transactions for data stores with no OTMA routing rule” on page 265.

RESUME
Changes the routing status of the IMS data store or stores specified in either the DATASTORE or DLIST parameters from Suspended to Normal and restores it as a candidate for routing.

DATASTORE
An IMS data store on which to perform the routing action. See “Defining IMS data stores” on page 339.

DSLIST
An OTMA routing list defined in the IMS Connect Extensions repository that contains a list of IMS data stores on which to perform the routing action. See “Defining OTMA routing lists” on page 361.

AUTORESUME
An optional keyword that indicates that IMS Connect Extensions should automatically resume routing to the IMS data stores specified in either the DATASTORE or DLIST parameters when an IMS Datastore Available
event record (X’10’) is received. The AUTORESUME parameter can only be used with the ACTION=DRAIN option.

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.

Result

To assist with automated operations, the result of the ROUTE command 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.

CEX5133I ROUTE RESUME for Datastore DS01.
CEX5133I ROUTE RESUME for Datastore DS03 Datastore is not currently drained

Examples

The CEXRXC21, CEXRXC22, and CEXRXC23 members in the SCEXSAMP library contain detailed examples of how to drain IMS data stores and then shut down and restart the IMS system.

SET command

The SET host command for REXX activates or deactivates an OTMA or ODBM routing plan.

Note: 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

```
option:
  SET PLAN TYPE=OTMA, ACTIVATE=plan, STEM=CEX
  TYPE=ODB, DEACTIVATE, STEM=stem

  CONID=conid
```

Parameters

TYPE The routing plan type: OTMA or ODBM.

ACTIVATE=plan Activate the named ODBM or OTMA routing plan defined in the IMS Connect Extensions repository. See “Defining routing plans” on page 370.

DEACTIVATE Deactivate the active routing plan. 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.

**Example**

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:

```bash
address CEX "SET PLAN,TYPE=OTMA,ACTIVATE=PEAK"
address CEX "REFRESH DEFN=RTG"
```

The CEXRXC31 member in the SCEXSAMP library contains an example of how to activate OTMA and ODBM routing plans and the rebuild the in-memory routing rules and routing lists.

**Related tasks:**

- [“Activating OTMA routing rules in IMS Connect” on page 259](#)
  OTMA routing rules defined in the IMS Connect Extensions repository only take effect in IMS Connect once you have activated OTMA rules-based routing.
- [“Activating ODBM routing rules in IMS Connect” on page 294](#)
  ODBM routing rules defined in the IMS Connect Extensions repository only take effect in IMS Connect once you have activated ODBM rules-based routing.
- [“Activating and deactivating ODBM routing plans” on page 197](#)
  Use the Update > ODBM Routing Plan context option in the Status Monitor of the Operations Console to activate or deactivate ODBM routing plan on several IMS Connect systems at once.
- [“Activating and deactivating OTMA routing plans” on page 198](#)
  Use the Update > OTMA Routing Plan context option in the Status Monitor of the Operations Console to activate or deactivate OTMA routing plan on several IMS Connect systems at once.

---

**SHELL command**

The **SHELL** host command for REXX runs an IMS Connect command or IMS type-1 command on a specified target system or IMS data store. You can use this to automate some procedures such as stopping all the IMS data stores associated with an IMS system across multiple IMS Connect instances.

**Syntax**

```bash
SHELL SYSTEM=target,COMMAND='command',STEM=CEX,STEM=stem,CONID=conid
```
Parameters

**SYSTEM**

The target system or IMS data store 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=HNSOPPS1,COMMAND='STOPPLXDH'
```

If `target` is set to a data store 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

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TARGET</td>
<td>Shell command target</td>
</tr>
<tr>
<td>TIME</td>
<td>Shell command time of day</td>
</tr>
<tr>
<td>LINE.0</td>
<td>Shell command response line count. Subsequent lines are qualified with a numeric in the range 1 to LINE.0</td>
</tr>
<tr>
<td>LINE.n</td>
<td>Shell command response line</td>
</tr>
</tbody>
</table>

Example

The CEXRXJC02 and CEXRXJC40 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.

The CEXRXJC81 member in the SCEXSAMP library contains a detailed example of how to identify and optionally attempt to resolve a situation where an IMS Connect system's socket usage is approaching the system's MAXSOC value using the "QUERY commands" on page 436 followed by the `SHELL` command to cancel any idle persistent OTMA sessions.

Related tasks:

"Issuing IMS Connect and IMS type-1 commands from the CONSOLE dialog" on page 165

Users who have the appropriate level of authority can use the IMS Connect Extensions ISPF command shell to issue IMS Connect WTOR commands, IMS Connect z/OS commands, and IMS type-1 commands from IMS Connect Extensions.

"Issuing IMS type-1 commands" on page 215

Use the **IMS Commands** tab in the IMS Connect Extensions Operations Console to issue IMS type-1 commands directly to an IMS data store.
SWITCH command

The SWITCH host command for REXX switches the active journal for an IMS Connect system.

Syntax

```
SWITCH TYPE=JOURNAL, STEM=stem, CONID=conid
```

Parameters

TYPE=JOURNAL
Switch the active IMS Connect Extensions journal.

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 IMS Connect Extensions journal for an IMS Connect system.

The CEXRXC02 member in the SCEXSAMP library issues the SWITCH command to perform a coordinated IMS Connect Extensions journal and IMS log switch.

The CEXRXC03 member in the SCEXSAMP library contains a detailed example of how to check the active IMS Connect Extensions journal utilization for an IMS Connect system using the “QUERY commands” on page 436 and then issue a SWITCH command to switch the active journal if the observed active journal utilization falls within a specified range.

Related reference:

“Managing the IMS Connect Extensions journal” on page 163

Use option 5 Journal on the Commands panel to dynamically switch the active IMS Connect Extensions journal and refresh journal settings from the repository.

TRACE command

The TRACE host command for REXX controls the IMS Connect Extensions trace. You can specify filter conditions to control which additional event records are written to the active IMS Connect Extensions journal data set.

Syntax

```
TRACE LEVEL=1, LEVEL=0, LEVEL=2, KEEP_ACTIVE=NO, KEEP_ACTIVE=YES, PORT=port, PORT=LOCAL
```

(1)
Notes:

1  Default only when no trace is currently active.

Parameters

LEVEL
Set the tracing level.

0  Deactivate tracing. IMS Connect Extensions stops writing trace records to the active journal.

1  Activate tracing, but do not include client application data in the trace records. If LEVEL is not specified and no trace is currently active, this is the default.

2  OTMA workloads only. Activate tracing and include client application data in the trace records.

KEEP_ACTIVE
The status of tracing after a system restart.

YES  Keep trace active after a system restart.

NO   Tracing continues until it is deactivated or until a system restart. If KEEP_ACTIVE is not specified and no trace is currently active, this is the default.

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. If PORT is not specified and no trace is currently active, the default is all ports (*).

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 do not need to specify conditions if you are only deactivating tracing (LEVEL=0). Otherwise, 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.

**PORTONLY**
Trace on the TCP/IP port only. No condition is set. If no COND parameter is supplied and no trace is active, this is the default.

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

**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:

```plaintext
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 concepts:**

Chapter 24, “Events recorded by IMS Connect Extensions,” on page 551
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.

**Related tasks:**

“Using the IMS Connect Extensions trace” on page 93
Tracing events are additional event records produced by IMS Connect Extensions when tracing is activated. Use the IMS Connect Extensions ISPF dialog to start and stop tracing for an active IMS Connect system.

### UPDATE command

The **UPDATE** host command for REXX 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 ODBM target, and the event collection level or session message limit options for an IMS Connect system.
Syntax

```
UPDATE COLLECTION_LEVEL=level
   DS=name,WEIGHTING=capacity_weight
   ODT=name,WEIGHTING=capacity_weight
   SES_MSG_LIMIT=ACTIVE
   SES_MSG_THRESHOLD=threshold
   CONID=conid

STEM=CEX
STEM=stem
```

Parameters

Restriction:
1. Only one of the COLLECTION_LEVEL, SES_MSG_LIMIT, SES_MSG_THRESHOLD, or WEIGHTING parameters can be specified per UPDATE command. Therefore, setting both the SES_MSG_LIMIT and the SES_MSG_THRESHOLD parameters requires separate UPDATE calls.
2. The order of the calls is significant. You can't first set SES_MSG_LIMIT=ACTIVE with an invalid threshold and then update the SES_MSG_THRESHOLD parameter to a valid value. Instead, set the SES_MSG_THRESHOLD parameter with a valid value and then set SES_MSG_LIMIT=ACTIVE.

COLLECTION_LEVEL
The event collection level determines which events are recorded in the IMS Connect Extensions 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.

Tip: For more information on how you can use the collected event data to report on IMS Connect performance, resource usage, and transit event tracing, see “IMS Connect Extensions event collection” in the IBM IMS Performance Analyzer for z/OS Report Reference.

DS with WEIGHTING
Set the capacity weight rating (CWR) for an IMS data store defined in the IMS Connect Extensions repository. See “Defining IMS data stores” on page 339.
Specify a value in the range 0 - 100. You can keep an IMS data store in reserve by assigning it a capacity of zero, making it temporarily ineligible as a routing candidate.

**ODT with WEIGHTING**

Set the capacity weight rating (CWR) for an ODBM target (ODT) defined in the IMS Connect Extensions repository. See “Defining ODBM targets” on page 362. Specify a value in the range 0 - 100. You can keep an ODBM target in reserve by assigning it a capacity of zero, making it temporarily ineligible as a routing candidate.

**SES_MSG_LIMIT**

The session message limit specifies the maximum number of input messages a persistent session can receive before it is closed. If SES_MSG_LIMIT=ACTIVE is set, 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 SES_MSG_LIMIT=ACTIVE is set.

**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, CEXRXC63, and CEXRXC64 members in the SCEXSAMP library contain examples of how to change various settings using the UPDATE command.

---

**WTO command**

The WTO host command for REXX issues a Write To Operator (WTO) message to the console. You can use this to automate responses to certain conditions.

The Write To Operator message is issued with ROUTCDE=11. This is commonly referred to as “write to programmer” (WTP).

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**

```
WTO TEXT=message_text [STEM=CEX] [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 CEXRXC21, CEXRXC22, and CEXRXC23 members in the SCEXSAMP library contain examples of using the WTO command to report various conditions.
Chapter 20. IMS Connect Extensions publisher API

Use the IMS Connect Extensions publisher API to externalize IMS Connect Extensions and IMS Connect data to third-party applications.

Data supplied by the publisher API

Clients can access the following types of data through the IMS Connect Extensions 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 in IMS Connect
- IMS Connect configuration data
- IMS data stores 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.

Activating the IMS Connect Extensions publisher API

Before you can use the IMS Connect Extensions publisher API, you must select the Activate Publisher API option in the IMS Connect system definition. See “Defining IMS Connect systems” on page 320.

Functions provided by the CEXPAPI macro

The IMS Connect Extensions 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.

![Diagram showing the relationship between functions](image)

**Figure 205. The relationship between the functions of the IMS Connect Extensions publisher API CEXPAPI macro**

**Related concepts:**
- Chapter 7, “Forwarding a live feed of IMS Connect events,” on page 99
- The IMS Connect Extensions feed collects events from running IMS Connect Extensions systems, consolidates the events into one record per IMS Connect transaction, and then forwards those records to one of three destinations. The destination can be: a remote analytics platform that is listening on a TCP/IP port for data in JSON Lines format, a data set, or System Management Facilities (SMF).

**Related tasks:**
- “IMS Connect Extensions publisher API client list” on page 170
- Use the IMS Connect Extensions publisher API client list to list summary information about client applications that are connected to the IMS Connect Extensions publisher API.

**Programming requirements and environment**

This topic provides reference information to help you develop and run clients that use 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=DSPECT
  ```
  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 CEXMAP 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 data store list returned by the `READSYNC` command.

- **CEXAPQLE**
  Maps buffer element list for active IMS Connect systems from the `QUERY` request.

- **CEXAPROQ**
  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 Chapter 25, “Mapping event records,” on page 559 for details.
Syntax

Each mapping macro has the following syntax:

\[
\text{label—macro_name—REQ=}
\begin{array}{c}
\text{DSECT} \\
\text{AREA} \\
\text{PFX=prefix}
\end{array}
\]

Parameters that can follow the macro name

REQ

Choose one of the following:

DSECT

Assembler mapping.

AREA

Assembler storage area generation.

PFX

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 of the publisher API 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.

Syntax

\[
\text{label—CEXPAPI—CONNECT—,AGENT=agent—,AUTHID=authorization_ID—}
\begin{array}{c}
\text{BUFSIZE=READSYNC_buffer_size—,TARGET=target_hws—,TOKEN=token—}
\end{array}
\begin{array}{c}
\text{TRMECB=terminate_ECB—}
\end{array}
\begin{array}{c}
\text{MF=(E-,parm_area—)}
\end{array}
\]

Parameters that can follow CONNECT

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 465 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 467 for details.

**Example**

```
CEXPAPI CONNECT, +
   TOKEN=0TOKEN, +
   AUTHID=AUTHID, +
   AGENT=CLIENTID, +
   TARGET=HWSID, +
   BUFSIZE=BUFSIZE, +
   TRMECB=0TERMECB, +
   MF=(E,PARMLST)
LTR R15,R15 Successful?
BNZ ...
```

```
0TOKEN DS A Address of token
0TERMECB DC A(TERMECB) Address of API termination ECB
AUTHID DC CLB' ' Authorization ID
CLIENTID DC CL16'SAMPLE CLIENT' Client ID
HWSID DC CLB'HWS1' IMS Connect System ID
BUFSIZE DC F'32768' Buffer size
TERMECB DC F'0' API termination ECB
```

**DISCONN**

The **DISCONN** function of the IMS Connect Extensions publisher API disconnects the client region from the target IMS Connect region.

**Syntax**

```
label—CEXPAPI—DISCONN—,TOKEN=token—MF=(E-,parm_area—)
```
Parameters that can follow DISCONN

**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 467 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 of the IMS Connect Extensions publisher API 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.

**Syntax**

```
label--CEXPAPI--ETERM--TOKEN=token--MF=(E-,parm_area--)--
```

Parameters that can follow ETERM

**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 467 for details.

**Example**

```
*---------------------------------------------------------------------*
| Terminate CEXPAPI interface and discard all pending functions     |
*---------------------------------------------------------------------*
```
**INIT**

The **INIT** function of the IMS Connect Extensions publisher API 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.

**Syntax**

```
label CEXPAPI INIT, MF=(E, parm_area)
```  

**Parameters that can follow **INIT****

**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 467 for details.

**Example**

```
*---------------------------------------------------------------------*
* Initialize CEXPAPI interface                                         *
*---------------------------------------------------------------------*
CEXPAPI INIT, +  
MF=(E,PARMLST)  
LTR R15,R15    Successful?  
BNZ ...       No, error  
ST R1, @TOKEN  After INIT,  
               R1 contains address of token  
PAPILST CEXPAPI MF=L  Parameter list  
@TOKEN DS A     Address of token
```  

**QUERY**

The **QUERY** function of the IMS Connect Extensions publisher API lists all eligible IMS Connect systems. An IMS Connect system is considered eligible if the **Activate Publisher API** option has been enabled in the IMS Connect system definition in the IMS Connect Extensions repository.

**Syntax**

```
label CEXPAPI QUERY, BUFFER=client_buffer, TOKEN=token
```
Parameters that can follow QUERY

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 467 for details.

Example

```c
*---------------------------------------------------------------------*
List all eligible IMS Connect systems
*---------------------------------------------------------------------*

CEXPAPI QUERY,
   TOKEN=@TOKEN,
   BUFFER=@BUFFER,
   MF=(E,PARMLST)
LTR R15,R15 Successful?
BNZ ...
@TOKEN DS A Address of token
@BUFFER DS A Address of buffer
BUFFER DSECT , Buffer
```

Related reference:

“Defining IMS Connect systems” on page 320

Use the System Definitions panel to create a new definition for an IMS Connect and to specify 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. To access this ISPF panel, select option 1.1 System Definitions from the IMS Connect Extensions primary menu.

READASYN

The READASYN function of the IMS Connect Extensions publisher API 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 event control block (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 465 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.

Syntax

```assembler
label—CEXPAPI—READASYN—,BUFFER=client_buffer—,TOKEN=token
         ,MF=(E,parm_area—)
```

Parameters that can follow READASYN

- **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 467 for details.

Example

```assembler
*---------------------------------------------------------------------*
Read IMS Connect Extensions event data into client buffer
*---------------------------------------------------------------------*
CEXPAPI READASYN,       +
     TOKEN=OTOKEN,       +
     BUFFER=OBUFFER,    +
     MF=(E,PARMLST)
LTR R15,R15           Successful?
BNZ ...             No, error
OTOKEN DS A           Address of token
OBUFFER DS A          Address of buffer
BUFFER DSECT ,        Buffer

READSYNC

The READSYNC function of the IMS Connect Extensions publisher API lists information about an IMS Connect system.
**Purpose**

The **READSYNC** function 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.
- IMS data stores 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 11] summarizes the settings for the RQB and the corresponding elements that are returned.

**Table 11. RQB requests**

<table>
<thead>
<tr>
<th>Request</th>
<th>RQB_PRIREQ</th>
<th>RQB_SECREQ</th>
<th>RQB_FILTER</th>
<th>Mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>All tasks</td>
<td>IIPRI_TASK</td>
<td>ISEC_ALL</td>
<td>IIFLT_NOFILTER</td>
<td>CEXPAPLLE</td>
</tr>
<tr>
<td>Port tasks only</td>
<td>IIPRI_TASK</td>
<td>ISEC_PORT</td>
<td>IIFLT_NOFILTER</td>
<td>CEXPAPLLE</td>
</tr>
<tr>
<td>All IMS data stores</td>
<td>IIPRI_DSTORE</td>
<td>ISEC_ALL</td>
<td>IIFLT_NOFILTER</td>
<td>CEXPAPDLE</td>
</tr>
<tr>
<td>Active IMS data stores only</td>
<td>IIPRI_DSTORE</td>
<td>ISEC_ALL</td>
<td>IIFLT_ACTIVE</td>
<td>CEXPAPDLE</td>
</tr>
<tr>
<td>All active sessions</td>
<td>IIPRI_SESSIONS</td>
<td>ISEC_ALL</td>
<td>IIFLT_NOFILTER</td>
<td>CEXCSKT</td>
</tr>
<tr>
<td>Active sessions that are not in Read Prepare state</td>
<td>IIPRI_SESSIONS</td>
<td>ISEC_ALL</td>
<td>IIFLT_NOTWAIT</td>
<td>CEXCSKT</td>
</tr>
<tr>
<td>Configuration member</td>
<td>IIPRI_CONFIG</td>
<td>ISEC_ALL</td>
<td>IIFLT_NOFILTER</td>
<td>CEXPAPCLE</td>
</tr>
</tbody>
</table>

**Syntax**

```plaintext
►►label—CEXPAPI—READSYNC—,BUFFER=client_buffer—►►
►RQB=request_control_block—,TOKEN=token—,MF=(E,—parm_area—)
```

**Parameters that can follow READSYNC**

**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 467 for details.
Example
*---------------------------------------------------------------------*
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 CEXPQRB REQ=AREA,PFX=RQB API request block

BUFFER DSECT Buffer

STARTMON

The STARTMON function of the IMS Connect Extensions publisher API 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.

Syntax

---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 that can follow STARTMON

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 event control block (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 467 for details.

---

**STOPMON**

The **STOPMON** function of the IMS Connect Extensions publisher API stops collecting IMS Connect events by ending the **STARTMON** process. Subsequent **READASYN** requests are rejected by the server until another **STARTMON** is issued.

**Syntax**

```
label—CEXPAPI—STOPMON—,TOKEN=token—,...,MF=(E-,parm_area—)
```

**Parameters that can follow STOPMON**

**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 467 for details.
Example

*---------------------------------------------------------------------*
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 of the IMS Connect Extensions publisher API terminates the CEXPAPI interface.

Syntax

```plaintext
label—CEXPAPI—TERM—,TOKEN=token—,MF=(E,parm_area—)
```

Parameters that can follow TERM

**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 467 for details.

Example

*---------------------------------------------------------------------*
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:

```plaintext
//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 206. 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:
  ```plaintext
  CONTROL MVSID=MVS1
  LIST JOBNAME=ALL
  ```

- List, verify and delete a job entry:
  ```plaintext
  CONTROL MVSID=MVS1
  VERIFY JOBNAME=USRADRV2
  DELETE JOBNAME=USRADRV2
  LIST JOBNAME=USRADRV2
  ```

- Delete system linkage header and name services (with FORCE option):
  ```plaintext
  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 [CASE=LOWER] [CASE=UPPER] [LOCK=KEEP] [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. This is the default.

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. This is the default.

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 TOKEN,FORCE=NO◄◄
```

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. This is the default.

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.
Part 7. Troubleshooting

These topics provide technical reference information to help you to troubleshoot and diagnose problems with IMS Connect Extensions.

Topics:
- Chapter 21, “Troubleshooting connection failures,” on page 483
- Chapter 22, “Messages and codes,” on page 487
- Chapter 23, “Gathering diagnostic information,” on page 547
Chapter 21. 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 an IMS Connect Extensions console listener

If an attempt by a client to connect to the IMS Connect Extensions console listener on an IMS Connect system was unsuccessful, use the following information to help resolve the issue.

Message HWSP1445E unknown exit name with MSGID=*/CONSOL*

You receive a message HWSP1445E containing the text */CONSOL* in the MSGID parameter. For example:

HWSP1445E UNKNOWN EXIT NAME SPECIFIED IN MESSAGE PREFIX; MSGID=*/CONSOL*

The likely cause of this issue is that an IMS Connect Extensions client (ISPF, Operations Console, or REXX) is attempting to connect to IMS Connect Extensions on an IMS Connect message port instead of the IMS Connect Extensions console port. To remedy this situation, check that your client is configured to connect to the console port. The port you require is reported in message “CEX5020I” on page 492 when IMS Connect is started. To change the port your client application is using, refer to the following topics:

- For issues connecting via the ISPF Operations dialog, see "Defining IMS Connect systems" on page 320.
- For issues connecting via the Operations Console for z/OS Explorer, change your system details. See "Adding systems manually" on page 177.
- For issues connecting via the IMS Connect Extensions host command environment for REXX, see "CONNECT command" on page 431.

IMS Connect systems are being reported as inactive

One or more IMS Connect systems are being reported as inactive in the IPSF operations dialog (status of INACT), Operations Console for z/OS Explorer (a red square next to the system name), or there is a failure to connect to the system by the "CONNECT command" on page 431 in the IMS Connect Extensions host command environment for REXX. Error messages may also be displayed in the Console view of the Operations Console.

A client may fail to connect to IMS Connect for one or more of the following reasons:

- The specified system name is not the name used in the IMS Connect configuration member.
- The system is not active.
- Your client application is attempting to connect to IMS Connect Extensions on an IMS Connect message port instead of the IMS Connect Extensions console port. See message HWSP1445E.
The user ID or password specified in the connection details or in the details for this system in the Operations Console for z/OS Explorer is incorrect. See “Managing connection profiles” on page 180.

The number of sockets available in IMS Connect has been exhausted. See “IMS Connect reaches MAXSOC.”

**IMS Connect reaches MAXSOC**

The HWSCFGxx member of the IMS PROCLIB data set contains a MAXSOC parameter inside the TCPIP statement that sets the maximum total number of sockets that the associated instance of IMS Connect can open at once. The IMS Connect Extensions console listener port uses two sockets from the MAXSOC socket pool to establish two-way communication with client applications such as the Operations Console for z/OS Explorer and ISPF client. When the number of sockets open in IMS Connect reaches the MAXSOC limit, connection requests to IMS Connect and to the IMS Connect Extensions listener port will fail. As such, any attempt to connect with IMS Connect Extensions client will also fail.

To prevent this problem, it is recommended that you fine tune the MAXSOC, WARNSOC, and WARNINC parameters on the TCPIP statement.

The WARNSOC parameter specifies a warning level when the number of sockets increases to a certain percentage of the MAXSOC limit. When the number of sockets that are currently supported reaches this warning level, IMS Connect issues message HWSS0772W, which you can use to detect message flood conditions. After you receive the warning, you can begin your analysis using IMS Connect Extensions clients (whilst sockets are still available), or to query IMS Connect and trigger an automated response. By default, the WARNSOC parameter will issue the warning when the number of sockets is 80% of MAXSOC. For example, if MAXSOC=2000, the warning is issued when the number of sockets reaches 1600. You can give yourself more time to analyze the problem if you set the WARNSOC value as close to a typical peak as possible. If, for example, your typical peak period only occupies 800 sockets, setting WARNSOC to 50 would issue a warning when the number of sockets reaches 1000, giving you more time to respond.

The WARNINC parameter specifies a warning level incremental percentage. After the WARNSOC value is reached, every time the number of sockets increases by the percentage value specified on WARNINC, IMS Connect reissues warning message HWSS0772W.

For more information, see **TCPIP statement** in the IMS documentation.

**Related concepts:**

“Troubleshooting connection failures” on page 181

If you are having trouble connecting to z/OS or your IMS Connect systems, check for messages in the Common Services Library console view.

---

**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:
The command shell attempts to connect to an available IMS Connect message port to submit IMS type-1 commands using message exit HWSJA0 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.

<table>
<thead>
<tr>
<th><em>errno</em></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>111</td>
<td>Permission is denied.</td>
</tr>
<tr>
<td>1128</td>
<td>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.</td>
</tr>
</tbody>
</table>

**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 nnnn)
  ```

- Ensure that any modifications to message exit HWSJA0 do not impact client message flows.
Chapter 22. 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 RSCCOD
   Possible reason codes are shown in the following table.

Table 12. Reason codes generated by IMS Connect Extensions

<table>
<thead>
<tr>
<th>OMUSR reason code</th>
<th>RSM (decimal value)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SECFAIL</td>
<td>40</td>
<td>Security failure: access denied</td>
</tr>
<tr>
<td>DESTUNA</td>
<td>72</td>
<td>Transaction routing: destination unavailable</td>
</tr>
<tr>
<td>PACEMLT</td>
<td>150</td>
<td>Transaction pacing: pacing threshold exceeded</td>
</tr>
<tr>
<td>N/A</td>
<td>153</td>
<td>Message exit not active</td>
</tr>
<tr>
<td>SECINOP</td>
<td>154</td>
<td>Security failure: user ID, password, or group information missing or invalid</td>
</tr>
<tr>
<td>N/A</td>
<td>156</td>
<td>Message exit deleted</td>
</tr>
</tbody>
</table>

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:

\[ \text{mmmmcccc} \]

where:

\[ \text{mmmm} \text{ (the first four digits)} \]
   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:

CEXnnnx Message text with symbol information

The message identifier has the format CEXnnnx or FUNnnnx where:

CEX Identifies the message as an IMS Connect Extensions product message.
FUN Identifies the message as a Common Services Library functional support
message.
nnn 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.
(WCorresponds to a return code 0.)

W Warning. Possible problems occurred that you should evaluate.
(CCorresponds to a return code 4.)

F Failure. Your action may be required as processing is incomplete.
(CCorresponds to a return code 8.)

E Error. Your action required. Processing is incomplete.
(CCorresponds to a return code 12.)

S Severe. Internal logic error requiring your attention.
(CCorresponds 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
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 IMS Connect Extensions host command environment for REXX.

Related concepts:

- “Message Log” on page 217
- “Browsing the message log for an IMS Connect” on page 164

The message log for an IMS Connect system reveals activity within IMS Connect and IMS Connect Extensions. Use the message log to understand more about the status of archiving, to view error messages, or to display a history of commands that have been issued and their responses.

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-prefixes messages

This topic describes messages with the CEXnnnmx message identifier format.

<table>
<thead>
<tr>
<th>CEX5004I</th>
<th>IMS Connect Extensions status change detected: message</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong> Indicates that a change in the status of the IMS Connect Extensions system has been detected. message identifies what has changed:</td>
<td></td>
</tr>
<tr>
<td>• IMS Connect Extensions now Active</td>
<td></td>
</tr>
<tr>
<td>• IMS Connect Extensions now Inactive</td>
<td></td>
</tr>
<tr>
<td>• Event Collection now Active</td>
<td></td>
</tr>
<tr>
<td>• Event Collection now Inactive</td>
<td></td>
</tr>
<tr>
<td>• Advanced Features now Active</td>
<td></td>
</tr>
<tr>
<td>• Advanced Features now Inactive</td>
<td></td>
</tr>
<tr>
<td>• Severe Error has occurred</td>
<td></td>
</tr>
<tr>
<td>• Event Collection level changed to new from old</td>
<td></td>
</tr>
<tr>
<td>• Trace recording level changed to new from old</td>
<td></td>
</tr>
<tr>
<td><strong>System action:</strong> IMS Connect Extensions continues.</td>
<td></td>
</tr>
<tr>
<td><strong>User response:</strong> None. Informational message only.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CEX5005I</th>
<th>Refresh affinity list=aff_name a datastore entry referencing the affinity list for Supermember routing was not found</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong> A datastore entry could not be found that references the affinity list member as a Supermember routing list.</td>
<td></td>
</tr>
<tr>
<td><strong>System action:</strong> Processing continues.</td>
<td></td>
</tr>
<tr>
<td><strong>User response:</strong> None. Informational message only.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CEX5006E</th>
<th>ERROR DETECTED, RC=rc, RSN=rsn</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong> 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.</td>
<td></td>
</tr>
<tr>
<td><strong>System action:</strong> IMS Connect Extensions quiesces.</td>
<td></td>
</tr>
<tr>
<td><strong>User response:</strong> 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.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CEX5007E</th>
<th>PROCESSING ERROR. SEE CEXPRINT DATA SET</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong> 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.</td>
<td></td>
</tr>
<tr>
<td><strong>System action:</strong> IMS Connect Extensions quiesces.</td>
<td></td>
</tr>
<tr>
<td><strong>User response:</strong> 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.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CEX5008E</th>
<th>Event Collection Init failed, RSN=rsn</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation:</strong> 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.</td>
<td></td>
</tr>
<tr>
<td><strong>System action:</strong> IMS Connect Extensions quiesces.</td>
<td></td>
</tr>
</tbody>
</table>
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: None. Informational message only.

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.

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.

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.

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.

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.

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.

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.

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.

Object identifies the name of the object. The object is one of the following:
• Warning
• Reject
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.

**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 wto, reject threshold rto

**Explanation:** Indicates that IMS Connect Extensions has initialized and Pacing is active at the IMS Connect system level. *icnt* identifies the pacing interval count, *wto* the warning threshold and *rto* 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 min, max, IP address rules status

**Explanation:** Indicates that IMS Connect Extensions has initialized. The status and configuration
information for several features is reported:

- **status** identifies the status of a feature as being 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

**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 in the SYSIN DD.

**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.
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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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 REXX host command and batch command control statements

Explanation: Lists the batch command utility control statements in the SYSIN data set. It is also issued by the IMS Connect Extensions host command environment for REXX.

System action: Processing continues

User response: None. Informational message only.

Related reference:

"Example REXX for a single system ACEE cache statistics query" on page 229

This sample REXX exec shows how to connect to an IMS Connect system, run a query to obtain ACEE cache statistics, and then display the results.

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 PASSWORD=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:** A TCPIP error occurred.

- *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 your request is being directed to the IMS Connect Extensions console port, not an IMS Connect message port.
2. Using the error number (ERRNO), refer to z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference to diagnose the error.

**Related information:**

- ["CEX5020I"] on page 492
- IMS Connect Extensions initialized and status, console port is port

---

**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:**

- *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.

CEX5061I  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.

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= dbname correct and refresh the affinity list then refresh the datastore.

System action: Processing continues.

User response: None. Informational message only.

<table>
<thead>
<tr>
<th>CEX5073E</th>
<th>error information</th>
</tr>
</thead>
</table>

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.

<table>
<thead>
<tr>
<th>CEX5074I</th>
<th>Message translation activation commencing for exit xname using client CCSID did and server CCSID svid</th>
</tr>
</thead>
</table>

Explanation: IMS Connect Extensions is attempting to initialize message translation.

System action: Processing continues.

User response: None. Informational message only.

<table>
<thead>
<tr>
<th>CEX5075I</th>
<th>CEX trace is active inactive, level=level, tracing by tracing_options</th>
</tr>
</thead>
</table>

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.

<table>
<thead>
<tr>
<th>CEX5076I</th>
<th>Message translation activation successful for exit xname</th>
</tr>
</thead>
</table>

Explanation: Message translation activation successful.

System action: Processing continues.

User response: None. Informational message only.

<table>
<thead>
<tr>
<th>CEX5077E</th>
<th>Message translation activation failed for exit xname, unable to detect a supported translation technique in the exit RSN=rsn</th>
</tr>
</thead>
</table>

Explanation: IMS Connect Extensions was unable to activate message translation for 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.

<table>
<thead>
<tr>
<th>CEX5078E</th>
<th>Message translation activation failed for exit xname, client-server CCSID pairing not defined to z/OS Unicode Services</th>
</tr>
</thead>
</table>

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.

<table>
<thead>
<tr>
<th>CEX5079E</th>
<th>Message translation activation failed for exit xname, client-server CCSID pairing not an ASCII-to-EBCDIC conversion</th>
</tr>
</thead>
</table>

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.

<table>
<thead>
<tr>
<th>CEX5080E</th>
<th>Message translation activation failed for exit xname, client-server CCSID pairing not a single byte character set (SBCS) conversion</th>
</tr>
</thead>
</table>

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 23, “Gathering diagnostic information,” on page 547, 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.

---

CEX5084E  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 | from the repository, member is in use, RCsrc, RSNsrc, 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 tracing 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.

Related reference: [Reviewing the active OTMA routing rules in IMS Connect](#)

When a system starts that has OTMA routing rules activated, a series of messages is written to CEXPRINT. These messages display the OTMA routing plan that is currently in effect (if any) and provides summarized information about the OTMA routing rules, OTMA routing lists, and IMS data stores that may be in use.

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 list 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.

Related reference: [Reviewing the active OTMA routing rules in IMS Connect](#)

When a system starts that has OTMA routing rules activated, a series of messages is written to CEXPRINT. These messages display the OTMA routing plan that is currently in effect (if any) and provides summarized information about the OTMA routing rules, OTMA routing lists, and IMS data stores that may be in use.

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.

Related reference:
**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 ds**

**Explanation:** IMS Connect Extensions found the Datastore Group ds 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.
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.

---

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.

---

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.

---

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.

---

EXplanation: A safe cross-memory post to a Publisher API client to notify it of an ASYNC event has failed. The likely reason for this message is that the given Publisher API client has failed in such a way that it was unable to notify the Publisher that it was in emergency termination, for example, abend at end-of-memory. In such a case “Return code=4 Reason code=0000401” would be expected, which informs you that a poorly behaved Publisher client terminated earlier and the Publisher has only just become aware of this failed client via a safe cross-memory post failure.

**THD** The client thread number

**CLN** The client ID

**RC** The return code from the IEAMXXMP service

**RSN** The reason code from the IEAMXXMP service

**System action:** Processing continues.

**User response:** Review the client application to identify the reason for the failure and correct the error. Consider running the ECSA cleanup utility to reclaim resources that may otherwise remain outstanding.

**Related reference:**

[CESA cleanup utility” on page 477](#)

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.

---

EXplanation: IMS Connect Extensions found the Affinity resource *name* *type* *type* not found on this system.

**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.

---

EXplanation: When PassTicket generation is requested, the USERID 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 Autosuspend was not 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 PM70277.

---

**CEX5126E** Password validation failed for user user, SAF return codes=safrc racfrc rafrsn

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 rafrsn 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 userid, SAF return code=safcr

Explanation: A request was made to validate access to IMS Connect system safnm using security class safcl for user ID userid. The security system rejected the request.

safcr is the SAF return code from the SAF FASTAUTH call. racfr / 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 23, “Gathering diagnostic information,” on page 547, 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 RCVPTPTGN 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 CEXPRIN.

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 RLST(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, SusCWZer, 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 blank. 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.

**Related reference:**

> [Reviewing the active ODBM routing rules in IMS Connect](#) on page 297

When a system starts that has ODBM routing rules activated, a series of messages is written to CEXPRINT. These messages display the ODBM routing plan that is currently in effect (if any) and provides summarized information about the ODBM routing rules, ODBM routing lists, and ODBM targets that may be in use.

---

**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.

**Related reference:**

---

**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. 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.
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.

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.

exit detected in HWS configuration file TCP/IP command. Not allowed. Results unpredictable.

**Explanation:** During initialization, CEXROUT0 or HWSROUT0 was detected in the HWS configuration file TCP/IP 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 TCP/IP commands and restart IMS Connect.

---

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.

---

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.
**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 • CEX5415E

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.


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 484.

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 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.
CEX551E  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: Vversion

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:

User response: Refer to the messages in CEXPRINT (data set referred to by that ddname).
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: err

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=exp, RSN=exp

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.
CEX5531I  SAF RC=src, SAF RSN=rsrn

Explanation: The contents of the RSM extension are formatted in three parts using the messages CEX5531I, 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.

CEX5532I  Len=ln, APAR=ap, flag=flag, content
flag=cont, #message segments=msgc, 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 Msg=msg

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 RC=src, SAF RSN=rsrn

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.

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.

CEX5537I  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 at the start of 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 configuration 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 contain 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 23, “Gathering
              diagnostic information,” on page 547, 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 linkededit 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.

CEX5572I  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.

CEX5572W  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.

CEX5573E  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.

CEX5574E  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 0 is set and processing is
completed.
User response: None.

CEX5575I  IMS Connect has closed the session, RSN=rssh

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
(CEXBDS00) 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=rssn

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= 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 data store credentials are still cached from when the datastore was originally added to IMS Connect, they are used. If the cached data store credentials are not available, the details supplied in the ADD 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 data store credentials were not used, the credentials shown in some IMS Connect Extensions displays and event records might be incomplete. Fully define the data store 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= 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 IWM4CRE (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 Rm - Rn 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.
Job TCB time tcb-time Enclave zIIP time ziiptime with zIIP offload active | inactive

Explanation: This message reports processor usage times. Times are in the format hh:mm:ss.xxxxx

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:
CEX5600E  COULD NOT UPDATE THE DATASTORE | ODBM TARGET | SYSTEM MEMBER = MEMBER; LMM ERROR. 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.

System action: The command fails. 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 = MEMBER; 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.

CEX5609E  Error at line nnn: the IP_ADDRESS address is errortype

Explanation: The IPv4 or IPv6 address value given at the specified line number in the SYSIN data set is either invalid, in an unsupported format, or possibly both.

Limited masking using wildcard character "*" to represent "all trailing node values" is supported. For example, 1.1.1.* is valid, but 1.1.*.1 is not. If trailing node wildcard masking is used then the * IPv6 short-hand form "::" is not supported.

Use of IPv4 mapped IPv6 addresses written with a mixed IPv6 format combined with the last 32 bits written in the IPv4 dot notation is not supported.

DNS names are not supported.

System action: Batch input utility execution ends.
User response: Correct the value and try again.

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 command" on page 434.

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=clientID 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=clientID 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 336 for information about installing and maintaining user exits.

**Related concepts:**

["Pre-routing user exit" on page 269]

You can use the pre-routing exit to influence the selection of IMS data stores as candidates for OTMA rules-based routing.

---

**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 ADMIN_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.

Explaination: 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.

CEX5630W  The IP Address Rule from member=member was superseded by member=member for IP address=address

Explaination: This message indicates that when building the IP address rule block the block was updated by more than one member with 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. This condition is not normally expected but is permitted.

System action: Processing continues.

User response: If it was not intentional for one rule to supersede another, make corrections to your IP address rules.

CEX5631F  IP address rule descriptions

Explaination: This message is used to list the IP address rules that would be in effect for this IMS Connect system.

System action: Processing continues.

User response: None. Informational message only.

Related tasks: Creating workload rules for specific IP addresses” on page 305

CEX6000E  UNKNOWN MESSAGE CODE: msg

Explaination: 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

Explaination: 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.

CEX6002E  BAD PARAMETER: parm

Explaination: 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.

CEX6004E  SERVICES INITIALIZATION ERROR, RC=rc, RSN=rsn

Explaination: 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

Explaination: 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.

Chapter 22. Messages and codes  525
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 DLIST option must be provided with the cmd_name command
Explanation: The ROUTE and QUERY PENDING_RESPONSES host commands require either a DATASTORE or DLIST option.
System action: Processing of the CEX REXX host command stops.
User response: Reissue the command using either the DATASTORE or DLIST option.

CEX6022E The DATASTORE and DLIST options are mutually exclusive on the cmd_name command
Explanation: The ROUTE and QUERY PENDING_RESPONSES host commands require either a DATASTORE or DLIST option.
System action: Processing of the CEX REXX host command stops.
User response: Reissue the command using either the DATASTORE or DLIST 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. T0KEN** 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.

---

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.

---

CEX6036I  HOST defaults to LOCALHOST

Explanation: None of the keywords HOST, DNSNAME, IPV4ADR, or IPV6ADR were specified. LOCALHOST will be used.

System action: Processing continues.

User response: None. Informational message only.

---

CEX6037W  Target IMS Connect Extensions server has commands inactive

Explanation: The only command which can be sent to Console services on this connection is QUERY.

System action: Processing continues.

User response: To send additional commands, activate commands in the IMS Connect system definition.
CEX6038W  Target IMS Connect Extensions server has access control active and the user has no valid authenticator.
Explanation: The only commands which can be sent to Console services on this connection are QUERY and SHELL.
System action: Processing continues.
User response: Connect with a username and password or use a PassTicket to send additional commands.

CEX6039I  Target IMS Connect Extensions server has access control inactive.
Explanation: The password provided on the CONNECT command will not be verified by the target IMS Connect Extensions server.
System action: Processing continues.
User response: None. Informational message only.

CEX6040W  TRACE command has no effect.
Explanation: No options were specified on the TRACE command.
System action: Processing continues.
User response: Specify TRACE options and retry the request.

CEX6041W  TRACE activation is implied, LEVEL defaults to 1.
Explanation: PORT or COND was specified but no LEVEL was specified, TRACE was in an inactive state, so the default LEVEL=1 was used.
System action: Processing continues.
User response: If you did not intend to activate the trace, use LEVEL=0 to deactivate it.

CEX7000E  Unknown error: description: function:
rc=decimal_rc(hex_rc)
Explanation: The request to stream an IMS Connect Extensions feed via TCP/IP has failed. The reason for the failure is unknown. The description, function, and return code are from the point of failure, and can help to identify the reason for the failure.
System action: The feed job stops.
User response: Contact IBM Software Support.

CEX7001E  System error: description: function:
rc=decimal_rc(hex_rc)
Explanation: The request to stream an IMS Connect Extensions feed via TCP/IP has failed while performing a POSIX system function. The return code is from that function. The description matches the return code.
System action: The feed job stops.
User response: Contact IBM Software Support.

CEX7002E  SSL/TLS error: description: function:
rc=decimal_rc(hex_rc)
Explanation: The request to stream an IMS Connect Extensions feed via TCP/IP has failed while performing a GSKit SSL/TLS function. The return code is from that function. The description matches the return code.
System action: The feed job stops.
User response: Contact IBM Software Support.

CEX7003E  Operation not permitted: description: function:
rc=decimal_rc(hex_rc)
Explanation: The request to stream an IMS Connect Extensions feed via TCP/IP has failed for one of the following reasons:
- The operation is not possible, perhaps due to temporary conditions. For example, no spare ports are currently available for network connections.
- The current user does not have permission to perform the operation.

The return code is from the function that attempted to perform the operation. The description matches the return code.
System action: The feed job stops.
User response: Use the description, function name, and return code to diagnose the reason for the failure.

CEX7004E  Connection could not be established: description: function:
rc=decimal_rc(hex_rc)
Explanation: The request to stream an IMS Connect Extensions feed via TCP/IP has failed because a connection was refused or the host was unreachable.
The return code is from the function that attempted to establish the connection. The description matches the return code.
System action: The feed job stops.
User response: Use the description, function name, and return code to diagnose the reason for the failure.
CEX7005E  Operation timed out: description: function: 
   rc=decimal_rc(hex_rc)

Explanation: The request to stream an IMS Connect Extensions feed via TCP/IP has failed because an operation timed out.

The return code is from the function that attempted to perform the operation. The description matches the return code.

System action: The feed job stops.

User response: Consider using the TIMEOUT feed configuration parameter to specify a longer timeout. Otherwise, contact your system network support.

CEX7006E  Connection lost: description: function: 
   rc=decimal_rc(hex_rc)

Explanation: The request to stream an IMS Connect Extensions feed via TCP/IP has failed because the connection was closed by the peer or dropped.

The return code is from the function that detected the lost connection. The description matches the return code.

System action: The feed job stops.

User response: Contact your system network support.

CEX7007E  Key ring password error: description: function: 
   rc=decimal_rc(hex_rc)

Explanation: The request to stream an IMS Connect Extensions feed via TCP/IP with SSL/TLS has failed because the key ring password was missing, wrong, or expired.

The return code is from the function that detected the error. The description matches the return code.

System action: The feed job stops.

User response: Use the STASH or PASSWORD feed configuration parameter to specify the correct password.

CEX7008E  Error opening key ring: description: function: 
   rc=decimal_rc(hex_rc)

Explanation: The request to stream an IMS Connect Extensions feed via TCP/IP with SSL/TLS has failed because an I/O or formatting error occurred opening the key ring.

The return code is from the function that detected the error. The description matches the return code.

System action: The feed job stops.

User response: Contact your system security administrator.

CEX7009E  Remote host's certificate could not be validated: description: function: 
   rc=decimal_rc(hex_rc)

Explanation: The request to stream an IMS Connect Extensions feed via TCP/IP with SSL/TLS has failed because the remote host's certificate could not be validated. Possible reasons include: the certificate could be self-signed, revoked, or have an unknown certificate authority (CA).

The return code is from the function that detected the error. The description matches the return code.

System action: The feed job stops.

User response: Contact your system security administrator.

CEX7010E  Remote host unsupported: description: function: 
   rc=decimal_rc(hex_rc)

Explanation: The request to stream an IMS Connect Extensions feed via TCP/IP with SSL/TLS has failed because the remote host performed an action that is not supported.

The return code is from the function that detected the error. The description matches the return code.

System action: The feed job stops.

User response: Contact your system security administrator.

CEX7010E  KeyRING is a required parameter when SECURITY has been specified

Explanation: The IMS Connect Extensions feed specified a SECURITY parameter, but not a KEYRING parameter.

System action: The feed job stops.

User response: For secure TCP, insert the missing KEYRING parameter. Otherwise, for unsecure TCP, remove the SECURITY parameter. Resubmit the job.

CEX7110W  Field not included in JSON.
   FIELD=field_name

Explanation: The FIELDS configuration parameter of the IMS Connect Extensions feed specified an unsupported field name.

System action: The feed job continues.

User response: Correct the FIELDS parameter value.

CEX7111E  Error opening SYSPRINT.
   detailed_error_message

Explanation: The IMS Connect Extensions feed job could not open the required SYSPRINT output data set.

System action: The feed job stops.
User response: Edit the feed JCL to specify a correct DD statement for SYSPRINT. Resubmit the job.

CEX712W  Socket connection to host host has been lost. A reconnection will be attempted in poll_time seconds.

Explanation: The IMS Connect Extensions feed lost its TCP/IP socket connection to the destination host. poll_time is the value of the POLL feed parameter.

System action: The feed job continues.

User response: Check the status of the destination host. If the destination host is active and listening on the specified port, then check the status of the network connection between the destination host and the system on which the feed job is running.

CEX7113E  Virtual storage exhausted.

Explanation: The IMS Connect Extensions feed exceeded its virtual storage limit.

System action: The feed job stops.

User response: Review the region size and increase it if necessary.

CEX714E  Error writing to SYSPRINT.

detailed_error_message

Explanation: The IMS Connect Extensions feed job could not write to the SYSPRINT output data set.

System action: The feed job stops.

User response: Review the DD statement for SYSPRINT in the feed JCL.

CEX7201I  Monitoring has started; Client=jobname and jobid AuthID=userid

Explanation: The IMS Connect Extensions feed job has connected to the IMS Connect system and is waiting for events to process. The feed job has connected to IMS Connect using the reported client name and RACF authorization ID.

System action: The feed job continues.

User response: None. Informational message only.

CEX7202I  Monitoring has stopped; CA20s=n Records=n READASYN=n

Explanation: The IMS Connect Extensions feed job has stopped monitoring the IMS Connect system and has disconnected from the IMS Connect system. The message reports the following statistics for the entire monitoring period:

CA20s=n

The number of IMS Connect transaction index records forwarded by the feed.

Records=n

The number of IMS Connect Extensions event records received and processed.

READASYN=n

The number of IMS Connect Extensions publisher API READASYN requests made to receive event buffers.

System action: The feed job stops monitoring the IMS Connect system. The feed job continues.

User response: None. Informational message only.

CEX7204I  Writing of CA20 records to SMF has commenced; Type=n SubType=n

Explanation: The IMS Connect Extensions feed job has successfully written the first IMS Connect transaction index record (also known as log code CA20) to SMF. The message reports the SMF record type and subtype specified by the TYPE and SUBTYPE feed parameters.

System action: The feed job continues.

User response: None. Informational message only.

CEX7205E  SMF record write failed;

MACRO=SMFWTM R15=hex_return_code

Explanation: The IMS Connect Extensions feed job called the SMFWTM macro to write an IMS Connect transaction index record to SMF. However, the macro failed with the reported hexadecimal return code in register 15 (R15).

The SMFWTM macro is supplied with z/OS. For descriptions of SMFWTM return codes, see the z/OS MVS SMF macros documentation in IBM Knowledge Center.
CEX7206E • CEX7212E

System action: The feed job stops.
User response: The appropriate response depends on the hexadecimal return code:
- 08, 38 Malformed SMF record. Contact IBM Software Support.
- 10, 18, 28, 2C, 34, 3C System problem. Check the status of SMF.

CEX7206E CEXPAPI REQUEST=api_function_name
ended with R15=hex_return_code
R0=hex_reason_code REASON=description
FEEDBACK=hex_feedback_code

Explanation: The IMS Connect Extensions feed job called an IMS Connect Extensions publisher API (CEXPAPI) function. However, the API request ended with the reported nonzero hexadecimal return code in register 15 (R15) and reason code in register 0 (R0).
System action: The feed job stops.
User response: For a more detailed description of the reason code, and a description of the feedback code, see macro CEXAPEQU in the IMS Connect Extensions SCEXMACS library. In the macro, reason codes have the prefix "IIRSN_", feedback codes have the prefix "IIFDB_".

CEX7207W CEXPAPI CONNECT TRM ECB has been posted; monitoring is being shut down with an ETERM

Explanation: The IMS Connect Extensions publisher API server is shutting down and has asked all clients, including this IMS Connect Extensions feed job, to stop monitoring.
System action: The feed job stops monitoring the IMS Connect system. The feed job continues.
The feed job polls the publisher API server at regular intervals, as specified by the POLL feed parameter. If the publisher API server restarts, the feed job resumes monitoring the IMS Connect system.
User response: If you expected the publisher API server to shut down, then this message is informational only, no response required. Otherwise, consider investigating why the publisher API server shut down.

CEX7208E ABEND=abend_code detected; monitoring is being shut down with an ETERM

Explanation: An abend occurred while the IMS Connect Extensions feed job was monitoring the IMS Connect system.
System action: The feed job stops with an emergency termination.
User response: Contact IBM Software Support.

CEX7209E IMSA error processing Connect events;
R15=hex_return_code

Explanation: The IMS Performance Analyzer routine failed while processing an event record from IMS Connect Extensions. The routine failed with the reported hexadecimal return code in register 15 (R15).
System action: The IMS Connect Extensions feed job stops.
User response: Contact IBM Software Support.

CEX7210I New archive data set was allocated;
DSN=data_set_name

Explanation: The IMS Connect Extensions feed job allocated a new data set for archiving IMS Connect transaction index records.
System action: The feed job continues.
User response: None. Informational message only.

CEX7211I Archive data set was closed; record_count records written to DSN=data_set_name

Explanation: The IMS Connect Extensions feed job closed the archive data set for one of the following reasons:
• The data set is full. The feed job is switching to a new data set.
• The feed job is stopping.
The message reports the number of records written to the data set.
The data set is now available for post-processing.
System action: The feed job switches to a new data set or stops.
User response: None. Informational message only.

CEX7212E New archive data set allocation has failed; ARCHDSN=data_set_name_pattern

Explanation: The IMS Connect Extensions feed job attempted but failed to dynamically allocate a new archive data set.
The message reports the data set name pattern, which might include substitution variable names.
System action: The feed job stops.
User response: See the subsequent CEX7213E message for the same IMS Connect system. That message contains details of the dynamic allocation error, including the actual data set name.
CEX7213E  DYNALLOC error  
EC=hex_error_reason_code  
IC=hex_information_reason_code  
DSN=data_set_name  

Explanation: The IMS Connect Extensions feed job attempted but failed to dynamically allocate a new archive data set.  
System action: The feed job stops.  
User response: See the JES log (JESMSGLG) output data set of the feed job, which might contain messages that explain this error. Otherwise, see the description of the system completion code and information reason code from the DYNALLOC macro in the z/OS MVS documentation in IBM Knowledge Center.

CEX7214E  Archive data set error; ABEND=Sxxx-cc  
DD=dname DSN=data_set_name  

Explanation: The IMS Connect Extensions feed job attempted to open or write to the archive data set. The attempt abended with the reported hexadecimal system completion code (xxx) and reason code (cc).  
System action: The feed job stops.  
User response: See the JES log (JESMSGLG) output data set of the feed job, which might contain messages that explain this error. Otherwise, see the description of the system completion code in the z/OS MVS documentation in IBM Knowledge Center.

CEX7220E  Configuration error on line line_number at "token"  

Explanation: The SYSIN input data set of the IMS Connect Extensions feed job contains a syntax error, "token", on the reported line number. The SYSIN data set contains the feed configuration parameters.  
System action: The feed job stops.  
User response: Correct the syntax and then resubmit the job.

CEX7221E  Configuration file missing; DDNAME=SYSIN  

Explanation: The IMS Connect Extensions feed job is missing the required SYSIN input data set containing feed configuration parameters.  
System action: The feed job stops.  
User response: Specify a SYSIN DD statement in the JCL, specify feed configuration parameters in that data set, and then resubmit the job.

CEX7222E  Configuration error. Required option not specified: missing_parameter  

Explanation: The SYSIN input data set of the IMS Connect Extensions feed job is missing a required feed configuration parameter.  
System action: The feed job stops.  
User response: Specify the missing configuration parameter, and then resubmit the job.

CEX7223W  Monitoring has stopped; CA20s=Records=LostData=LostRecords=  

Explanation: The IMS Connect Extensions feed job has stopped monitoring the IMS Connect system and has disconnected from the IMS Connect system.  
Data has been lost. IMS Connect Extensions could not keep up with the volume of events being generated.  
The message reports the following statistics for the entire monitoring period:  
CA20s=  
The number of IMS Connect transaction index records forwarded by the feed.  
Records=  
The number of IMS Connect Extensions event records received and processed.  
READASYN=  
The number of IMS Connect Extensions publisher API READASYN requests made to receive event buffers.  
LostData=  
The number of READASYN requests that lost data.  
LostRecords=  
The number of events that were lost across all READASYN requests.  
System action: The feed job stops monitoring the IMS Connect system. The feed job continues.  
User response: Adjust the values of the BUFSIZE and THRESHOLD feed parameters to prevent data loss.

CEX7224W  CEXPAPI status: CA20s=Records=READASYN=LostData=LostRecords=  

Explanation: This message reports interval statistics for an IMS Connect system that is being monitored by the IMS Connect Extensions feed job.  
Data has been lost. IMS Connect Extensions could not keep up with the volume of events being generated.  
The message reports the following statistics for the time interval:
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.

Related reference:
This sample REXX exec shows how to connect to an IMS Connect system, run a query to obtain ACEE cache statistics, and then display the results.

**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.

**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 posit at stick
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. Values: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: Correct the value to conform to the expected data type 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: Edit the input 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/racfrs 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/racfrs 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.

---

FUN2509W Journal data set DEVTYPE=x'dt' unknown for DSN=dsn

Explanation: The Journal Manager has started to use the named journal data set. In order to select an optimal BLKSIZE to open the data set the Journal Manager reviewed the device type associated with this data set and found a value that is not currently supported. BLKSIZE=32760 will be used to open the data set.

System action: Processing continues.

User response: If this data set is not extended-format, then the Journal Manager can efficiently utilize the given data set and this message can be treated as informational. However, if it is an extended-format data set, then the use of BLKSIZE=32760 is likely to be suboptimal in terms of data set utilization and you should consider relocating the data set. Either way, if the DEVTYPE represents a DASD device currently supported by IBM, then notify the IBM support team.

---

FUN2510E Journal Manager is recovering from a D37 abend for DSN=dsn

Explanation: The Journal Manager has captured a D37 abend for the given journal data set and is attempting to recover from this event.

System action: The Journal Manager will attempt to recover from the event by suppressing the abend, performing a journal data set switch, issuing diagnostic messages and then continuing.

User response: Contact IBM Software Support. This message is part of extended serviceability support and should be reported to IBM. Message "FUN2711I" on page 540 contains additional diagnostic information that may be used by IBM support for problem diagnosis.

---

FUN2511I Journal data set. dm active journal(s) processed

Explanation: The Archive Manager has completed and archived 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.

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:

V10  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 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 happened 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.

FUN2584W 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.
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.

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.

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.

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.

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.

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 db-name, failure-type=r=return-code, h=handle-address, f.flag-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.

**FUN2709I** Jobname=jobname, Comp Code=task_code, 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** Rn - Rn xxxxxxx xxxxxxx xxxxxxx xxxxxxx

**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, xxxxxxxxx xxxxxxxxx xxxxxxxxx

**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 23. 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:
  - CE XDIAG
    - IMS Connect Extensions diagnostic information
  - CEXITRINT
    - 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 SYSTPR, include the OPTION MSGVLVL=VERBOSE statement in the exec after initializing the IMS Connect Extensions host command environment:

  parse arg PARM
  address LINK "CEXRXENV INIT"
  address CEX "OPTION MSGVLVL=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 24, “Events recorded by IMS Connect Extensions,” on page 551
- Chapter 25, “Mapping event records,” on page 559
- Chapter 26, “Control input data set options,” on page 561
- Chapter 27, “Reporting utilities,” on page 571
- Chapter 28, “Archiving utilities,” on page 585
- Chapter 29, “Installation verification programs (IVP),” on page 591
- Chapter 30, “Client services exit,” on page 597
- Chapter 31, “Legacy OTMA routing methods,” on page 607
- Chapter 32, “How to read syntax diagrams,” on page 621
Chapter 24. Events 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. To browse the contents of event records, use IBM IMS Problem Investigator for z/OS. See "Browse events recorded in the IMS Connect Extensions journal" on page 73. For in-depth reporting on IMS and IMS Connect performance, use IBM IMS Performance Analyzer for z/OS. See "Reporting and analysis with IMS Performance Analyzer" on page 82.

**Event flow**

The following diagrams demonstrates the flow of events for a send-then-commit (commit mode 1) transaction with sync level confirm. The first diagram shows the IMS request message (IRM) received by IMS Connect, which is then forwarded to IMS. IMS processes the request and forwards a response back to IMS Connect, which then returns it to the client.

IMS has sent a response to the client, but it is waiting for an acknowledgment from the client before committing changes to the database.

**Tip:** This wait time is a significant performance factor, because IMS holds syncpoint processing, and prevents the message processing region from processing another transaction, until it receives the acknowledgment from the client. Use IMS Problem Investigator or IBM Transaction Analysis Workbench to determine the length of this delay. To avoid delays caused by waiting for client acknowledgments, you might need more message processing regions. For more
information, see “Merge IMS and IMS Connect log files” on page 78.
The following diagram shows a successful database commit initiated by the client's acknowledgment message.

![Diagram showing successful database commit]

To see other types of event flow, see “Sample event flows in IMS Connect” on page 58.

**Connect status event records**

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.

The following table lists all single event records. The IMS Connect Extensions collection level determines whether the event record is written to the IMS Connect Extensions journal.

<table>
<thead>
<tr>
<th>Code (Hex)</th>
<th>Code (Dec)</th>
<th>Event description</th>
<th>Collection level</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>00</td>
<td>Control record</td>
<td>0</td>
</tr>
<tr>
<td>01</td>
<td>01</td>
<td>IMS Connect region initialization</td>
<td>0</td>
</tr>
<tr>
<td>02</td>
<td>02</td>
<td>IMS Connect region has completed termination</td>
<td>0</td>
</tr>
<tr>
<td>03</td>
<td>03</td>
<td>A support task (TCB) has been created</td>
<td>1</td>
</tr>
<tr>
<td>04</td>
<td>04</td>
<td>A support task (TCB) is terminating</td>
<td>1</td>
</tr>
<tr>
<td>05</td>
<td>05</td>
<td>Begin INIT API</td>
<td>4</td>
</tr>
<tr>
<td>06</td>
<td>06</td>
<td>End INIT API</td>
<td>4</td>
</tr>
<tr>
<td>07</td>
<td>07</td>
<td>Begin Bind Socket</td>
<td>4</td>
</tr>
<tr>
<td>08</td>
<td>08</td>
<td>End Bind Socket</td>
<td>4</td>
</tr>
</tbody>
</table>
Table 13. Connect status event records (continued)

<table>
<thead>
<tr>
<th>Code (Hex)</th>
<th>Code (Dec)</th>
<th>Event description</th>
<th>Collection level</th>
</tr>
</thead>
<tbody>
<tr>
<td>09</td>
<td>09</td>
<td>Listen on socket</td>
<td>4</td>
</tr>
<tr>
<td>0A</td>
<td>10</td>
<td>Begin Accept Socket</td>
<td>4</td>
</tr>
<tr>
<td>0B</td>
<td>11</td>
<td>End Accept Socket</td>
<td>3</td>
</tr>
<tr>
<td>0E</td>
<td>14</td>
<td>Begin Message Exit INIT</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>16</td>
<td>IMS data store becomes available</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>17</td>
<td>IMS data store becomes unavailable</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>18</td>
<td>An IMS TMEMBER joins the XCF group</td>
<td>0</td>
</tr>
<tr>
<td>13</td>
<td>19</td>
<td>An IMS TMEMBER leaves the XCF group</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>20</td>
<td>Begin SCI Registration</td>
<td>4</td>
</tr>
<tr>
<td>15</td>
<td>21</td>
<td>End SCI Registration</td>
<td>1</td>
</tr>
<tr>
<td>16</td>
<td>22</td>
<td>Begin SCI De-registration</td>
<td>4</td>
</tr>
<tr>
<td>17</td>
<td>23</td>
<td>End SCI De-registration</td>
<td>1</td>
</tr>
<tr>
<td>18</td>
<td>24</td>
<td>Recorder trace DCB has been opened</td>
<td>1</td>
</tr>
<tr>
<td>19</td>
<td>25</td>
<td>Recorder trace DCB pre-close</td>
<td>1</td>
</tr>
<tr>
<td>1A</td>
<td>26</td>
<td>User message exit return from INIT</td>
<td>1</td>
</tr>
<tr>
<td>1B</td>
<td>27</td>
<td>User message exit return from TERM</td>
<td>1</td>
</tr>
<tr>
<td>1C</td>
<td>28</td>
<td>Begin Secure Environment Open</td>
<td>1</td>
</tr>
<tr>
<td>1D</td>
<td>29</td>
<td>End Secure Environment Open</td>
<td>1</td>
</tr>
<tr>
<td>20</td>
<td>32</td>
<td>Begin Secure Environment Close</td>
<td>1</td>
</tr>
<tr>
<td>21</td>
<td>33</td>
<td>End Secure Environment Close</td>
<td>1</td>
</tr>
<tr>
<td>22</td>
<td>34</td>
<td>Begin Local Port Setup</td>
<td>3</td>
</tr>
<tr>
<td>23</td>
<td>35</td>
<td>End Local Port Setup</td>
<td>3</td>
</tr>
<tr>
<td>24</td>
<td>36</td>
<td>Begin RRS Connect</td>
<td>1</td>
</tr>
<tr>
<td>25</td>
<td>37</td>
<td>End RRS Connect</td>
<td>1</td>
</tr>
<tr>
<td>26</td>
<td>38</td>
<td>List In-doubt Context</td>
<td>1</td>
</tr>
<tr>
<td>27</td>
<td>39</td>
<td>Begin RRS Disconnect</td>
<td>1</td>
</tr>
<tr>
<td>28</td>
<td>40</td>
<td>End RRS Disconnect</td>
<td>1</td>
</tr>
<tr>
<td>29</td>
<td>41</td>
<td>Begin ODBM Registration</td>
<td>1</td>
</tr>
<tr>
<td>2A</td>
<td>42</td>
<td>End ODBM Registration</td>
<td>1</td>
</tr>
<tr>
<td>2B</td>
<td>43</td>
<td>Begin ODBM De-registration</td>
<td>1</td>
</tr>
<tr>
<td>2C</td>
<td>44</td>
<td>End ODBM De-registration</td>
<td>1</td>
</tr>
<tr>
<td>2D</td>
<td>45</td>
<td>Datastore Status Update</td>
<td>1</td>
</tr>
<tr>
<td>2E</td>
<td>46</td>
<td>Return From Port Exit INIT Call</td>
<td>1</td>
</tr>
<tr>
<td>2F</td>
<td>47</td>
<td>Return From Port Exit TERM Call</td>
<td>1</td>
</tr>
<tr>
<td>30</td>
<td>48</td>
<td>Begin ODBM Routing Exit INIT</td>
<td>1</td>
</tr>
<tr>
<td>31</td>
<td>49</td>
<td>End ODBM Routing Exit INIT</td>
<td>1</td>
</tr>
<tr>
<td>32</td>
<td>50</td>
<td>Begin ODBM Routing Exit TERM</td>
<td>1</td>
</tr>
<tr>
<td>33</td>
<td>51</td>
<td>End ODBM Routing Exit TERM</td>
<td>1</td>
</tr>
<tr>
<td>34</td>
<td>52</td>
<td>XML Adapter INIT Call Begin</td>
<td>1</td>
</tr>
</tbody>
</table>
Table 13. Connect status event records (continued)

<table>
<thead>
<tr>
<th>Code (Hex)</th>
<th>Code (Dec)</th>
<th>Event description</th>
<th>Collection level</th>
</tr>
</thead>
<tbody>
<tr>
<td>35</td>
<td>53</td>
<td>XML Adapter INIT Call End</td>
<td>1</td>
</tr>
<tr>
<td>36</td>
<td>54</td>
<td>XML Adapter TERM Call Begin</td>
<td>1</td>
</tr>
<tr>
<td>37</td>
<td>55</td>
<td>XML Adapter TERM Call End</td>
<td>1</td>
</tr>
<tr>
<td>38</td>
<td>56</td>
<td>OM Registration</td>
<td>1</td>
</tr>
<tr>
<td>39</td>
<td>57</td>
<td>OM Deregistration</td>
<td>1</td>
</tr>
<tr>
<td>A7</td>
<td>167</td>
<td>Internal Command Event</td>
<td>0</td>
</tr>
</tbody>
</table>

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.

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.

Table 14. Workload-related event records

<table>
<thead>
<tr>
<th>Code (Hex)</th>
<th>Code (Dec)</th>
<th>Event description</th>
<th>Collection level</th>
</tr>
</thead>
<tbody>
<tr>
<td>0C</td>
<td>12</td>
<td>Begin Close Socket</td>
<td>4</td>
</tr>
<tr>
<td>0D</td>
<td>13</td>
<td>End Close Socket</td>
<td>3</td>
</tr>
<tr>
<td>3C</td>
<td>60</td>
<td>Prepare for socket read</td>
<td>2</td>
</tr>
<tr>
<td>3D</td>
<td>61</td>
<td>User message exit entered for READ, XMIT, EXER, or RXML</td>
<td>2</td>
</tr>
<tr>
<td>3E</td>
<td>62</td>
<td>User message exit return for READ, XMIT, EXER, or RXML</td>
<td>1</td>
</tr>
<tr>
<td>3F</td>
<td>63</td>
<td>Begin SAF security request</td>
<td>3</td>
</tr>
<tr>
<td>40</td>
<td>64</td>
<td>End SAF security request</td>
<td>3</td>
</tr>
<tr>
<td>41</td>
<td>65</td>
<td>Message sent to OTMA</td>
<td>2</td>
</tr>
<tr>
<td>42</td>
<td>66</td>
<td>Message received from OTMA</td>
<td>2</td>
</tr>
<tr>
<td>43</td>
<td>67</td>
<td>Message sent to SCI</td>
<td>2</td>
</tr>
<tr>
<td>44</td>
<td>68</td>
<td>Message received from SCI</td>
<td>2</td>
</tr>
<tr>
<td>45</td>
<td>69</td>
<td>OTMA Timeout</td>
<td>1</td>
</tr>
</tbody>
</table>
Table 14. Workload-related event records (continued)

<table>
<thead>
<tr>
<th>Code (Hex)</th>
<th>Code (Dec)</th>
<th>Event description</th>
<th>Collection level</th>
</tr>
</thead>
<tbody>
<tr>
<td>46</td>
<td>70</td>
<td>Deallocate Request</td>
<td>3</td>
</tr>
<tr>
<td>47</td>
<td>71</td>
<td>Session Error. This event is called when an unrecoverable error has been encountered and the session is being aborted</td>
<td>1</td>
</tr>
<tr>
<td>48</td>
<td>72</td>
<td>Trigger event. This is the end-of-frame event recorded by IMS Connect when a multi-event process has completed</td>
<td>2</td>
</tr>
<tr>
<td>49</td>
<td>73</td>
<td>Read socket</td>
<td>3</td>
</tr>
<tr>
<td>4A</td>
<td>74</td>
<td>Write socket</td>
<td>3</td>
</tr>
<tr>
<td>4B</td>
<td>75</td>
<td>Local Client Connect</td>
<td>3</td>
</tr>
<tr>
<td>4C</td>
<td>76</td>
<td>Local Message Send</td>
<td>3</td>
</tr>
<tr>
<td>4D</td>
<td>77</td>
<td>Local Message Receive</td>
<td>3</td>
</tr>
<tr>
<td>4E</td>
<td>78</td>
<td>Local Message Send-then-Receive</td>
<td>3</td>
</tr>
<tr>
<td>4F</td>
<td>79</td>
<td>Local Disconnect</td>
<td>3</td>
</tr>
<tr>
<td>50</td>
<td>80</td>
<td>Begin Create Context</td>
<td>4</td>
</tr>
<tr>
<td>51</td>
<td>81</td>
<td>End Create Context</td>
<td>3</td>
</tr>
<tr>
<td>52</td>
<td>82</td>
<td>Begin RRS Prepare</td>
<td>4</td>
</tr>
<tr>
<td>53</td>
<td>83</td>
<td>End RRS Prepare</td>
<td>3</td>
</tr>
<tr>
<td>54</td>
<td>84</td>
<td>Begin RRS Commit/Abort</td>
<td>4</td>
</tr>
<tr>
<td>55</td>
<td>85</td>
<td>End RRS Commit/Abort</td>
<td>3</td>
</tr>
<tr>
<td>56</td>
<td>86</td>
<td>Begin Secure Environment Select</td>
<td>3</td>
</tr>
<tr>
<td>57</td>
<td>87</td>
<td>End Secure Environment Select</td>
<td>3</td>
</tr>
<tr>
<td>58</td>
<td>88</td>
<td>Hold compensation queue</td>
<td>1</td>
</tr>
<tr>
<td>59</td>
<td>89</td>
<td>Port Edit Exit Called</td>
<td>2</td>
</tr>
<tr>
<td>5A</td>
<td>90</td>
<td>Port Edit Exit Returned</td>
<td>1</td>
</tr>
<tr>
<td>5B</td>
<td>91</td>
<td>DRDA Request</td>
<td>2</td>
</tr>
<tr>
<td>5C</td>
<td>92</td>
<td>DRDA Reply</td>
<td>2</td>
</tr>
<tr>
<td>5D</td>
<td>93</td>
<td>An Allocate PSB Command is Received</td>
<td>2</td>
</tr>
<tr>
<td>5E</td>
<td>94</td>
<td>An Allocate PSB Command is Sent</td>
<td>2</td>
</tr>
<tr>
<td>5F</td>
<td>95</td>
<td>A Deallocate PSB Command is Received</td>
<td>2</td>
</tr>
<tr>
<td>60</td>
<td>96</td>
<td>A Deallocate PSB Command is Sent</td>
<td>2</td>
</tr>
<tr>
<td>61</td>
<td>97</td>
<td>ODBM Routing Exit Entered</td>
<td>3</td>
</tr>
<tr>
<td>62</td>
<td>98</td>
<td>ODBM Routing Exit Returned</td>
<td>3</td>
</tr>
<tr>
<td>63</td>
<td>99</td>
<td>ODBM Security Exit Entered</td>
<td>3</td>
</tr>
<tr>
<td>64</td>
<td>100</td>
<td>ODBM Security Exit Returned</td>
<td>3</td>
</tr>
<tr>
<td>65</td>
<td>101</td>
<td>RRS Parent UR Token Creation Begins</td>
<td>3</td>
</tr>
<tr>
<td>66</td>
<td>102</td>
<td>RRS Parent UR Token Creation Ends</td>
<td>3</td>
</tr>
<tr>
<td>69</td>
<td>105</td>
<td>Message is Sent to ODBM</td>
<td>3</td>
</tr>
<tr>
<td>6A</td>
<td>106</td>
<td>Message is Received from ODBM</td>
<td>3</td>
</tr>
<tr>
<td>6B</td>
<td>107</td>
<td>RRS Delegate Commit Agent UR Begins</td>
<td>3</td>
</tr>
<tr>
<td>6C</td>
<td>108</td>
<td>RRS Delegate Commit Agent UR Ends</td>
<td>3</td>
</tr>
<tr>
<td>6D</td>
<td>109</td>
<td>XML Adapter CALL Begin</td>
<td>1</td>
</tr>
</tbody>
</table>

Chapter 24. Events recorded by IMS Connect Extensions  555
Table 14. Workload-related event records (continued)

<table>
<thead>
<tr>
<th>Code (Hex)</th>
<th>Code (Dec)</th>
<th>Event description</th>
<th>Collection level</th>
</tr>
</thead>
<tbody>
<tr>
<td>6E</td>
<td>110</td>
<td>XML Adapter CALL End</td>
<td>1</td>
</tr>
<tr>
<td>6F</td>
<td>111</td>
<td>XML Converter CALL Begin</td>
<td>1</td>
</tr>
<tr>
<td>70</td>
<td>112</td>
<td>XML Converter CALL End</td>
<td>1</td>
</tr>
<tr>
<td>71</td>
<td>113</td>
<td>Connected to Remote IMS Connect</td>
<td>3</td>
</tr>
<tr>
<td>72</td>
<td>114</td>
<td>Disconnected from Remote IMS Connect</td>
<td>3</td>
</tr>
<tr>
<td>73</td>
<td>115</td>
<td>Start Remote Thread for Partner ICON</td>
<td>3</td>
</tr>
<tr>
<td>74</td>
<td>116</td>
<td>Message Received for Remote ALTPCB</td>
<td>3</td>
</tr>
<tr>
<td>75</td>
<td>117</td>
<td>ALTPCB Message Sent to Remote Partner</td>
<td>3</td>
</tr>
<tr>
<td>76</td>
<td>118</td>
<td>ALTPCB Message Received from Remote Partner</td>
<td>3</td>
</tr>
<tr>
<td>77</td>
<td>119</td>
<td>Message Sent to OTMA for ALTPCB</td>
<td>3</td>
</tr>
<tr>
<td>78</td>
<td>120</td>
<td>MSC Message Received from MSC</td>
<td>3</td>
</tr>
<tr>
<td>79</td>
<td>121</td>
<td>MSC Message Sent to Partner IMS Connect</td>
<td>3</td>
</tr>
<tr>
<td>7A</td>
<td>122</td>
<td>MSC Message Received from Partner IMS Connect</td>
<td>3</td>
</tr>
<tr>
<td>7B</td>
<td>123</td>
<td>MSC Message Sent to MSC</td>
<td>3</td>
</tr>
<tr>
<td>7C</td>
<td>124</td>
<td>Connection to Partner IMS Connect Time-out</td>
<td>3</td>
</tr>
<tr>
<td>7D</td>
<td>125</td>
<td>Start of a session</td>
<td>3</td>
</tr>
<tr>
<td>7E</td>
<td>126</td>
<td>Trigger end of a session</td>
<td>3</td>
</tr>
<tr>
<td>80</td>
<td>128</td>
<td>IMS Connect Extensions Automatic Trigger</td>
<td>1</td>
</tr>
<tr>
<td>81</td>
<td>129</td>
<td>IMS Connect Extensions Exception Event</td>
<td>1</td>
</tr>
<tr>
<td>82</td>
<td>130</td>
<td>Event Collection Status</td>
<td>1</td>
</tr>
<tr>
<td>AD</td>
<td>173</td>
<td>IMS Connect Extensions ACEE Cache Delete Ageing Cycle Event</td>
<td>1</td>
</tr>
<tr>
<td>AE</td>
<td>174</td>
<td>Pre-Routing Exit Audit Event</td>
<td>1</td>
</tr>
<tr>
<td>Code (Hex)</td>
<td>Code (Dec)</td>
<td>Extended event number and event description</td>
<td>Collection level</td>
</tr>
<tr>
<td>-----------</td>
<td>------------</td>
<td>-------------------------------------------------------------------------------------------------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>FF</td>
<td>255</td>
<td>Socket connected on RMTCICS</td>
<td>1</td>
</tr>
<tr>
<td>256</td>
<td></td>
<td>Socket disconnected from RMTCICS</td>
<td></td>
</tr>
<tr>
<td>257</td>
<td></td>
<td>IMS Connect refreshed a cached RACF user ID after receiving a type 71 Event Notification Facility (ENF) notification</td>
<td></td>
</tr>
<tr>
<td>258</td>
<td></td>
<td>IMS Connect sent a health status report to Work Load Manager (WLM)</td>
<td></td>
</tr>
<tr>
<td>259</td>
<td></td>
<td>Communication thread started for a RMTCICS connection</td>
<td></td>
</tr>
<tr>
<td>2050</td>
<td></td>
<td>ISC message received from IMS</td>
<td></td>
</tr>
<tr>
<td>2051</td>
<td></td>
<td>ISC message sent to IMS</td>
<td></td>
</tr>
<tr>
<td>2052</td>
<td></td>
<td>ISC message received on RMTCICS socket connection</td>
<td></td>
</tr>
<tr>
<td>2053</td>
<td></td>
<td>ISC message sent on RMTCICS socket connection</td>
<td></td>
</tr>
<tr>
<td>2054</td>
<td></td>
<td>ISC message sent on CICSPORT socket connection</td>
<td></td>
</tr>
<tr>
<td>2055</td>
<td></td>
<td>ISC message received on CICSPORT socket connection</td>
<td></td>
</tr>
<tr>
<td>2056</td>
<td></td>
<td>ISC message sent on CICSPORT socket connection</td>
<td></td>
</tr>
</tbody>
</table>

### IMS Connect Extensions trace records

IMS Connect Extensions trace 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.

IMS Connect Extensions trace event records are produced when tracing has been activated. See "Using the IMS Connect Extensions trace" on page 93.

<table>
<thead>
<tr>
<th>Code (Hex)</th>
<th>Code (Dec)</th>
<th>Event description</th>
<th>Tracing level</th>
</tr>
</thead>
<tbody>
<tr>
<td>A3</td>
<td>163</td>
<td>Event Collection OTMA Trace</td>
<td>1</td>
</tr>
<tr>
<td>A4</td>
<td>164</td>
<td>Event Collection IRM Trace</td>
<td>1</td>
</tr>
<tr>
<td>A5</td>
<td>165</td>
<td>Event Recording RSM Message Trace</td>
<td>1</td>
</tr>
<tr>
<td>A6</td>
<td>166</td>
<td>Event Recording EXIT Output Message Trace</td>
<td>1</td>
</tr>
<tr>
<td>A9</td>
<td>169</td>
<td>RXML Trace for XML Adapter</td>
<td>1</td>
</tr>
<tr>
<td>AA</td>
<td>170</td>
<td>ODBM trace record</td>
<td>1</td>
</tr>
<tr>
<td>AC</td>
<td>172</td>
<td>IMS Connect Extensions Command and Response</td>
<td>1</td>
</tr>
</tbody>
</table>

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 is issued to suspend and resume processing for an IMS data store. 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** 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 IMS data store in the routing list. The response record for the routing list will contain the highest return code and reason code for all of the data store records. For more information on this command, see “**ROUTE command**” on page 452.

### IMS Connect Recorder Trace facility records

Trace data produced by IMS Connect Recorder Trace facility is converted to event record format by the IMS Connect Extensions Recorder Trace utilities. See “**RECODER trace utilities**” on page 579.

**Table 17. Recorder trace event records produced by IMS Connect**

<table>
<thead>
<tr>
<th>Code (Hex)</th>
<th>Code (Dec)</th>
<th>Event description</th>
<th>Collection level</th>
</tr>
</thead>
<tbody>
<tr>
<td>A0</td>
<td>160</td>
<td>HWS Recorder Base Section Trace</td>
<td>Not applicable</td>
</tr>
<tr>
<td>A1</td>
<td>161</td>
<td>HWS Recorder IPB Section Trace</td>
<td>Not applicable</td>
</tr>
<tr>
<td>A2</td>
<td>162</td>
<td>HWS Recorder OPB Section Trace</td>
<td>Not applicable</td>
</tr>
</tbody>
</table>

### User data logging record

The Event Recording User Data Logging record is produced when user data is submitted to the journal for logging. 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.

**Table 18. User data logging event record**

<table>
<thead>
<tr>
<th>Code (Hex)</th>
<th>Code (Dec)</th>
<th>Event description</th>
<th>Collection level</th>
</tr>
</thead>
<tbody>
<tr>
<td>AB</td>
<td>171</td>
<td>Event Recording User Data Logging</td>
<td>1</td>
</tr>
</tbody>
</table>

**Related tasks:**

“**Updating the event collection level**” on page 195

Use the **Update > Event Collection Level** context menu option in the Status Monitor of the Operations Console to change the IMS Connect event collection level on selected IMS Connect systems. The setting is changed both in-memory and in the repository definition.

**Related reference:**

“**Sample event flows in IMS Connect**” on page 58

The following sample event flows demonstrate a typical sequence of events that can be captured in IMS Connect with IMS Connect Extensions event collection.
Chapter 25. Mapping event records

Use the CEXCRnn members in the cexpr.USERMAC library to 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 PFX=CERE PFX=prefix ID=ALL

HEXRID=NO

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 26. 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.

AUTOADD_DATASTORE option

The AUTOADD_DATASTORE option specified in the CEXCTLIN data set sets the number of entries in the IMS data store table reserved for adding data stores that are dynamically defined in IMS Connect.

Syntax

```
AUTOADD_DATASTORE COUNT=n
```
**Parameters**

**COUNT**

The number of entries in the data store table.

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 data stores 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 data stores in the IMS Connect configuration member so that they do not have to be dynamically added again.

**Example**

AUTOADD_DATASTORE COUNT=(20)

---

**CEXROUTE option**

The CEXROUTE option specified in the CEXCTLIN data set configures routing options for IMS Connect Extensions.

**Syntax**

```
CEXROUTE
  IMSCMDROUTE=NO
  IMSCMDROUTE=YES
  INELIGIBLEIF=GLOBALFLOODSEVERE
  INELIGIBLEIF=GLOBALFLOODWARNING
  RBR_FAILURE=ORIGINAL
  RBR_FAILURE=REJECT
  RBR_NODEST=PASS
  RBR_NODEST=destid
  RBR_NOALIAS=PASS
  RBR_NOALIAS=alias
```

**Parameters**

**IMSCMDROUTE**

When 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 IMS data stores that are experiencing a Global Flood Warning condition. Specify one of the following options:
GLOBALFLOODWARNING
If INELIGIBLEIF=GLOBALFLOODWARNING is specified and the Global Flood Warning condition occurs (Degraded state) then the candidate data store will become ineligible for routing as if the status were Unavailable.

GLOBALFLOODSEVERE
If INELIGIBLEIF=GLOBALFLOODSEVERE is specified and a Global Flood Severe condition occurs (Unavailable state) then the candidate data store 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 OTMA rules-based routing is being used and IMS Connect Extensions determines that it cannot route to any of the destinations specified in the rule that matched the incoming destination ID. Specify one of the following options:

ORIGINAL
If RBR_FAILURE=ORIGINAL is specified, the input message is passed to IMS Connect with the destination set to the original destination. If the original destination is a generic IMS data store name, message HWS50742W is issued. This is the default.

REJECT
If RBR_FAILURE=REJECT is specified, the input message is rejected with an RSM indicating that no destination is available. Use this option to force the use of OTMA rules-based routing for all message routing in IMS Connect.

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 incoming IRM_IMSDestID in the IRM is not matched with a routing rule during the initial pass. Specify one of the following options:

destid
If RBR_NODEST=destid is specified, the value supplied in the IRM_IMSDestID of the IRM is changed to the value specified and the rules are scanned again to see if a rule matching the destination ID specified is present. If a matching rule is not found, processing continues as if no rule is matched.

PASS
If RBR_NODEST=PASS is specified, the IRM is passed through unmodified. This is the default.

RBR_NOALIAS
Controls how ODBM rules-based routing functions if the inbound alias is not found in the active ODBM routing rules during the initial pass. Specify one of the following options:

alias
If RBR_NOALIAS=destid is specified, the alias specified in the client request 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: The ODBM routing rule that matches the value you specify for alias must not refer to a PSB name list.

PASS
If RBR_NOALIAS=PASS is specified, the DRDA request is passed through unmodified. This is the default.

Example
CEXROUTE IMSCMDROUTE=NO +
       INELIGIBLEIF=GLOBALFLOODSEVERE +
       RBR_FAILURE=ORIGINAL +
       RBR_NODEST=PASS +
       RBR_NOALIAS=PASS

Related concepts:
“OTMA flood control processing in IMS Connect Extensions” on page 267
IMS Connect Extensions OTMA rules-based routing can be used to avoid message flood conditions in IMS.

“Pre-routing user exit” on page 269
You can use the pre-routing exit to influence the selection of IMS data stores as candidates for OTMA rules-based routing.

Related tasks:
“Rejecting transactions for data stores with no OTMA routing rule” on page 265
You can enforce the use of OTMA routing-rules in IMS Connect Extensions by creating a special routing rule that contains no target IMS data stores and by using a combination of CEXCTLIN control options that work with this rule to reject workloads that cannot be matched to your existing set of active routing rules.

CEXTRACE option
The CEXTRACE option specified in the CEXCTLIN data set configures the size of the IMS Connect Extensions trace buffer.

Syntax
CEXTRACE BUFSIZE=n

Parameters
BUFSIZE
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:

\[(4 + \#ports) \times (cextrace_value \times 1024)\]
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.

**Example**

CEXTRACE BUFSIZE=1

---

**EVENTLOGGING option**

The EVENTLOGGING option specified in the CEXCTLIN data set determines if event records should be buffered 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.

**Syntax**

```
EVENTLOGGING
```

**Parameters**

**WRITE**

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. This is the default.

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.

**Example**

EVENTLOGGING WRITE=BUFFER

---

**ICON_CONTROL option**

The ICON_CONTROL option specified in the CEXCTLIN data set is used to configure a dedicated IMS Connect port which is to be used to route IMS type-1 commands that are issued from IMS Connect Extensions. The command shell connects to an available IMS Connect message port to submit IMS type-1 commands to an IMS data store 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.

**Syntax**

```
ICON_CONTROL
```

---
**Parameters**

**PORT**

Use the 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 data stores for an IMS system are stopped then there will be no pathway for executing IMS type-1 commands.

**Example**

```
ICON_CONTROL PORT=65535
```

**MESSAGE option**

The MESSAGE option specified in the CEXCTLIN data set controls which messages prefixed with CEX and FUN are written to SYSLOG. The ADD_WTO suboption writes additional messages to SYSLOG. The DROP_CEXPRINT suboption stops specified messages being written to CEXPRINT.

**Syntax**

```
MESSAGE ADD_WTO=(msglist), DROP_CEXPRINT=(msglist)
```

**Parameters**

**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.

**Note:** 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.
Example

The following command will write the messages FUN2515I and CEX5403E to SYSLOG and stop CEX5407I being written to CEXPRINT:

```plaintext
MESSAGE ADD_WTO=(2515,5403),DROP_CEXPRINT=(5407)
```

**ODBMDEFAULTS option**

The ODBMDEFAULTS option specified in the CEXCTLIN data set configures the default behavior for ODBM routing in IMS Connect Extensions.

**Syntax**

```
ODBMDEFAULTS DEFAULTACTION=INPUTALIAS
```

**Parameters**

**DEFAULTACTION**

Sets the default action for ODBM routing when the routing exit cannot make a routing decision due to one 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.

The DEFAULTACTION parameter accepts the following options:

**INPUTALIAS**

If DEFAULTACTION=INPUTALIAS is specified, the alias received in the client request is used as the output alias and the ODBM name is set to blanks. This is the default.

**Notes:**

1. This is the normal behavior of the HWSROUT0 exit supplied with IMS Connect.
2. This is also the default 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.
   - During routing processing, a required ODBM routing rule could not be found.

**BLANKALIAS**

If DEFAULTACTION=BLANKALIAS is specified, the request is passed through with the ODBM name and alias both set to blanks.
SECURITY option

The SECURITY option specified in the CEXCTLIN data set allows you to configure security-related options for IMS Connect Extensions.

Syntax

```
SECURITY
  CACHED_USER_MAX=10000
  PWCASE=SYSTEMDEFINITION
  PWCASE=UPPER
  PWCASE=MIXED
  VALIDATE_TRUSTED=NO
  VALIDATE_TRUSTED=YES
```

Parameters

**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. See “Caching user credentials to improve performance” on page 301. 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**

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. Select from the following options:

**SYSTEMDEFINITION**

The setting from the Activate Mixed Case Passwords option in the system definition is used. See “Defining IMS Connect systems” on page 320.

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**

If Activate Security and Activate Validation are selected in the IMS Connect system definition, and the Validation type is set to 2. IMS Connect + IP
Address + Port (see “Defining IMS Connect systems” on page 320), 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 parameter determines whether or not to perform IP validation for messages that are considered trusted. A message is considered trusted if the OMUSR_TRSTUSR flag is set in the incoming message, or the message acquired the trusted attribute from an IP address rule. For more information on IP address rules, see “Creating workload rules for specific IP addresses” on page 305.

NO Messages from trusted users are not validated. This is the default.

YES Messages from trusted users are 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.

For more information on trusted-users in IMS Connect, see Trusted-user support for IMS Connect messages in the IMS user documentation.

Example

```
SECURITY CACHED_USER_MAX=10000 +
PWCLASS=AUTOMATIC +
VALIDATE_TRUSTED=NO
```

**SERVICE_EXIT option**

The SERVICE_EXIT option specified in the CEXCTLIN data set controls the behavior of the IMS Connect Extensions Service exit (CEXSVC01) when inspecting the inbound message header.

**Syntax**

```
SERVICE_EXIT
  CHECK_MSG_HDR=N
  CHECK_MSG_HDR=Y
```

**Parameters**

**CHECK_MSG_HDR**

Inspect the 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.

**Example**

```
SERVICE_EXIT CHECK_MSG_HDR=N
```

**ZIIP_OFFLOAD option**

The ZIIP_OFFLOAD option specified in the CEXCTLIN data set determines if IMS Connect Extensions is to offload event collection processing to available zIIP processors.
Syntax

```
ZIIP_OFFLOAD OPTION=N
 OPTION=Y
 OPTION=A
```

Parameters

**OPTION**

zIIP offload option. Select from the following values:

- **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.

Example

```
ZIIP_OFFLOAD OPTION=Y
```
Chapter 27. 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. Switch the active IMS Connect Extensions 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 "Managing the IMS Connect Extensions journal" on page 163.

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. See "Browsing the message log for an IMS Connect" on page 164.

3. Run the utility against the desired journal. See "Sample JCL: CEXASUT" on page 572 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 plug-in for z/OS Explorer; 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 activity for a system, event collection must be active and the collection level must be set to a value of 1 or higher. For more information, see "Defining IMS Connect systems" on page 320.

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.
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.V3R1M0.SFUNLNX,DISP=SHR
//            DD DSN=CEX.V3R1M0.SCEXLNK,DISP=SHR
//SYSUDUMP   DD SYSOUT**
//MSGOUT     DD SYSOUT**
//EVNTIN     DD DISP=SHR,DSN=journal.data.set
```

*Figure 209. 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.

**Related concepts:**

"Active Sessions” on page 130

Use the Active Sessions panel in the IMS Connect Extensions ISPF dialog to manage and monitor active OTMA, ODBM, and MSC sessions in IMS Connect. 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, the Active Sessions panel is of most use when you are aware of a transaction that is taking longer to complete than expected.

"Active sessions” on page 209

Active sessions in the IMS Connect Extensions Operations Console allows you to monitor 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.

**Related reference:**

"QUERY commands” on page 436

The **QUERY** host commands for REXX reports on statistics and settings for IMS Connect Extensions resources.

---

**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
Determine 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.V3R1M0.SCEXLINK
// DD DISP=SHR,DSN=FUN.V3R1M0.SFUNLINK
//EVNTIN DD DISP=SHR,DSN=journal.data.set
//SYSIN DD *
PRINT FORMAT=T
/*
//SYSUDUMP DD SYSOUT**
//MSGOUT DD SYSOUT**
/*
```

Figure 210. JCL to run the IMS Connect Extensions print utility

PRINT command

The print utility PRINT command determines the output format and the range of records to process.

Syntax

```
PRINT FORMAT=T
  ZONE=GMT
FORMAT=D
  ZONE=LOCAL
  SERVER
  STAFT=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 command

The print utility RECORD command enables you to select records for the formatted output based on their ID and event record key. If you enter both ID and EVNTKEY keywords, found records will have to match one ID and one event record key.

Syntax

```
>>>RECORD
    ID=(event_ID)
    EVNTKEY=(event_key)
``` 

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 IMS Connect status event records. Entering more than one value will return those events matching any of the values specified.
READ62 control card example

RECORD EVNTKEY=(0123456789ABCDEF,
+ BD44160E8790300,
+ BD44160C98C65201,
+ EVNT)

READ62 command
The print utility READ62 command enables you to specify conditions to select records for the formatted output.

Syntax

```
READ62
  COND=OR
  COND=AND
  CLIENT=(client_ID)
  LTERM=(LTERM_name)
  TRANCODE=(transaction_type)
  EXIT=(exit_name)
  IPV4ADR=IP_address
  IPV6ADR=IP_address
```

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=BAEBA9E652E7C00, PREPARE READ SOCKET
ID=49,073, V=12, TSKID=0105, KEY=BAEBA9E652E7C00, READ SOCKET
ID=49,073, V=12, TSKID=0105, KEY=BAEBA9E652E7C00, READ SOCKET
ID=3D,061, V=12, TSKID=0105, KEY=BAEBA9E652E7C00, MSG EXIT ENTERED
ID=3E,062, V=12, TSKID=0105, KEY=BAEBA9E652E7C00, MSG EXIT RETURN
ID=41,065, V=12, TSKID=010E, KEY=BAEBA9E652E7C00, MSG SENT TO OTMA
ID=42,066, V=12, TSKID=010E, KEY=BAEBA9E652E7C00, MSG RCV FROM OTMA
ID=3D,061, V=12, TSKID=0105, KEY=BAEBA9E652E7C00, MSG EXIT ENTERED
ID=3E,062, V=12, TSKID=0105, KEY=BAEBA9E652E7C00, MSG EXIT RETURN
ID=4A,074, V=12, TSKID=0105, KEY=BAEBA9E652E7C00, WRITE SOCKET
ID=49,073, V=12, TSKID=0105, KEY=BAEBA9E652E7C00, READ SOCKET
ID=49,073, V=12, TSKID=0105, KEY=BAEBA9E652E7C00, READ SOCKET
ID=49,073, V=12, TSKID=0105, KEY=BAEBA9E652E7C00, READ SOCKET
ID=3D,061, V=12, TSKID=0105, KEY=BAEBA9E652E7C00, MSG EXIT ENTERED
ID=3E,062, V=12, TSKID=0105, KEY=BAEBA9E652E7C00, MSG EXIT RETURN
ID=41,065, V=12, TSKID=010E, KEY=BAEBA9E652E7C00, MSG SENT TO OTMA
ID=48,072, V=12, TSKID=010E, KEY=BAEBA9E652E7C00, TRIGGER EVENT
```

Figure 211. 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 Chapter 25, “Mapping event records,” on page 559 for a layout of every event record.
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)
Converting 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 CEX RTCNV utility.

Related tasks:
"Controlling the IMS Connect Recorder Trace facility" on page 96
Use IMS Connect Extensions ISPF dialog to start and stop the IMS Connect Recorder Trace facility.
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
//STEP02 DD DISP=SHR,DSN=CEX.V3R1M0.SCEXLINK
// DD DISP=SHR,DSN=FUN.V3R1M0.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 213. JCL to run RECORDER trace conversion utility CEXRTCNV

Related concepts:
Chapter 24, “Events recorded by IMS Connect Extensions,” on page 551

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.

Related tasks:
“Using Recorder Trace” on page 202

Use the Start Recorder Trace and Stop Recorder Trace context menu options in the Status Monitor of the IMS Connect Extensions Operations Console to start and stop the IMS Connect Recorder Trace facility.

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.V3R1M0.SCEXLINK
    DD DISP=SHR,DSN=FUN.V3R1M0.SFUNLINK
//SYSUDUMP DD SYSOUT=*  
//CEXPRINT DD SYSOUT=*   
//MSGOUT DD SYSOUT=*   
//RCDRIN DD DISP=SHR,DSN=your.recorder.output
/*

Figure 214. 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=CO
SMF: LEN=0052, RECORD TYPE=77, TIME=11410582, SEQ=0104122F
UOW: CLNT=TRAN0004, RCVT=BZ26BB8839516C06, ENQT=BZ26BB8830D512E40
DS IST MSG DOT=0000000000000000, DS CLR DOT=0000000000000000
ERR TIME=BZ26BB8830D516E40, # MSG XMIT=0000, # MSG RCVD=0000
TIME=BZ26BB883B38672F200, LSN=0000000000000000

ID=A1,161, TOKEN=00000001
ITOC REC: LEN=0010, APAR=0001, CON=80, CON1=02, TYPE=ITOC / RC / IPB
REMOTE CLIENT IRM: LLLL=x’00000001’
DUMP OF IRM HEADER FOR LENGTH=x’0058’
+0000 00580000 2A53414D 504C452A 00000000 .........(&<......
+0010 00000000 5452241E 30303034 00200020 .........+
+0020 50415254 20202020 49404433 20202020 ......&{......
+0030 20202020 20202020 44565020 20202020 ..........&......
+0040 20202020 20202020 20202020 20202020 .............
+0050 70656163 68202020 ;;;
DUMP OF CLIENT MSG SEGMENT FOR LENGTH=x’0011’
+0000 00010000 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 00000040 *
+0040 40404040 40404040 *
DUMP OF OTMA SECURITY SECTION FOR LENGTH=x’006A’
+0000 006A0000 00000000 00000000 00000000 
+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 00000000 00000000 *
DUMP OF OTMA USER SECTION FOR LENGTH=x’0100’
+0000 00100000 00000000 00000000 00000000 
+0010 00000000 00000000 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 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 215. RECORDER trace print utility: Sample output
Chapter 28. 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
//REPOSTARY DD DISP=SHR, DSN=your.def.file
//ARCHCNTL DD
CLEAN PRODUCT=CEX, LENGTH=8, NAME=(HWS1, HWS2), FUNCTION=R, CHECK=YES
/*
Figure 216. 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 Common Services Library 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,...hwn),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,...hwn)
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=mode
The acceptable values are:
L List mode. Prints to CEXPRINT a list of all archive journal records in the repository for the specified systems. This is the default.
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” on page 543 and “FUN2571I” on page 543. The processing start time that is reported by “FUN1003I” on page 534 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.

DELELERROR=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),DELELERROR=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: CEXARCCJ**

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▌).

```
//CEXARCCJ 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 217. JCL to run the Archive Manager utility

**EXEC statement**

The format of the EXEC statement is:

```
---------- //stepname ---<---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" on page 541 or "FUN2533W" on page 541.

**DD statements**

**STEPLIB DD**

The IMS Connect Extensions product and Common Services Library 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=CEX310.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=CEX310.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=CEX310.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.

**REPOSTR 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 /*
```

**Important:** Do not use this statement in an archive JCL skeleton.

**Related concepts:**
These topics describe how to define active IMS Connect Extensions journal data sets and then archive them to a direct access storage device (DASD) or magnetic tape.

Related reference:

“Defining active journal data set processing options” on page 329

The Active Journal Data Set panel contains processing options for the active IMS Connect Extensions journal. To access this ISPF panel, select option 1.1 System Definitions from the IMS Connect Extensions primary menu and then select Active Data Set.
Chapter 29. 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.
Parameters

NAME
The name of the test case. You must use one of the test case names from the following table:

*Table 19. Available IVP test cases*

<table>
<thead>
<tr>
<th>Name</th>
<th>Commit mode</th>
<th>SYNCELEVEL</th>
<th>Transaction</th>
<th>Exit</th>
<th>Persistent socket?</th>
</tr>
</thead>
<tbody>
<tr>
<td>CEXCSE01</td>
<td>1</td>
<td>None</td>
<td>PART</td>
<td>HWSSMPL1</td>
<td></td>
</tr>
<tr>
<td>CEXCSE02</td>
<td>1</td>
<td>Confirm</td>
<td>PART</td>
<td>HWSSMPL1</td>
<td></td>
</tr>
<tr>
<td>CEXCSE03</td>
<td>1</td>
<td>None</td>
<td>PART</td>
<td>HWSSMPL1</td>
<td>Yes</td>
</tr>
<tr>
<td>CEXCSE04</td>
<td>1</td>
<td>Confirm</td>
<td>PART</td>
<td>HWSSMPL1</td>
<td>Yes</td>
</tr>
<tr>
<td>CEXCSE05</td>
<td>1</td>
<td>Confirm</td>
<td>N/A</td>
<td>CEXSVT01</td>
<td></td>
</tr>
<tr>
<td>CEXCSE06</td>
<td>1</td>
<td>None</td>
<td>PART</td>
<td>HWSSMPL0</td>
<td></td>
</tr>
<tr>
<td>CEXCSE07</td>
<td>1</td>
<td>Confirm</td>
<td>PART</td>
<td>HWSSMPL0</td>
<td></td>
</tr>
<tr>
<td>CEXCSE08</td>
<td>1</td>
<td>None</td>
<td>PART</td>
<td>HWSSMPL0</td>
<td>Yes</td>
</tr>
<tr>
<td>CEXCSE09</td>
<td>1</td>
<td>Confirm</td>
<td>PART</td>
<td>HWSSMPL0</td>
<td>Yes</td>
</tr>
</tbody>
</table>

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_expression)

**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_expression)
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 security is enabled in the IMS Connect system definition in the IMS Connect Extensions repository. See "Defining IMS Connect systems" on page 320.
- 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.

```
SEND,PRINT=Y
```

**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.

```
RECEIVE, PRINT=N, F, Y, 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.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Purpose</th>
<th>Default value</th>
<th>Validation performed</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPVers</td>
<td>Internet Protocol version</td>
<td>IPV4</td>
<td>IPV4 or IPV6</td>
</tr>
</tbody>
</table>
| IPAddr   | DNS name or IP address of the IMS Connect system | none | 1. Must be set to a DNS name or an IP address. 
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 |
| ODAlias  | (Optional) The datastore alias for the target ODBM | none | length(ODAlias) < 5 |
| SRVNAME  | (Optional) The server name | MVS1 | none |

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 security is enabled in the IMS Connect system definition in the IMS Connect Extensions repository. See “Defining IMS Connect systems” on page 320.
- If IMS Connect Extensions security is enabled in the system definition.

Otherwise, PASSWORD is optional.
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 cexpr.USERMAC library.

Note: The socket is always closed after issuing a service exit request, even if the socket is persistent.

Activating the client services exit

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.”

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.

Related reference:
“Defining user exits” on page 336

The User Exits panel allows you to configure user message exits. To access this ISPF panel, select option 1.2 User Exits from the IMS Connect Extensions primary menu. You can define the standard IBM-supplied exits to IMS Connect Extensions using a LOAD command.

---

**Input message format**

Here is the general format of an input message.

![Message Format Diagram](image)

**Figure 218. Input message format**

**Figure 218** 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.

![Output message format diagram](image)

Figure 219. Output message format

<table>
<thead>
<tr>
<th>LLLL</th>
<th>Fullword that specifies the total length of all data in the message.</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSM</td>
<td>The request status message (RSM). The RSM has the following form:</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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 20 shows the format of the input message service header.

<table>
<thead>
<tr>
<th>Offset</th>
<th>Type</th>
<th>Length</th>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>SIGNED</td>
<td>2</td>
<td>SVC1_HDR_LEN</td>
<td>This segment's length. Enter 32.</td>
</tr>
<tr>
<td>2</td>
<td>N/A</td>
<td>2</td>
<td>ZZ</td>
<td>low values</td>
</tr>
<tr>
<td>4</td>
<td>CHARACTER</td>
<td>4</td>
<td>SVC1_HDR_ID</td>
<td>Control block ID. You must use: SVC1</td>
</tr>
<tr>
<td>8</td>
<td>SIGNED</td>
<td>2</td>
<td>SVC1_HDR_SR</td>
<td>The service request type:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>240 Verify user</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>241 Password change</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>242 Who-am-I</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>243 Log user data</td>
</tr>
<tr>
<td>10</td>
<td>SIGNED</td>
<td>2</td>
<td>SVC1_HDR_NSEG</td>
<td>Number of segments in the message</td>
</tr>
<tr>
<td>12</td>
<td>N/A</td>
<td>20</td>
<td></td>
<td>low values</td>
</tr>
</tbody>
</table>

The following table shows the format of the output message service header.

Table 21. Service response header segment format

<table>
<thead>
<tr>
<th>Offset</th>
<th>Type</th>
<th>Length</th>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>SIGNED</td>
<td>2</td>
<td>SVC1_HDR_LEN</td>
<td>This segment's length (always 32)</td>
</tr>
<tr>
<td>2</td>
<td>N/A</td>
<td>2</td>
<td>ZZ</td>
<td>low values</td>
</tr>
<tr>
<td>4</td>
<td>CHARACTER</td>
<td>4</td>
<td>SVC1_HDR_ID</td>
<td>Control block ID</td>
</tr>
<tr>
<td>8</td>
<td>SIGNED</td>
<td>2</td>
<td>SVC1s_HDR_SR</td>
<td>The service request type:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>240 Verify user</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>241 Password change</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>242 Who-am-I</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>243 Log user data</td>
</tr>
<tr>
<td>10</td>
<td>SIGNED</td>
<td>2</td>
<td>SVC1_HDR_NSEG</td>
<td>Number of segments in the message</td>
</tr>
</tbody>
</table>
Table 21. Service response header segment format (continued)

<table>
<thead>
<tr>
<th>Offset</th>
<th>Type</th>
<th>Length</th>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>SIGNED</td>
<td>4</td>
<td>SVC1_HDR_RC</td>
<td>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.</td>
</tr>
<tr>
<td>16</td>
<td>BITSTRING</td>
<td>4</td>
<td>SVC1_HDR_RSN</td>
<td>Reason code</td>
</tr>
<tr>
<td>20</td>
<td>CHARACTER</td>
<td>4</td>
<td>SVC1_HDR_INFO</td>
<td>For some return codes that are not zero, provides the offset to the segment containing the error.</td>
</tr>
<tr>
<td>24</td>
<td>N/A</td>
<td>8</td>
<td></td>
<td>Low values</td>
</tr>
</tbody>
</table>

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 cexpr.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 encrypt them.

Table 22. Verify user request format

<table>
<thead>
<tr>
<th>Offset</th>
<th>Type</th>
<th>Length</th>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>SIGNED</td>
<td>2</td>
<td>SVC1_VUSI_LEN</td>
<td>This segment's length. Enter 24.</td>
</tr>
<tr>
<td>2</td>
<td>N/A</td>
<td>2</td>
<td>ZZ</td>
<td>Low values</td>
</tr>
<tr>
<td>4</td>
<td>SIGNED</td>
<td>2</td>
<td>SVC1_VUSI_SEGN</td>
<td>Segment number</td>
</tr>
<tr>
<td>6</td>
<td>N/A</td>
<td>2</td>
<td>ZZ</td>
<td>Low values</td>
</tr>
<tr>
<td>8</td>
<td>CHARACTER</td>
<td>8</td>
<td>SVC1_VUSI_UID</td>
<td>User ID. If shorter than 8 characters, pad with spaces.</td>
</tr>
<tr>
<td>16</td>
<td>CHARACTER</td>
<td>8</td>
<td>SVC1_VUSI_PW</td>
<td>Password. If shorter than 8 characters, pad with spaces.</td>
</tr>
</tbody>
</table>

The following table shows the format of the “verify user” formatted segments that are returned for a service request type of 240.

Table 23. Verify user response format

<table>
<thead>
<tr>
<th>Offset</th>
<th>Type</th>
<th>Length</th>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>SIGNED</td>
<td>2</td>
<td>SVC1_VUSO_LEN</td>
<td>This segment's length (always 28).</td>
</tr>
<tr>
<td>2</td>
<td>N/A</td>
<td>2</td>
<td>ZZ</td>
<td>Low values</td>
</tr>
<tr>
<td>4</td>
<td>SIGNED</td>
<td>2</td>
<td>SVC1_VUSO_SEGN</td>
<td>Segment number</td>
</tr>
<tr>
<td>6</td>
<td>SIGNED</td>
<td>2</td>
<td>SVC1_VUSO_MSMSG</td>
<td>Number of messages in list.</td>
</tr>
<tr>
<td>8</td>
<td>CHARACTER</td>
<td>8</td>
<td>SVC1_VUSO_UID</td>
<td>User ID</td>
</tr>
<tr>
<td>16</td>
<td>SIGNED</td>
<td>4</td>
<td>SVC1_VUSO_SAFRC</td>
<td>SAF return code</td>
</tr>
</tbody>
</table>
Table 23. Verify user response format (continued)

<table>
<thead>
<tr>
<th>Offset</th>
<th>Type</th>
<th>Length</th>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Offset</th>
<th>Type</th>
<th>Length</th>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>28</td>
<td>0</td>
<td>SVC1_VUSO_MSGLIST</td>
<td>Message list</td>
<td></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>MSGLL</td>
<td>SVC1_VUSO_MSG</td>
<td>Message text</td>
</tr>
</tbody>
</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 308 for details.

Table 24. Password change request format

<table>
<thead>
<tr>
<th>Offset</th>
<th>Type</th>
<th>Length</th>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>SIGNED</td>
<td>2</td>
<td>SVC1_CPWI_LEN</td>
<td>This segment’s length. Enter 32.</td>
</tr>
<tr>
<td>2</td>
<td>N/A</td>
<td>2</td>
<td>ZZ</td>
<td>low values</td>
</tr>
<tr>
<td>4</td>
<td>SIGNED</td>
<td>2</td>
<td>SVC1_CPWI_SEGN</td>
<td>Segment number</td>
</tr>
<tr>
<td>6</td>
<td>N/A</td>
<td>2</td>
<td>ZZ</td>
<td>low values</td>
</tr>
<tr>
<td>8</td>
<td>CHARACTER</td>
<td>8</td>
<td>SVC1_CPWI_UID</td>
<td>User ID. If shorter than 8 characters, pad with spaces.</td>
</tr>
<tr>
<td>16</td>
<td>CHARACTER</td>
<td>8</td>
<td>SVC1_CPWI_OPW</td>
<td>Old password. If shorter than 8 characters, pad with spaces.</td>
</tr>
<tr>
<td>24</td>
<td>CHARACTER</td>
<td>8</td>
<td>SVC1_CPWI_NPW</td>
<td>New password. If shorter than 8 characters, pad with spaces.</td>
</tr>
</tbody>
</table>

The following table shows the format of the “password change” formatted segments that are returned for a service request type of 241.

Table 25. Password change response format

<table>
<thead>
<tr>
<th>Offset</th>
<th>Type</th>
<th>Length</th>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>SIGNED</td>
<td>2</td>
<td>SVC1_CPWO_LEN</td>
<td>This segment’s length (always 28)</td>
</tr>
<tr>
<td>2</td>
<td>N/A</td>
<td>2</td>
<td>ZZ</td>
<td>low values</td>
</tr>
<tr>
<td>4</td>
<td>SIGNED</td>
<td>2</td>
<td>SVC1_CPWO_SEGN</td>
<td>Segment number</td>
</tr>
<tr>
<td>6</td>
<td>SIGNED</td>
<td>2</td>
<td>SVC1_CPWO_#MSG</td>
<td>Number of messages in list</td>
</tr>
<tr>
<td>8</td>
<td>CHARACTER</td>
<td>8</td>
<td>SVC1_CPWO_UID</td>
<td>User ID</td>
</tr>
<tr>
<td>16</td>
<td>SIGNED</td>
<td>4</td>
<td>SVC1_CPWO_SAFRC</td>
<td>SAF return code</td>
</tr>
</tbody>
</table>
Table 25. Password change response format (continued)

<table>
<thead>
<tr>
<th>Offset</th>
<th>Type</th>
<th>Length</th>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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>
</tbody>
</table>

When extended RSM support is requested for the Service Exit, SVC1_CPWO_RSMSG 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_RSMSG.

<table>
<thead>
<tr>
<th>Offset</th>
<th>Type</th>
<th>Length</th>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>28</td>
<td>0</td>
<td></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>8</td>
<td>SVC1_CPWO_MSG</td>
<td>Message text</td>
</tr>
</tbody>
</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 26. Who-am-I request format

<table>
<thead>
<tr>
<th>Offset</th>
<th>Type</th>
<th>Length</th>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>SIGNED</td>
<td>2</td>
<td>SVC1_WHOI_LEN</td>
<td>This segment's length. Enter 16.</td>
</tr>
<tr>
<td>2</td>
<td>N/A</td>
<td>2</td>
<td>ZZ</td>
<td>low values</td>
</tr>
<tr>
<td>4</td>
<td>SIGNED</td>
<td>2</td>
<td>SVC1_WHOI_SEGN</td>
<td>Segment number</td>
</tr>
<tr>
<td>6</td>
<td>N/A</td>
<td>2</td>
<td>ZZ</td>
<td>low values</td>
</tr>
<tr>
<td>8</td>
<td>CHARACTER</td>
<td>8</td>
<td>SVC1_WHOI_NAME</td>
<td>Queried field name, which is one of the following values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CLIENTID</td>
<td>Client ID</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>HWSID</td>
<td>Server ID (pad with spaces)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>IPVER</td>
<td>IP protocol version (pad with spaces)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>IPADDR</td>
<td>Client IP address (pad with spaces)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PORTNO</td>
<td>Client outgoing port number (pad with spaces)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SOCKTYPE</td>
<td>Socket type (pad with spaces)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>EVENTKEY</td>
<td>Event key (pad with spaces)</td>
</tr>
</tbody>
</table>

The following table shows the format of the “who-am-I” formatted segments that are returned for a service request type of 242.

Table 27. Who-am-I response format

<table>
<thead>
<tr>
<th>Offset</th>
<th>Type</th>
<th>Length</th>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>SIGNED</td>
<td>2</td>
<td>SVC1_WHOO_LEN</td>
<td>This segment's length (maximum length is 32 bytes)</td>
</tr>
<tr>
<td>2</td>
<td>N/A</td>
<td>2</td>
<td>ZZ</td>
<td>low values</td>
</tr>
</tbody>
</table>
Table 27. Who-am-I response format (continued)

<table>
<thead>
<tr>
<th>Offset</th>
<th>Type</th>
<th>Length</th>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>SIGNED</td>
<td>2</td>
<td>SVC1_WHOO_SEGN</td>
<td>Segment number</td>
</tr>
<tr>
<td>6</td>
<td>N/A</td>
<td>2</td>
<td>ZZ</td>
<td>low values</td>
</tr>
<tr>
<td>8</td>
<td>CHARACTER</td>
<td>8</td>
<td>SVC1_WHOO_NAME</td>
<td>The name of the field that was queried</td>
</tr>
</tbody>
</table>

One of the following, depending on the queried field:

<table>
<thead>
<tr>
<th>Offset</th>
<th>Type</th>
<th>Length</th>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>CHARACTER</td>
<td>8</td>
<td>SVC1_WHOO_CLIENTID</td>
<td>The client ID</td>
</tr>
<tr>
<td>16</td>
<td>CHARACTER</td>
<td>8</td>
<td>SVC1_WHOO_HWSID</td>
<td>The server ID</td>
</tr>
<tr>
<td>16</td>
<td>CHARACTER</td>
<td>2</td>
<td>SVC1_WHOO_IPVER</td>
<td>The IP protocol version, where:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>V4 IPv4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>V6 IPv6</td>
</tr>
<tr>
<td>16</td>
<td>BITSTRING</td>
<td>4 or 16</td>
<td>SVC1_WHOO_IPADDR</td>
<td>The IP address. 16 bytes for an IPv6 address. 4 bytes for an IPv4 address.</td>
</tr>
<tr>
<td>16</td>
<td>UNSIGNED</td>
<td>2</td>
<td>SVC1_WHOO_PORTNO</td>
<td>The port from which the client is connecting</td>
</tr>
<tr>
<td>16</td>
<td>CHARACTER</td>
<td>1</td>
<td>SVC1_WHOO_SOCKTYPE</td>
<td>The type of socket connection:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>P Persistent socket</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>N Non-persistent or transaction socket</td>
</tr>
<tr>
<td>16</td>
<td>BITSTRING</td>
<td>8</td>
<td>SVC1_WHOO_EVENTKEY</td>
<td>The event key.</td>
</tr>
</tbody>
</table>

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 28. Log user data request format

<table>
<thead>
<tr>
<th>Offset</th>
<th>Type</th>
<th>Length</th>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>SIGNED</td>
<td>2</td>
<td>SVC1_LOGI_LEN</td>
<td>This segment’s length.</td>
</tr>
<tr>
<td>2</td>
<td>N/A</td>
<td>2</td>
<td>ZZ</td>
<td>low values</td>
</tr>
<tr>
<td>4</td>
<td>SIGNED</td>
<td>2</td>
<td>SVC1_LOGI_SEGN</td>
<td>Segment number</td>
</tr>
<tr>
<td>6</td>
<td>N/A</td>
<td>2</td>
<td>ZZ</td>
<td>low values</td>
</tr>
<tr>
<td>8</td>
<td>CHARACTER</td>
<td>1</td>
<td>SVC1_LOGI_TYPE</td>
<td>User data type:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>C Text</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X Binary (default)</td>
</tr>
<tr>
<td>9</td>
<td>CHARACTER</td>
<td>1</td>
<td>SVC1_LOGI_CODE</td>
<td>User log code (optional, user-defined).</td>
</tr>
<tr>
<td>10</td>
<td>SIGNED</td>
<td>2</td>
<td>SVC1_LOGI_CCSID</td>
<td>User data CCSID (optional, user-defined).</td>
</tr>
<tr>
<td>12</td>
<td>CHARACTER</td>
<td>8</td>
<td>SVC1_LOGI_FMTID</td>
<td>User data format identifier (optional, user-defined).</td>
</tr>
<tr>
<td>20</td>
<td>CHARACTER</td>
<td>n</td>
<td>SVC1_LOGI_DATA</td>
<td>User data (maximum 960 bytes).</td>
</tr>
</tbody>
</table>

The following table shows the format of the “log user data” formatted segments that are returned for a service request type of 243.

Table 29. Log user data response format

<table>
<thead>
<tr>
<th>Offset</th>
<th>Type</th>
<th>Length</th>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>SIGNED</td>
<td>2</td>
<td>SVC1_LOGO_LEN</td>
<td>This segment’s length (always 16).</td>
</tr>
</tbody>
</table>
Table 29. Log user data response format (continued)

<table>
<thead>
<tr>
<th>Offset</th>
<th>Type</th>
<th>Length</th>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>N/A</td>
<td>2</td>
<td>ZZ</td>
<td>low values</td>
</tr>
<tr>
<td>4</td>
<td>SIGNED</td>
<td>2</td>
<td>SVC1_LOGO_SEGN</td>
<td>Segment number.</td>
</tr>
<tr>
<td>6</td>
<td>N/A</td>
<td>2</td>
<td>ZZ</td>
<td>low values</td>
</tr>
<tr>
<td>8</td>
<td>SIGNED</td>
<td>4</td>
<td>SVC1_LOGO_RC</td>
<td>Log writer return code.</td>
</tr>
<tr>
<td>12</td>
<td>SIGNED</td>
<td>4</td>
<td>SVC1_LOGO_RSN</td>
<td>Log writer reason code.</td>
</tr>
</tbody>
</table>

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 30. Reason and return codes

<table>
<thead>
<tr>
<th>Return code</th>
<th>Reason code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Successfully completed.</td>
</tr>
<tr>
<td>8</td>
<td>041E0008</td>
<td>Unsuccessful user verification. The SAF return code and RACF return and reason codes are posted in the response segment.</td>
</tr>
<tr>
<td>8</td>
<td>041E0009</td>
<td>Unsuccessful password change request. The SAF return code and RACF return and reason codes are posted in the response segment.</td>
</tr>
<tr>
<td>8</td>
<td>041E000B</td>
<td>Unsuccessful verify user request. The SAF return code and RACF return and reason codes are posted in the response segment.</td>
</tr>
<tr>
<td>12</td>
<td>041E0001</td>
<td>Length specified for message does not match total length of all segments.</td>
</tr>
<tr>
<td>12</td>
<td>041E0002</td>
<td>A message field format is invalid. Check the segment lengths, for negative values or values that are out of range.</td>
</tr>
<tr>
<td>12</td>
<td>041E0003</td>
<td>Header segment length field (SVC1_HDR_LEN) does not match actual length.</td>
</tr>
<tr>
<td>12</td>
<td>041E0004</td>
<td>Control block ID (SVC1_HDR_ID) not recognized. Use SVC1.</td>
</tr>
<tr>
<td>12</td>
<td>041E0005</td>
<td>The number of segments specified in the header (SVC1_HDR_NSEG) does not match the actual number of segments.</td>
</tr>
<tr>
<td>12</td>
<td>041E0006</td>
<td>Service request (SVC1_HDR_SR) not recognized. Use a known request value.</td>
</tr>
<tr>
<td>12</td>
<td>041E0007</td>
<td>The length specified for a service request segment does not match the actual segment length.</td>
</tr>
<tr>
<td>12</td>
<td>041E0008</td>
<td>The number of the service request segment does not match its actual order.</td>
</tr>
<tr>
<td>Return code</td>
<td>Reason code</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>12</td>
<td>041E000A</td>
<td>A who-am-I request was made with an unrecognized field name (SVC1_WHOI_NAME).</td>
</tr>
<tr>
<td>12</td>
<td>041E000C</td>
<td>Client not authorized. The SAF return code and RACF return and reason codes are posted in the response segment.</td>
</tr>
<tr>
<td>12</td>
<td>041E000D</td>
<td>User data type error. Value must be C=text or X=binary. Default is binary.</td>
</tr>
<tr>
<td>12</td>
<td>041E000E</td>
<td>Event logging failed. The event writer return and reason codes are posted in the response segment.</td>
</tr>
<tr>
<td>16</td>
<td>041E0101</td>
<td>IMS Connect Extensions not active.</td>
</tr>
<tr>
<td>16</td>
<td>041E0102</td>
<td>Internal error in EXIT function.</td>
</tr>
<tr>
<td>16</td>
<td>041E0103</td>
<td>Internal error when XMIT was entered.</td>
</tr>
<tr>
<td>16</td>
<td>041E0104</td>
<td>Internal error when EXER entered.</td>
</tr>
</tbody>
</table>
Chapter 31. Legacy OTMA routing methods

The best way to balancing workloads and routing IRM messages in IMS Connect Extensions is by defining OTMA routing rules that direct messages with specific IMS data store IDs to a collection of IMS data stores defined in an IMS Connect Extensions routing list. If you are not yet using OTMA routing rules, you can continue to use IMS Connect Extensions with two legacy routing methods for OTMA messages: transaction routing and primary datastore routing.

Related concepts:

Chapter 12, “OTMA workload routing in IMS Connect,” on page 245

OTMA rules-based routing in IMS Connect Extensions allows you to route and distribute OTMA workload received by IMS Connect between available IMS data stores.

Transaction routing

Transaction routing is a legacy process by which IMS Connect Extensions selects a destination IMS data store for an incoming IMS request message (IRM) 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 IMS data stores and no others, regardless of the destination ID specified on the incoming messages. This can be useful for security and system testing.
Important: OTMA 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 Chapter 12, “OTMA workload routing in IMS Connect,” on page 245). 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.

Transaction routing can be set on several levels:

1. If system transaction routing is inactive, the message is passed to IMS Connect with the IMS data store 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 IMS data store should be used, and the primary data store is available, the message is passed to IMS Connect with the primary data store rather than the original destination. Otherwise, processing continues.

   For more information on using primary data stores, see “Primary data store routing” on page 616.

---

**Figure 220. Determining if a transaction is eligible for transaction routing in IMS Connect Extensions using the data store (DS) in the request message**

**Diagram:**

- **Msg[Original DS]**
  - Set Destination DS to the Original DS
  - System Transaction Routing active?
    - Yes: Use primary datastore?
      - Yes: Set Destination DS to the Primary DS
        - Primary DS available?
          - Yes: Build candidate list for transaction and Destination DS
          - No: Transaction eligible for Routing?
            - Yes: Destination DS eligible for Routing?
              - Yes: Pass to IMS Connect with Destination DS
              - No: Transaction eligible for Routing?
                - No: Build candidate list for transaction and Destination DS
        - No: Transaction eligible for Routing?
          - Yes: Build candidate list for transaction and Destination DS
          - No: Transaction eligible for Routing?
            - Yes: Destination DS eligible for Routing?
              - Yes: Build candidate list for transaction and Destination DS
              - No: Build candidate list for transaction and Destination DS
    - No: Transaction eligible for Routing?
      - Yes: Build candidate list for transaction and Destination DS
      - No: Transaction eligible for Routing?
        - Yes: Build candidate list for transaction and Destination DS
        - No: Build candidate list for transaction and Destination DS
3. If the transaction is not eligible for routing, the message is passed to IMS Connect with either the original destination or the primary IMS data store. 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 data store is not eligible for routing, the message is passed to IMS Connect with the destination data store. Otherwise, processing continues. The destination data store is either the original data store or the primary data store, if the primary data store option is active.

To determine the eligibility of data store for routing, select or clear **Activate Transaction Routing** in the Datastore definition panel.

5. A list of candidate data stores is generated by IMS Connect Extensions. Candidate data stores are those that are associated with both the transaction and the destination data store. The destination data store is either the original or primary data store. For more information on how the candidate list is constructed, see [“Candidate datastore list” on page 611.](#)

If a candidate list is built for the transaction, IMS Connect Extensions will dynamically select a destination data store.
Figure 221 shows how a destination data store 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 data store. 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 347.

  To override the routing error processing option on the Application level, and specify routing error processing for the transaction, see “Defining transactions” on page 349.

- If the candidate list contains only one candidate, the transaction is passed to IMS Connect with the destination data store set to the candidate data store.

- If the candidate list contains more than one candidate, the destination data store 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 data store routing includes logic designed to avoid individual data stores 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 data store that is not in Degraded (flood warning) state.
2. If no available data store can be found, the search is expanded to include data stores in Degraded state.

**Note:** If the CEXROUTE INELIGIBLEIF=GLOBALFLOODWARNING parameter is set in the CEXCTLIN data set, any data store that is in “Degraded” state is treated as being in the “Unavailable” state and therefore cannot be a candidate for routing. See “CEXROUTE option” on page 562.

3. If no available candidate can be found, the standard process for handling an empty candidate list is followed, as shown in Figure 221 on page 610.

**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 616.

**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.

1. TX1, IMD2

2. Transaction affinity for TX1
   - IMP1
   - IMP2
   - IMP3
   - IMP4
   - IMT1
   - IMT2

3. IMD1
   - IMD2
   - Datastore affinity for IMD2

4. Route to:
   - IMD1

*Figure 222. 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 223 on page 613*, 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.

1. TX1, IMP1

2. IMP1  IMP2  IMP3  IMP4  Datastore affinity for IMP1
   IMT1  IMT2

3. IMD1  IMD2  Transaction affinity for TX1

4. No candidates

5. Error processing:
   IMP1  OR  Reject message

Figure 223. Example: Candidate list with error routing option

When IMS Connect Extensions constructs the candidate list, the incoming message has the transaction set to TX1 and the destination datastore set to IMP1.

The destination datastore IMP1 has Datastore Affinity with datastores IMP1, IMP2, IMP3, and IMP4.

The transaction TX1 has Transaction Affinity with datastores IMD1 and IMD2.

None of the datastores have a common affinity, so there are no eligible candidates.

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 data store routing” on page 616 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" on page 616.

**Procedure**

1. Define transaction routing rules. See "Routing messages based on the transaction code and destination ID" on page 613.
2. Activate Send Only routing for those data stores 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 data stores 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 data stores. Instead, affinity lists are used to associate candidate data stores 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 data stores 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

1. Activate transaction routing for the IMS systems that process the Resume TPIPE requests.
2. Create an affinity list with the candidate data stores.
3. Activate Resume TPIPE routing for the datastore definitions for which you want to route Resume TPIPE and select the candidate affinity list.

Primary data store routing

Primary data store routing is a legacy routing method lets you nominate a data store as a primary for an IMS Connect system. Messages are routed to the primary data store unless it is unavailable.

If the data store is unavailable, you can automatically revert to transaction routing as a backup mechanism (see Figure 220 on page 608).

You can use primary data store routing to handle the following scenarios:

• You want to route to a specified data store but have backup routing available if this data store is unavailable.

• You want to use a generic destination data store name and route to the appropriate data store for the IMS Connect system handling the message.

   For example (shown in Figure 224 on page 617), route all messages with the destination data store ORG1 to the data store IMS1 if passing through the IMS Connect system HWS1, and to the data store 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 data store is unavailable under rules-based routing, you cannot revert to transaction routing as a backup mechanism.
Figure 224 shows that primary data stores are defined in datastore groups. The datastore group contains a data store and the IMS Connect system for which it is the primary.

As shown in Figure 225 on page 618,

- If the destination data store is defined as a data store in IMS Connect Extensions, a primary data store is not used.
- If the destination data store is not defined, IMS Connect Extensions checks if a data store group with the same name as the destination exists. If one does not exist, the message is passed to IMS Connect with the original data store as the destination. If the data store group exists, IMS Connect Extensions checks the data store group to find which data store is configured as the primary, and sets the destination to the primary data store.
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.
In this example, an incoming message with the destination ID ORG1, which passes through the IMS Connect system HWS1, is routed to IMD1.

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=GLOBAI_FLOODWARNING parameter is set in the CEXCTLIN data set, any datastore that is in “Degraded” state is treated as being in the “Unavailable” state and therefore cannot be a candidate for routing. See CEXROUTE option on page 562.

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 32. 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.

►►required_item◄◄
  ►optional_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◄

• 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

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.

Index

Special characters
(+1) symbol in data set name 62
$NOTRANS transaction code 349, 562
&DATE/&JDATE symbols 62
&FIRST/&FIRST symbols 62
&ID symbol 62
&TIME symbol 62

A
accessibility 24
ACEE cache activating 326
CLEAR command 431
clear user ID 200
display statistics 124
maximum number of cached users 568
properties 326
statistics, display 199
using SECURITY command to clear 156
ACK/NAK report 87
ACTION=DRAIN|RESUME parameter 452
Activate Commands option 235
activate IMS Connect Extensions 322
activate mixed case passwords 322
Activate Session Message Limit field 276
active data set 324
configuration 394
active journal configuring 68
data set name 330
definition 329
deleting data sets 70
number of data sets 330
overflow data set 331, 396
overview 55
primary quantity 331, 396
reset 332
resizing data sets 70
sizing 62
skeleton JCL 332
active session 209
active session utility 571
help 571
active sessions batch reporting utility 571
draining 434
eligible 434
export 186
filter 136
overview 130
reset panel to default layout 141
session network details 135
session wait time filter 131
Active sessions Managing 211
Active sessions (continued)
Opening 210
ADD command, definition maintenance utility 380
ADD host command 429
ADD_WTO keyword 566
advanced features 325
affinity list definition 346
definition maintenance utility ADD command 381
refresh command 153
AFL (affinity list) ADD command 381
ageing interval 326
Alternate transaction code 351, 412
APF-authorize load libraries 35
API 463
APP (application) ADD command 382
application associated with system definition 335
definition 347
definition maintenance utility ADD command 382
refresh command 153
routing error 348
routing option 348
transaction routing 348
application management solutions 22
APPLID 238
ARCHCNTL DD 588
archive data set 324
configuration 396
Archive JCL Skeleton warning 328
archive journal activate journal archiving 329
browsing in IMS Problem Investigator 73
CEXARCHn sample members 63
cleanup utility 585
data set name 62, 333, 392
date and time symbols 62, 588
definition 332
GDG 333, 392, 587
maximum active 333, 396
maximum size 333, 396
maximum time 333, 396
maximum volumes 334, 397
overview 55
skeleton JCL 63, 396
template 69
Archive Manager activating 330, 395
overview 55
skeleton 332, 396
utility 64, 587
archiving activating 330, 395
manually, JCL sample 587
skeleton JCL 63
using magnetic tape 63
utilities 585
ARCCOUT DD 588
ASCII CHECK_MSG_HDR control option 569
Message ID format in user exit 338
to EBCDIC translation 311
authorize program libraries 35
authorized programs list (AUTHPGM and AUTHTSF) 241
auto refresh 189, 210
AUTOADD_DATASTORE control option 561
Automating IMS Connect with REXX 429
AUTORESUME keyword 452

B
BLANKALIAS option 567
BPE exit list member 31
BPECFG member 31
BUFFSIZE keyword 564

C
CACHED_USER_MAX option 568
candidate data stores
ODBM rules-based routing 279
OTMA rules-based routing 246
candidate list example 612
introduction 611
capacity rating 343
ODBM target 363
CASE (IVP statement) 591
CCSID 311
CEX control options data set 30
CEXAEQUO macro 464
CEXAPxxx mapping macros 467
CEXARCH 332, 396
CEXARCHn sample members 63, 64, 66
CEXASUT sample active sessions utility 571
CEXAUTH0 security exit configuring 31
HWSAUTH0 security exit 31
CEXCA2DP feed program 101
CEXCER mapping macro 467, 559
CEXCESA sample member 478
CEXCRnn mapping macros 559
CEXCSKT mapping macro 467
CEXCSVFI DD statement 415
CEXCTLIN DD statement 30, 561
CEXDFDSI (definition maintenance) utility 379
CEXDFDST (definition takeup) utility 376
CEXEVT sample JCL 572

629
command shell (continued)  
issuing commands 166  
TCP/IP connection failure 484  
commands  
activating 235  
Clear ACEE 156  
configuration 397  
issuing from the command shell 166  
JOURNAL REFRESH 163  
JOURNAL SWITCH 163  
list of 238  
REFRESH 150  
Refresh SAF Class rules 156  
resource names 238  
securing 235, 238  
SECURITY 155  
SET 157  
SET APPLICATION 161  
SET DATASTORE 159  
SET DATASTORE GROUP 160  
SET SYSTEM DEFINITION 158  
SET TRANSACTION 162  
USER EXIT 147  
using 145  
Common Services Library  
authorize program libraries 35  
console 177  
Common Services Library messages 534  
conditional tracing 200  
configuration file  
import 177  
configuration member  
HWS_CONFIG parameter 394  
configuration parameters  
feed 102  
CONNECT (CEXAPI function) 468  
Connect Commands shell 212  
CONNECT host command 431  
connect status event records 56, 552  
connection failure 484  
connection profile 180  
console  
Common Services Library 177  
console client 21  
console listener 12  
console port 321  
CONTROL control card, ECSA cleanup utility 479  
control input data set 30, 561  
cookie policy 625, 628  
CSV export mode 415  
customization 25  
CX line action 145  
data set name, generating unique 62  
data set not cataloged (warning) 328  
data store definition maintenance utility ADD command 383  
data stores  
dynamically added 274  
dat stencil group  
definition 344  
dat astore group (continued)  
definition maintenance utility ADD command 385  
reject threshold 345  
transaction pacing 345  
warning threshold 345  
Datostore Group refresh command 153  
Datostore Monitor 141  
Datostore refresh command 152  
datapstore routing  
flood-control processing 619  
datastores  
overview 204  
DATE symbol in data set name 62  
DEFAULTACTION keyword 567  
definition extract utility 415  
definition maintenance utility 379  
ADD command 380  
definition takeover utility 376  
definition utilities, overview 375  
definitions commands and fields in list panels 318  
commands in edit panels 319  
refreshing 150, 449  
Definitions Setup utility 32, 315  
TAKEUP command 377  
DEFN ADD command, general syntax 380  
DELAY host command 433  
DELETE command, definition maintenance utility 413  
DELETE control card, ECSA cleanup utility 480  
DESTUNA 487  
diagnostic information, gathering 547  
dialog  
ISPF Environment Considerations 47  
disability 24  
DISCONN (CEXAPI function) 469  
DNS Name 322  
documentation accessing 23  
changes 4  
DRAIN  
active sessions 434  
dat astore 452  
host command 434  
DRDA  
ODBM rules-based routing 279  
DRDA port 126, 203  
dynamically added 125  
DROP_CEXPRINT option 566  
DS (data store) ADD command 383  
DSG (dat astore group) ADD command 385  
dynamically added data stores 274  
dynamically added ports 125  

EBCDIC  
CHECK MSG_HDR control option 569  
Message ID format in user exit 338  
to ASCII translation 311
export data option

EXTRACT command 419

F
feed, IMS Connect Extensions 99
fields feed 110
filter
active sessions 136
status monitor 128
systems in operations dialog 122
Filter field
in active sessions 136
in definition list panels 318
in status monitor 128
FILTER primary command 128
FIRST symbol in data set name 62
flood-control processing
datatrace routing 619
OTMA rules-based routing 267
transaction routing 611
force-stop command (publisher API client) 173
FORM command
active sessions 138
forwarding a live feed of IMS Connect events 99
framing events 56
function keys 47
Functional Support Library
Common Services Library 28
IMS Connect startup job 35

G
gap analysis report 88
GDG 333, 392, 587
GDG in archive data set 62
GEN_PASSSTICKET parameter 397
getting started
IMS Connect Extensions Operations Console 180
GLOBALFLOODSEVERE option 562
GLOBALFLOODWARNING option 562
GRP ADD command 389
GUI client 12
main window 50

H
hardware requirements 27
help, using 48
high-level qualifier (cepre) 32
HOST (IVP statement) 592
host command environment for
REXX 429
example 229, 231
samples 223
variables 234
Host command environment for
REXX 221
host name warning 328
HWS (IMS Connect system) ADD command 389
HWS ID parameter 392
HWS_CONFIG parameter 394

HWSAUTH0 security exit 14
HWSCFG
See IMS Connect configuration member
HWSEXITN exit list member 31
HWSJAVA exit 14
HWSPL1442E message 483
HWSROUT0 exit 14
replace with CEXROUT0 31
HWSMPn exits 14
IVP jobs 38

I
IBM OMEGAMON for IMS 20
IBM Transaction Analysis Workbench 20
ICON_CONTROL control option 565
ID symbol in data set name 62
IMPORT command 424
import configuration file 177
import/export utility 421
EXPORT command 422
import command 424
SET command 426
VERIFY command 426
IMS alias
overview 208
IMS Batch Backout Manager 22
IMS Batch Terminal Simulator 22
IMS command
TCP/IP connection failure 484
IMS commands
routings 562
IMS Commands shell 215
IMS Configuration Manager for z/OS
plug-in 215
IMS Connect
ACEE cache statistics 199
activating command access 235
commands, issuing 212
definition maintenance utility, system
ADD command 389
events, forwarding a live feed of 99
message port 565
port, access control by 303
RECORDER trace 93, 558
reporting 73
security enhancements 301
security validation 302
start/stop tracing 96
startup JCL 35
trace event records 557
validation 327
viewing system in IMS Connect
Extensions 37
IMS Connect configuration member
data set 335
DEFN—HWS parameter 392
edit 328
HWS_CONFIG parameter 394
member name and PROCLIB 30
IMS Connect Extensions
client 21
features and benefits 8
feed 99
overview 3
IMS Connect Extensions (continued)
product libraries 27
start/stop Recorder Trace 96
start/stop trace 93
start/stop tracing 93
terminology 21
utilities 19
IMS Connect Extensions Operations
Console
online help 51
IMS Connect transaction index 99
IMS Connects view 194
IMS data store
capacity rating 343
definition 340
reject threshold 341
transaction pacing 341
transaction routing 342
warning threshold 341
IMS MODBLKS, transaction takeup 353
IMS Performance Analyzer 20
reports 82
IMS Problem Investigator 20, 73, 571
merging log files 79
reporting ODBM activity 80
IMS Program Restart Facility 22
IMS sysplex distributor 276
IMS type-1 commands, issuing 215
IMS Type-2 commands, issuing 215
IMS-to-CICS TCP/IP communications
event records 554
IMS-to-IMS TCP/IP communications
event records 56, 554
IMSCMDROUTE keyword 562
INACT status, troubleshooting 483
INELIGIBLEIF keyword 562
INIT (CEXPIAPI function) 471
initial configuration, Definitions Setup
utility 32
INPUTALLIAS option 567
installation 25
preparing information for 29
verifying 37
viewing IMS Connect system to verify 37
integration 463
internal 487
internal reason codes 487
interval count 325
IP address 322
access control by 301, 303
IP address rule
definition maintenance utility ADD command 399
IP address rules 372
IP Address Rules refresh command 155
IPR (IP address rule) ADD command 399
IPv4, IPv6 profile strings 303
IRM (IPv statement) 592
IRM offsets 337
IRM options 277, 305, 349, 410
ISC (intersystem communication) 554
ISPF dialog 47
client 11
CXA conventions 47
customizing 42
ISPF dialog (continued)
environment considerations 47
maintaining definitions in 315
ISPF libraries 27
IVP
CEXIVP sample members 38
CEXIVPO sample members 40
reference 591
J
JCL
active session utility 572
archive skeleton 63
feed 101
IMS Connect 35
print utility 572
RECODER conversion utility 580
RECODER print utility 581
sample resource profile 240
JDATE symbol in data set name 62
JFIRST symbol in data set name 62
journal
deleting data sets 70
generating unique name 62
JOURNAL REFRESH command 163
JOURNAL SWITCH command 55, 163
Journal switching 194
K
Keep Alive (timeout) value 125
keyboard shortcuts 24, 51
legal notices
cookie policy 625, 628
notices 625
programming interface
information 625, 627
trademarks 625, 627
libraries 32
Limit threshold field 276
limit threshold, sessions 196
links
IBM-Web sites 627
LIST control card, ECSA cleanup utility 479
listener, console 12
LOAD command 338
LOCATE 319
log files 79
log record number 323
log user data
event records 558
format, CEXSVC01 client services exit 684
log, message 217
LookAt 489
M
manual refresh 189, 210
Maximum clients option
Publisher Status - Client List panel 173
system definition 173, 324
MAXSOC 483
merging in IMS Problem Investigator 79
MESSAGE control option 489, 566
message count 196
Message Count field 276
message log 217
export 186
message logs 164
message recall count 322
message translation 311
message-related event records 56, 554
messages
CEX prefixed 490
FUN prefixed 534
methods for accessing 489
written to SYSLOG 488, 566
messages and codes 487
migration actions 44
mixed case passwords 322
MODBLKS, transaction takeup 353
MSC
overview 209
MSCVRL
QUIET or VERBOSE 435
MSGOUT
IVP output 39
N
National Language Support 42
NEW command, definition 319
NEWAPPL 42
NOLOGON status, publisher API console 173
normal default processing
ODBM 299
notices 625, 627
ODBM
and IMS PI reporting 80
overview 208
ODBM routing lists 367
definition maintenance utility ADD command 400
ODBM routing rule
definition maintenance utility ADD command 401
ODBM routing rules 364
ODBM Routing Rules refresh command 155
ODBM rules-based routing 279
activating 294, 327
CEXCTLIN control options 299
configuring 281
creating a routing rule 282
default processing 299
fallback data stores 287
processing rules 279
PSB name 288
pacing
activating 325
PASSAPPL 42
PassTicket processing
enabling 240
PassTicket program libraries,
authorizing 241
Passwords
enabling PassTicket processing 240
pending responses, query 438, 441
persistent socket sessions 131
PLN (transaction) ADD command 405
port
dynamically added 125
keepalive (timeout) value 125
suffix on port number (D or S) 126, 203
port overview 125, 203
port usage report 85
pre-routing user exit 269
primary datastore routing 616
PRINT command
print utility 573
print utility
introduction 572
JCL 572
PRINT command 573
READ62 command 575
RECORD command 574
transaction detail report 576
transaction summary report 576
problems
diagnostic information about 547
product libraries 27
profile
time zone 125
profile settings
time zone 32
warning override 32
programming interface information 625, 627
prompt 48
Properties view 183
PSB ADD command 404
PSB name lists 368
definition maintenance utility ADD command 404
PU (publisher API console) line command 170
publisher API
activate 323
console 170
disconnect clients 173
maximum clients 173, 324
NOLOGON status 173
OMEGAMON configuration 393
overview 463
programming environment 464
reason codes and return codes 466
stop clients 173
view clients 170
publisher APIxxx mapping macros 467
PWCASE option 568

Q

QUERY (CEXPAPI function) 471
QUERY host command 441
QUERY TYPE=ACCE_CACHE host command 437
QUERY TYPE=JOURNAL host command 438
QUERY TYPE=PENDING_RESPONSES host command 438
QUERY TYPE=PLAN host command 440
QUERY TYPE=SOCKETS host command 443
QUERY TYPE=STATUS_MONITOR host command 444
QUERY TYPE=TRACE host command 449

R

RACF 599
class 238
command security 238
enabling PassTicket processing 241
example 238
resource names 238
sample job 240
RACROUTE FASTAUTH request 302
RBR_FAILURE keyword 562
RBR_NOALIAS keyword 562
RBR_NODEST keyword 562
Read Prepare event (start of frame) 56
READ62 command
print utility 575
READASYN (CEXPAPI function) 472
READSYNC (CEXPAPI function) 474
reason codes 487
RECEIVE (IVP statement) 594
RECORD command
print utility 574
RECORDER trace
conversion utility 580
event records 56, 558
introduction 579
print utility 581
starting and stopping 93
Recorder Trace facility
start 96
Recorder Tracing 202
refresh 189, 210
refresh commands
Affinity List 153
Application 153
Datastore 152
Datastore Group 153
IP Address Rules 155
ODBM Routing Rules 155
OMEGA Routing Rules 154
System Definition 151
Transaction 154
Refresh Commands menu 150
REFRESH host command 449
reject threshold 325, 341, 345
rejected messages, reason 487
Remote Connects
overview 209
rules-based routing
configuring ODBM 281
configuring OTMA 247
dynamically added data stores 274
ODBM, activating 327
OTMA, activating 327
pre-routing user exit 269
sample routing rules 262, 297

S
SAF class rules
using SECURITY command to refresh 156
samples
host command environment for
REXX 223
SCEXEXEC library 28
SCEXLINK library 28
SCXMENU library 28
SCXPENU library 28
SCXSAMP sample library 28
active journal
CXARCHn sample members 63
CXARCHn sample members 63
CXCESA sample member 478
CXEXPRT sample member 421
CXIMPRT sample member 422
CXIVP sample members 38, 398
CXIVPO sample members 40
CXPAPI sample members 467
SCXSENU library 28
SCXSTNU library 28
screen readers and magnifiers 24
SECFAIL 487
SECINOP 487
security
activating Security Management 326
configuration 397
preprocessing exit 308
security class 327
SECURITY command 155
SECURITY control option 568
PWCASE keyword 302
VALIDATE TRUSTED keyword 302
SECURITY_APPLID parameter 397
SEND (IPV statement) 593
Send Only routing 615
service header, CEXSVC01 client services exit
service information 23
SERVICE_EXIT control option 569
session 209
message limit 196
Session Details panel
ODBM sessions 132
Session Message Limit option 276
session rebalancing 276
session wait time 139
filter active sessions by 131
Sessions Form panel
active sessions 138
SET APPLICATION command 161
SET command
definition extract utility 420
definition maintenance utility 414

SET command (continued)
definition lookup utility 378
import/export utility 426
SET commands menu 157
SET DATASTORE command 159
SET DATASTORE GROUP command 160
Set F1_TRNEXP option 278, 351
SET host command 453
SET SYSTEM DEFINITION command 158
SET TRANSACTION command 162
setup
dynamic 42
static 43
utility 34
SFUNLINK library 27
SHELL host command 454
Show View menu 183
skeleton, Archive Manager 332, 396
socket sessions, persistent 131
software requirements 27
SORT 319
SSL port 126, 203
STARTMON (CEXPAPI function) 475
statistics collection
activating 328
for Status Monitor 125
Status Monitor 187
active sessions 130
Aliases view 208
customizing 128
Datastores view 204
exits overview 204
export 186
MSC’s view 209
Opening 188
port overview 125, 203
Remote Connects view 209
reset to default layout 128
system overview 125, 126, 194
using 125
STEM variables 234
stop command (publisher API)
client) 173
STOPMON (CEXPAPI function) 476
storage requirements 27
summary of changes 4
super member
routiing Resume TPIPE requests 616
view systems 127
support
required information 547
support information 23
SVT token 434
switch active journal 163
SWITCH host command 55, 456
syntax diagrams
how to read 621
sysplex distributor 565
sysplex, grouping by 355
System Definition 320
ACEE cache 326
Activate IMS Connect Extensions 322
Activate Mixed Case Passwords 322
Activate ODBM rules 327
Activate OTMA rules 327
workload balancing (continued)
  OTMA requests  255
workspace
  import systems to  177
WRITE=BUFFER keyword  565
WTO host command  460

X
XML Adapter, XML-related event records  554

Z
z/OS Explorer  50
zIIP offload control option  570