IBM IMS Queue Control Facility for z/OS
3.2

User's Guide

IBM

> **Note:**
>
> Before using this information and the product it supports, read the information in "Notices" on page 467.

Contents

About this information

IBM® IMS Queue Control Facility for z/OS® (also referred to as *IMS Queue Control Facility* or simply *QCF*) is an IMS Tools product that improves message queue handling and manipulates the IMS message queues in both the shared and nonshared queue environments.

These topics provide instructions for installing, configuring, and using IMS Queue Control Facility.

To use these instructions, you must have already installed IMS Queue Control Facility by completing the instructions in the *Program Directory for IBM IMS Queue Control Facility for z/OS* (GI10-8713), which is included with the product media and is also available on the IMS Tools Product Documentation page.

These topics are designed to help database administrators, system programmers, application programmers, and system operators perform the following tasks:

- Understand the capabilities of the functions that are associated with IMS Queue Control Facility
- Install and operate IMS Queue Control Facility
- Customize your IMS Queue Control Facility environment
- Diagnose and recover from IMS Queue Control Facility problems
- Use IMS Queue Control Facility with other IMS products

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

- The z/OS operating system
- ISPF
- SMP/E
- IMS

Always refer to the IMS Tools Product Documentation web page for complete product documentation resources:

https://www.ibm.com/support/pages/node/712955

The IMS Tools Product Documentation web page includes:

- Links to IBM Documentation for the user guides ("HTML")
- PDF versions of the user guides ("PDF")
- Program Directories for IMS Tools products
- Recent updates to the user guides, referred to as "Tech docs" ("See updates to this information!")
- Technical notes from IBM Software Support, referred to as "Tech notes"
- White papers that describe product business scenarios and solutions

Part 1. IMS Queue Control Facility overview

IBM IMS Queue Control Facility for z/OS (also referred to as *IMS Queue Control Facility* or simply *QCF*) is an IMS tool that improves message queue handling and manipulates the IMS message queues in both the shared- and nonshared-queues environments.

Topics:

Chapter 1. IMS Queue Control Facility overview

IBM IMS Queue Control Facility for z/OS (also referred to as *IMS Queue Control Facility* or simply *QCF*) is an IMS queue management tool that manipulates the local, live IMS message queues in both the shared- and nonshared-queues environments.

Topics:

- "What's new in IMS Queue Control Facility" on page 3
- "What does IMS Queue Control Facility do?" on page 4
- "IMS Queue Control Facility terminology" on page 7
- "Service updates and support information" on page 8
- "Product documentation and updates" on page 8
- "Accessibility features" on page 9

What's new in IMS Queue Control Facility

This topic summarizes technical changes for this edition.

New and changed information is indicated by a vertical bar (|) to the left of a change.

Revision markers follow these general conventions:

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

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

SC19-4005-07 - February 2022

The following updates have been made for this edition:

Description	Related APARs
This enhancement changes the message age filter precision for QUERY, BROWSE, RECOVER, and UNLOAD functions. The following topic has been added: • Message IQC4175E The following topics have been changed: • "FUNCTION statement" on page 130 • "FUNCTION BROWSE statement syntax" on page 153 • "FUNCTION QUERY statement syntax" on page 181 • "FUNCTION RECOVER statement syntax" on page 191 • "FUNCTION UNLOAD statement syntax" on page 253 • "Query panels and dialog boxes" on page 277 • "Query for Shared Queues panel " on page 278 • Message IQC4170E	PH42117

SC19-4005-06 - December 2021

The following updates have been made for this edition:

Description	Related APARs
This enhancement enables IMS Queue Control Facility to check whether the 64 KB HIOP buffer size is defined in the IMS control region during IMS initialization. The following topics have been added: • "Overriding buffer definitions for HIOP" on page 52 • Message IQC6221W	PH37593
The following topics have been added for this APAR: • "ACTIONPOOL control statement" on page 57 • Messages IQC6628E, IQC6821I, IQC6822W	PH08814
This enhancement changes the naming conventions for COMMIT, UNCOMMIT, and NOTPARMS data sets. With this enhancement, you can move your IMS subsystems across z/OS systems and LPARs in case of z/OS failure, LPAR failure, or even for workload balancing. The following topic has been changed: • "2. Creating configuration members for message processing" on page 24	PI87887
Other major documentation changes: • "Prerequisite checklist before configuring extensions" on page 32 • "Defining threshold jobs" on page 38 • "AREA control statement" on page 58 • "FAILSAFE control statement" on page 64 • AIB reason codes X'1080', X'1084', X'A008', and X'A074'	N/A

SC19-4005-05 - May 2018

The following updates have been made for this edition:

- SELECT statement, update syntax diagrams and provide details for DI, DIO and DO options in field TYPE
- PI79265 - ONCOLDSTART should not be triggered after **/ERE COLDBASE**
- Updates to FAILSAFE examples
- WTODESTINATION(MTO) note
- New error messages
- PI87212 - new keywords in TOIPARMS statement for UNLOAD

SC19-4005-04 - September 2016

The following updates have been made for this edition:

- Maintenance roll-up
- Packaged for inclusion in IBM IMS System Management for z/OS

What does IMS Queue Control Facility do?

In nonshared-queues environments, IMS Queue Control Facility can query, browse, load, unload, and reprocess messages, provide overflow protection, and rebuild queues after cold starts or emergency restarts of IMS. In shared-queues environments, IMS Queue Control Facility can query, browse, load,

unload, and reprocess messages. IMS Sysplex Manager provides more functionality for shared-queues environments.

Product features

IMS Queue Control Facility provides the following capabilities:

Function	Nonshared-queues environment	Shared-queues environment
Protects your IMS nonshared-queues system from a queues-full condition (U758 abend) with Queue Space Usage Notification (QSUN). Several features are used to diagnose this condition.	Supported	-
Requeues unprocessed messages from your IMS nonshared-queues system onto the IMS message queues after an IMS cold start.	Supported	-
Reinserts previously processed messages from your IMS shared- or nonshared-queues system onto the message queues for reprocessing in IMS environments.	Supported	Supported
Schedules functions that you specify during an IMS cold start or warm start of your IMS shared- or nonshared-queues system.	Supported	Supported
Initiates an offload of the message queues in your IMS nonshared-queues system when a threshold is crossed from A-to-B, B-to-C, C-to-D, and when the threshold that you specify is crossed.	Supported	-
Notifies you of specified queue space usage in your IMS nonshared-queues system.	Supported	-
Performs the following test functions in your IMS shared- or nonshared-queues system when you need transaction data to simulate production loads or application input: • Stress testing • Regression testing • Application testing	Supported	Supported
Recovers messages from your IMS shared-queues system onto the cold queue after an IMS cold start.	-	Supported

Queue overflow protection

IMS Queue Control Facility implements queue overflow protection by monitoring the message queue usage. IMS Queue Control Facility analyzes the message queue usage to determine when to send an alert or to take action to prevent the excessive queue usage from continuing.

You can configure queue overflow protection by specifying various parameters to specify partition values, set queue usage limits, and specify appropriate alerts and actions.

IMS Queue Control Facility provides three methods for implementing queue overflow protection. Each subsequent method provides more capability and flexibility in configuring queue overflow protection.

Type 1
   Queue space notification exit (DFSQSPC0), which is available in your IMS nonshared-queues system

Type 2
 Queue overflow protection using threshold settings to define queue space partitions in your IMS nonshared-queues system

Type 3
 Queue overflow protection using area and fail-safe settings to define queue space partitions in your IMS nonshared-queues system

Message processing

You can use IMS Queue Control Facility to process messages from any of the following locations:

- IMS system log
- IMS queue manager message queues
- Shared queues (in a shared-queues environment only)

You can process selected messages based on the criteria that you specify in control statements. IMS Queue Control Facility produces an output data set and a data report that includes quantitative information about the selected messages.

IMS Queue Control Facility supports all types of messages except for Fast Path messages.

Partial support exists for conversational messages. IMS Queue Control Facility requeues conversational messages; however, the conversation is not restarted. Partial support allows the conversational transaction to proceed for an additional iteration.

Recovery and maintenance functions

IMS Queue Control Facility functions are designed to help you perform the following recovery and maintenance tasks:

- Message queue recovery after either a planned or unplanned system outage
- Application recovery when it is practical to return messages to the message queue for reprocessing
- IMS maintenance for shared- and nonshared-queues environments

  You can query, browse, unload, and load IMS messages from, or to, the IMS message queues.

- Load messages to a new IMS release (for migration) or to an old IMS release (for fallback)

  Messages can be loaded to other IMS systems for testing, offloading, or for recovery purposes.

- Queue overflow protection for nonshared queues

  Queue overflow protection monitors queue usage, detects the source of queue overflow, and takes action to prevent your queue usage from reaching critical thresholds.

Message requeuing

Two circumstances require that messages be requeued:

- When a system redefinition or failure requires a cold start and messages are lost from the IMS message queue without being processed
- When messages are incorrectly processed because of an application program error

In both of these cases, messages must be returned to the IMS message queue if the messages are to be processed or reprocessed correctly.

- In the nonshared-queues environment, IMS Queue Control Facility gets messages that need to be requeued from the system log data sets (SLDS) or the IMS queue manager message queues.
- In the shared-queues environment, IMS Queue Control Facility gets messages from the cold queue; a special queue for messages that were in process when IMS terminated abnormally and was cold started. Recovery in the shared-queues environment only applies to messages on the cold queue.

You can use IMS Queue Control Facility to maintain the message queues to prevent performance degradation, to save space in the shared queues, and to prevent a buffer shortage.

Queue maintenance

Maintenance of your message queues involves examining the queues and removing only those messages that are no longer needed.

In the shared-queues environment, messages that were in process stay on the cold queue and accumulate until you remove them. This accumulation of messages degrades performance and wastes space on the shared queues.

In the nonshared-queues environment, messages can use enough buffer resource to cause a buffer shortage. A buffer shortage can cause IMS to abnormally terminate with a user 758 abend code.

TSO client / ISPF interface

You can use the IMS Queue Control Facility TSO client (ISPF interface) to perform the following tasks:

- Select the server and IMS subsystem to be used
- Show the IMS environment that you are using
- Show your local queue status
- Query messages and list destinations with queued messages
- Unload (delete) messages from the IMS message queue
- Load or reload messages onto the IMS message queues
- Maintain the tables that are associated with queue overflow protection

You can use the IMS Queue Control Facility TSO client to perform the following queue overflow protection tasks:

- List and perform operations on nonshared-queues environment waited tasks
- View, modify, and load table overflow parameters
- Modify queue space utilization notification parameters

IMS Queue Control Facility terminology

IMS Queue Control Facility includes several unique terms that you should understand before you begin to use IMS Queue Control Facility.

Functions
Functions are the tasks that IMS Queue Control Facility performs for you: BROWSE, LOAD, QUERY, RECOVER, REPROCESS, and UNLOAD. The functions are also used as the keywords that you specify in control statements.

Procedures
Procedures are sets of tasks to accomplish a function of IMS Queue Control Facility.

Procedures are usually called PROCs and are also reusable components that contain application program control statements and parameter definitions.

Routines
Routines are the application program exit routines or other internal logic components that are called in order to perform the functions of IMS Queue Control Facility.

Jobs
Jobs are JCL job streams that are made up of different cards, control statements, and optionally symbolic parameters, and comments. These JCL job streams are the jobs that you submit to run the IMS Queue Control Facility functions in a true batch mode.

Service updates and support information

Service updates and support information for this product, including software fix packs, PTFs, frequently asked questions (FAQs), technical notes, troubleshooting information, and downloads, are available from the web.

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

IBM Support: IMS Queue Control Facility for z/OS

Product 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 Notifications service.

Information on the web

Always refer to the IMS Tools Product Documentation web page for complete product documentation resources:

https://www.ibm.com/support/pages/node/712955

The IMS Tools Product Documentation web page includes:

- Links to IBM Documentation for the user guides ("HTML")
- PDF versions of the user guides ("PDF")
- Program Directories for IMS Tools products
- Recent updates to the user guides, referred to as "Tech docs" ("See updates to this information!")
- Technical notes from IBM Software Support, referred to as "Tech notes"
- White papers that describe product business scenarios and solutions

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

http://www.redbooks.ibm.com

The IBM Information Management System website shows how IT organizations can maximize their investment in IMS databases while staying ahead of today's top data management challenges:

https://www.ibm.com/software/data/ims/

Receiving documentation updates automatically

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

To register with the My Notifications service:

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

How to send your comments

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

- Click the Feedback button at the top of the IBM Documentation topic that you are commenting on.
- Click the Contact Us tab at the bottom of any IBM Documentation topic.
- Send an email to ibmdocs@us.ibm.com. Be sure to include the book title, topic or section title, specific text, and your comment.

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

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 this product enable users to perform the following activities:

- 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 the ISPF interface, 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. Product architecture and business scenarios

The following topics provide additional background information for using IMS Queue Control Facility.

Topics:

IMS Queue Control Facility architecture

IMS Queue Control Facility consists of several components. Some of these components are initiated by your input on the ISPF interface, and others are initiated by JCL jobs that you must customize for your environment.

The main components of IMS Queue Control Facility include:

- The IMS Queue Control Facility server
- Queue Threshold and Queue Space Usage Notification
- Extensions
- The queue structures
- The ISPF front-end interface
- Batch jobs, PROCs, and control members that contain control statements

The following diagram depicts the relationships of the main components of IMS Queue Control Facility.
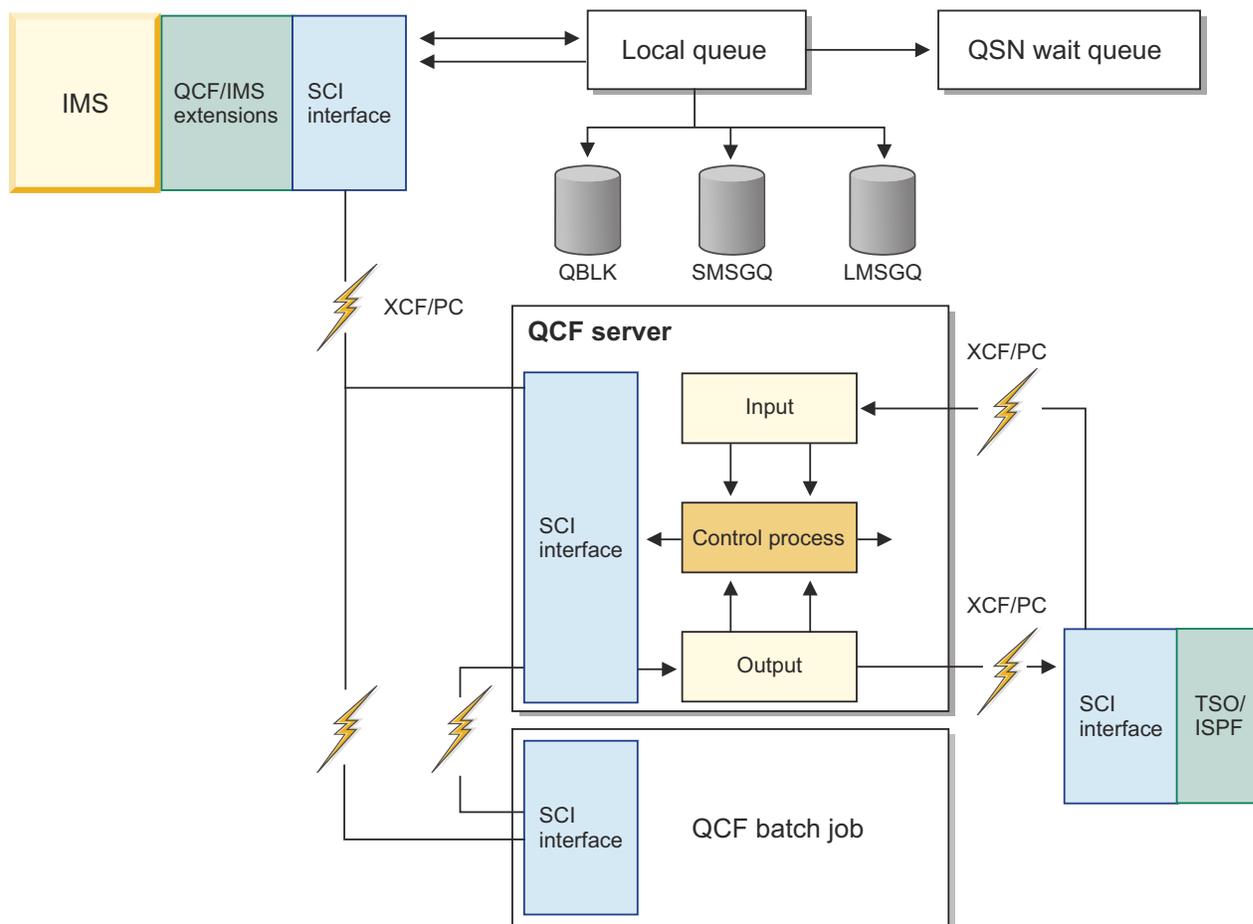
Figure 1. The main components of IMS Queue Control Facility

IMS Queue Control Facility consists of a batch address space, a TSO client address space, a server address space, and an IMS Queue Control Facility extension (which runs in the IMS control region address space).

Communication between the various components is through the IMS structure call interface (IMS SCI). The IMS Queue Control Facility extension performs the actual manipulation of the IMS queues.

You can access the extensions through either a batch interface (using JCL) or through a TSO client/server interface.

Overview of IMS queue structures

The following topic provides you with an understanding of the IMS queue structures in the shared- and nonshared-queues environments.

In both the shared- and nonshared-queues environments, messages are held in specific areas for processing. These areas comprise the following queues:

- Local queue for shared and nonshared queues
- Primary queue for shared queues
- Overflow queue for shared queues
- Queue space notification (QSN) wait queue for nonshared queues
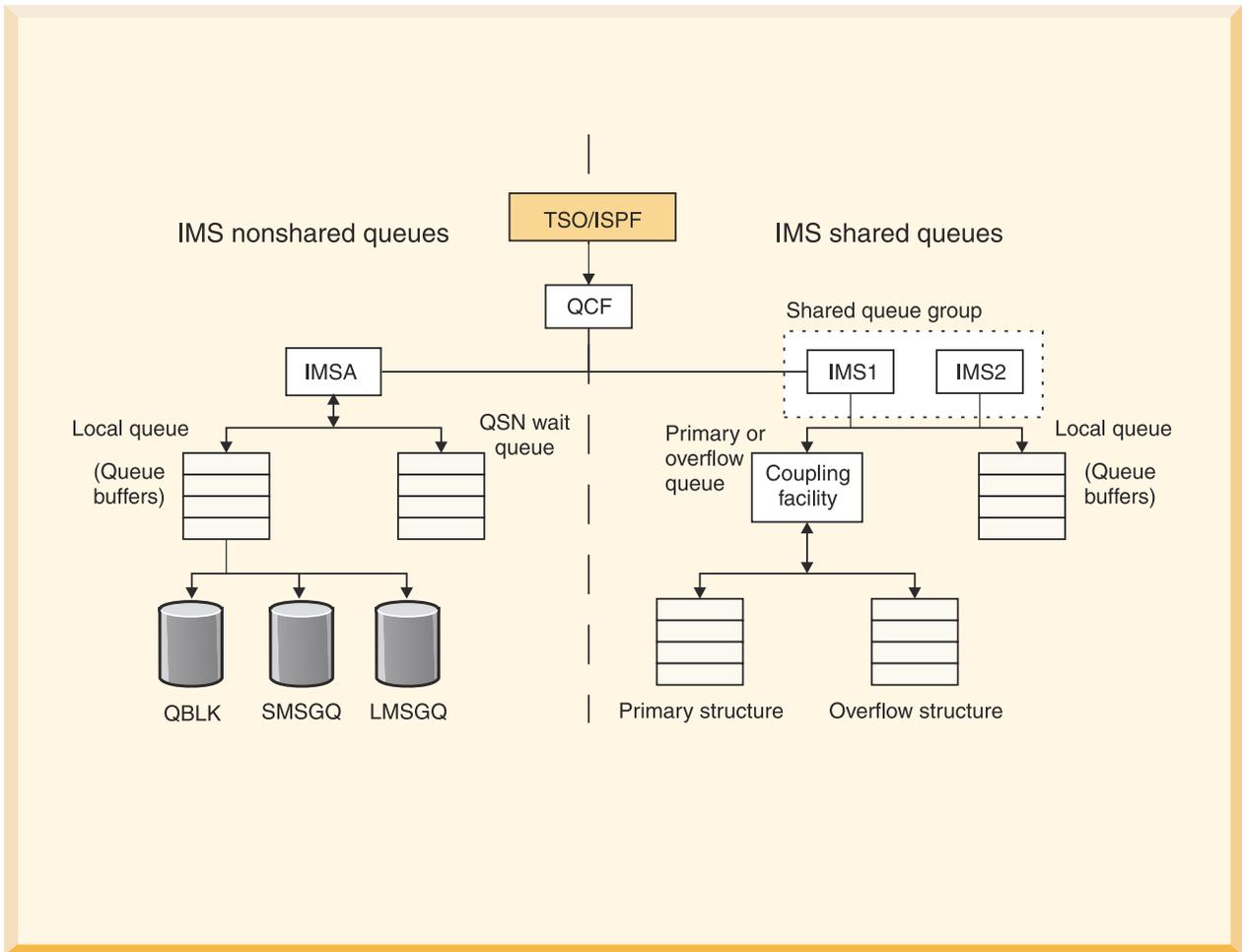
Figure 2. IMS queue structures

Local queue for nonshared queues

In a nonshared-queues environment, messages are initially placed in a working storage queue buffer pool in the local (nonshared queues) IMS A. IMS A writes the messages either to the short message queue (SHMSG) data set or to the large message queue (LGMSG) data set.

Messages that are queued to logical terminals (LTERMS), to Multiple System Coupling (MSC) MSNAMES, to APPC lunames, or to OTMA tpipenames are pointed to these locations by queue blocks (QBLKS), which are written to a QBLK data set. Both the working storage buffers and data sets comprise the local queues.

A queue type of DEADQ allows you to identify when all of the following conditions exist:

- All of the LTERM destinations were created in an IMS ETO environment
- All of these LTERM destinations contain messages that have been on these LTERM destinations for the amount of time that was specified on the DLQT run parameter (1 - 365 days).

These Dead Letter Queue messages were returned from a run of the QUERY LIST function from the destinations panel.

Local queue (shared queues)

In a shared-queues environment, messages are initially placed in a working storage queue buffer pool in the IMS that receives the message.

When the message is complete, the message is written to one of the shared queue structures in the coupling facility, either the primary queue or the overflow queue.

When the message is queued to be processed by one of the IMS subsystems in the shared queue group, the message is again read into one of the local queue buffers. These working storage buffers comprise the local queue in a shared queue IMS.

Primary message queue structure (shared queues)

Most messages in a shared-queues environment are placed on the primary message queue to be processed by one of the IMS subsystems in the shared queue group. This queuing is done through an IMS component called the Common Queue Server (CQS), which writes the messages to the coupling facility of the sysplex.

The coupling facility is referred to as the primary queue by IMS. Messages are queued by their destination names and types; the IMS systems in the shared queue group that process the message types and destinations are notified that messages to be processed are on the coupling facility.

Overflow queue structure (shared queues)

When the primary queue reaches a threshold, messages are written to the overflow queue in the coupling facility. The overflow queue is a single queue of messages for all destinations and destination types. When the primary queue has space again, messages are written to the primary queue instead of to the overflow queue.

IMS writes messages to the overflow queue by destination. When a destination is selected for overflow, all of its messages are moved from the primary to the overflow queue and the destination stays in overflow mode until the threshold is resolved. Other destinations might stay in a normal primary queue mode.

Sample business scenario

You can use IMS Queue Control Facility to help you to solve many IMS message queue management issues. The following scenario illustrates three IMS Queue Control Facility control regions in a sysplex environment.

To adequately manage your message queues you need to have three IMS Queue Control Facility control regions in a sysplex environment.

You want one instance of IMS Queue Control Facility server model on an MVS™ system to connect to a first, second, and third IMS subsystem.

You can have the IMS Queue Control Facility subsystem communicate with one or more of the IMS Queue Control Facility servers on other IMS subsystems using XCF/PC to communicate between SCI interfaces.

You also want your IMS Queue Control Facility batch job steps to have the ability to communicate with an IMS Queue Control Facility server on an IMS subsystem using XCF/PC to communicate between SCI interfaces.
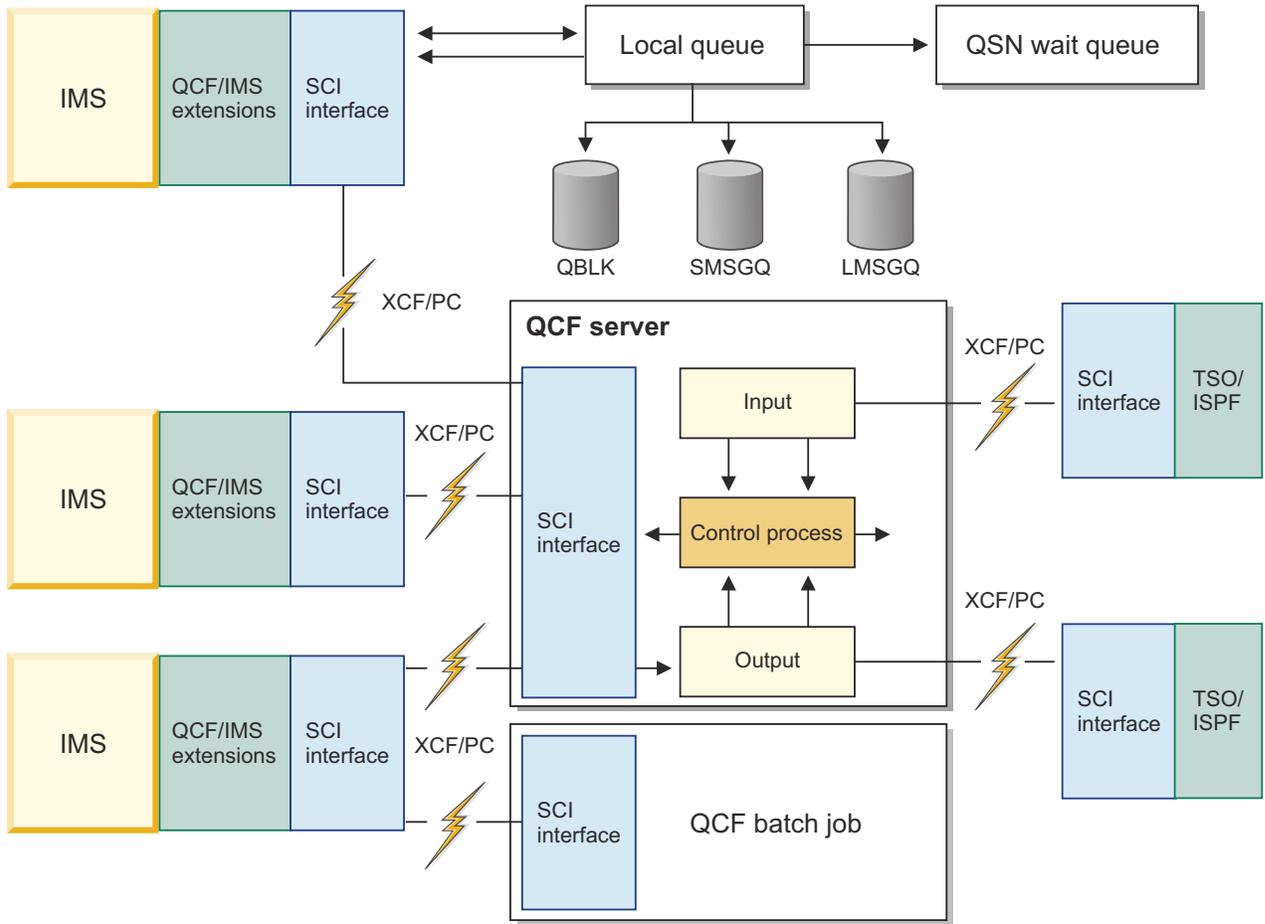
Figure 3. IMS Queue Control Facility control regions in a sysplex environment

Part 2. Configuring IMS Queue Control Facility

The topics in this section provide the detailed information that you require to set up, configure, and start IMS Queue Control Facility.

Topics:

Chapter 3. Setting up IMS Queue Control Facility

You must perform the following procedures to correctly set up IMS Queue Control Facility in your environment.

It is recommended that you perform the procedures in the order listed in this section.

Chapter 5, "Server and extension reference," on page 55 provides supplemental detail on PROCLIB members used during the setup procedures.

Topics:

1. Creating the SCI address space

IMS Queue Control Facility requires the IMS Structured Call Interface (SCI) facility for communication between the IMS Queue Control Facility extensions member and the IMS Queue Control Facility server.

The IMS Queue Control Facility extensions member is responsible for manipulating IMS message queues and transmitting the output data to the IMS Queue Control Facility server by way of the IMS SCI.

The IMS SCI resides in its own Address Space and provides the communication channel between the IMS Queue Control Facility extensions member, which resides in the IMS Control Region, and the IMS Queue Control Facility server, which resides in its own address space. One copy of the SCI is required in each logical partition (LPAR) where an IMS system has enabled the IMS Queue Control Facility extensions.

Important: In this topic, the term *IMS Queue Control Facility extensions* refers to the IMS Queue Control Facility portion of the IMS system, and the term *IMS system* refers to all other functions of the IMS system that might enable the SCI. This distinction can help clarify when IMS systems and IMS Queue Control Facility extensions have a dependency on the SCI address space. An IMS system is *enabled for SCI* when it is using the IMS Operations Manager, Resource Manager, and other sysplex functions.

Note: For a multiple LPAR environment, the information in this discussion applies to each LPAR.

For more background information on the creating the SCI address space, refer to "Considerations for creating the SCI address space" on page 49.

Connecting IMS Queue Control Facility to SCI

Both IMS Queue Control Facility extensions and the IMS Queue Control Facility server connect to the Structured Call Interface (SCI) facility. Both the extensions and the sever must have completed connection processing for IMS Queue Control Facility to be fully connected to the SCI.

IMS Queue Control Facility is connected to the SCI and all IMS Queue Control Facility functions are fully functional only when the following message is displayed:

```
IQC6001I QCF EXTENSION ENABLED
```

IMS Queue Control Facility extensions allow you to start the SCI job and to specify the SCI job name that is started by IMS Queue Control Facility extensions when the SCIPROG= statement is specified in the IMS Queue Control Facility extensions configuration statements.

See "7. Enabling and configuring the IMS Queue Control Facility extensions" on page 31.

If message `IQC6900I QCF IMS EXTENSION WAITING FOR SCI: (PLEX NAME: plex_name)` is issued, the following conditions might be possible:

- The SCI was not started
- The SCI job failed after you specified the SCI job name in the IMS Queue Control Facility extensions configuration

  In this case, you must start the SCI job with an MVS start command, or submit the PROC from the PROCLIB to allow IMS Queue Control Facility extensions and the IMS Queue Control Facility server to connect to the SCI.

If message `IQC6915I NO SCI PROC SPECIFIED` is issued, then you must start the SCI job with an MVS start command or submit the PROC from the PROCLIB.

If the SCI terminates and is restarted (with an MVS start command or by submitting the PROC from the PROCLIB), the IMS Queue Control Facility server and IMS Queue Control Facility extensions will automatically reconnect.

Refer to (and modify if required) the SCI start JCL procedure (IQCSCI), the SCI BPE configuration file (BPECONF1), and the SCI INIT configuration file (CSLSI000) in the IMS Queue Control Facility *hlq*.SIQCSAMP data set.

SCI address space configuration scenarios

The following options are possible for creating the SCI address space during IMS Queue Control Facility setup:

- Option 1: If an IMS system or systems is enabled for SCI and IMS Queue Control Facility is installed, a single SCI address space can be shared by the IMS system or systems and the IMS Queue Control Facility extensions.
- Option 2: If an IMS system or systems is enabled for SCI and IMS Queue Control Facility is installed, a separate SCI address space can be created and dedicated to each component.
- Option 3: If an IMS System is not enabled for SCI and IMS Queue Control Facility is installed, a separate SCI address space must be created and dedicated to support the IMS Queue Control Facility extensions only.

Example configuration scenarios:

- (One or more IMS systems, per LPAR) - If you currently have one or more IMS systems in a single LPAR, and none of the IMS systems are currently using any IMS function that requires the SCI address space, then you would not have previously enabled the SCI address space for that LPAR.

  If one or more of the IMS systems will now be using the IMS Queue Control Facility extensions, you will be required to have one SCI address space for that LPAR which will provide communications between the IMS Queue Control Facility extensions and the IMS Queue Control Facility server.

- (One or more IMS systems, per LPAR) - If you currently have one or more IMS systems in a single LPAR, and one or more of the IMS systems are currently using the IMS Operations Manager or IMS Resource Manager which requires the SCI address space for that LPAR, then you currently have enabled the SCI address space for that LPAR.

  If one or more of the IMS systems will now also be using the IMS Queue Control Facility extensions, you can share the SCI address space for that LPAR which will provide communications between the IMS Queue Control Facility extensions and the IMS Queue Control Facility server, and between the IMS systems and the IMS Operations Manager or IMS Resource Manager.

- (One or more IMS systems, per LPAR) - If you currently have one or more IMS systems in a single LPAR, and one or more of the IMS systems are currently using the IMS Operations Manager or IMS Resource

Manager which requires the SCI address space for that LPAR, then you currently have enabled the SCI address space for that LPAR which will provide communications between the IMS systems and the IMS Operations Manager or IMS Resource Manager.

If one or more of the IMS systems will now also be using the IMS Queue Control Facility extensions, you can share the SCI address space for that LPAR, or you can choose to start a second SCI address space for the same LPAR which will provide communications between the IMS Queue Control Facility extensions and the IMS Queue Control Facility server.

SCI address space startup JCL

Because SCI is shipped with IMS, you need to create only the SCI address space startup JCL and the SCI/BPE configuration members.

Be aware of the following conditions:

- IMS Queue Control Facility does not require your current IMS systems to be enabled for SCI.
- IMS Queue Control Facility does not require Operations Manager or Resource Manager to be enabled.

To create the SCI address space, use the following example of an SCI address space startup JCL. In this example, the startup JCL is presented as a console startable procedure that can be added to a PROCLIB and modified to run as a job.

This example (IQCSCI) is shipped as a sample in *hlq*.SIQCSAMP data set.

```
//*---------------------------------------------------------------------
//*
//* DESCRIPTIVE NAME = SAMPLE PROCEDURE - WILL BE STARTED FROM QCF
//*                    QSN WHEN QCF FAILS TO CONNECT TO THE SCI.
//*
//* STATUS = VERSION 03  RELEASE 02  MODIFICATION LEVEL 00
//*
//* FUNCTION = SCI ADDRESS SPACE START UP PROCEDURE
//*
//*---------------------------------------------------------------------
//*      Licensed Materials - Property of IBM
//*
//*      IMS QCF for z/OS Version 3 Release 2
//*
//*      5697-N50
//*
//*      Copyright IBM Corp. 2000, 2014
//*
//*      All Rights Reserved.
//*      US Government Users Restricted Rights -
//*      Use, duplication or disclosure restricted
//*      by GSA ADP Schedule Contract with IBM Corp.
//*
//*---------------------------------------------------------------------
//*************************************************************/
//* NOTES =                                                  */
//* 1) REVIEW ALL STATEMENTS BEFORE SUBMITTING THIS JOB.     */
//* 2) CHANGE hlqual FIELDS TO VALID HIGH LEVEL QUALIFIERS    */
//*    THAT COMPLY WITH YOUR SITES NAMING STANDARDS.         */
//*    CHANGE imshlqual FIELDS TO VALID HIGH LEVEL QUALIFIERS */
//*    FOR IMS LOAD LIBRARY.                                 */
//* 3) CHANGE BPECONF1 IQCBPECF TO YOUR CONFIGURATION        */
//*    MEMBER CREATED IN PROCLIB                             */
//* 4) CHANGE SCIINIT=000 TO YOUR CONFIGURATION MEMBER       */
//*    CREATED FOR THE SCI INIT CONFIGURATION WHERE 000 IS   */
//*    THE THREE DIGIT SUFFIX ON CSLSInnn                    */
//*                                                          */
//*---------------------------------------------------------------------
//IQCSCI   PROC
//*
//STEP1    EXEC PGM=BPEINI00,REGION=6M,
//         PARM='BPECFG=IQCBPECF,BPEINIT=CSLSINI0,SCIINIT=nnn'
//*
//STEPLIB  DD  DSN=imshlqual.RESLIB,DISP=SHR
//PROCLIB  DD  DSN=hlqual.PROCLIB,DISP=SHR
//SYSPRINT DD  SYSOUT=A
//SYSUDUMP DD  SYSOUT=A                                   //*
//*---------------------------------------------------------------*
```

```
//* End of Procedure SCI  ..........                              *
//*----------------------------------------------------------------*
```

Parameters for SCI address space startup procedure

SCI runs in the Base Primitive Environment (BPE). SCI requires a BPE configuration member in a partitioned data set.

BPEINIT=*bpe_user_product_initialization_parameters_module*

Specify the name of the BPE user product initialization parameters module. BPE address space initialization uses the initialization values defined in this module to configure a user product address space.

A setting of BPEINIT=CSLSINI0 contains the initialization values to create a Structured Call Interface address space.

BPECFG=*bpe_config_proclib_mbr*

Specify an 8-character name for a BPE configuration PROCLIB member to define various BPE attributes, such as trace table levels and user exit specifications. This parameter can only be specified as a runtime parameter.

If you want to change the default BPE attributes, you must create a BPE configuration PROCLIB member for your address spaces. Optionally, you can create different BPE configuration PROCLIB members for each address space.

This parameter is optional. If you do not specify a PROCLIB member for your BPE configuration, BPE uses the following default values for all of its parameters:

- No user exits

- A trace level of error

- US English as the language

The *hlq*.SIQCSAMP (IQCBPECF) member contains an example of a BPE configuration. Use the sample member unless you want to set different trace values or to define exits. For example:

```
***************************************************************/
*                                                            */
* DESCRIPTIVE NAME = SAMPLE SCI BPE CONFIGURATION SAMPLE      */
*                 START UP MEMBER                             */
* STATUS = VERSION 03  RELEASE 02  MODIFICATION LEVEL 00      */
*                                                            */
* FUNCTION = BPE CONFIGURATION MEMBER FOR IQCSCI PROC         */
*          MEMBER                                             */
*                                                            */
***************************************************************/
*      Licensed Materials - Property of IBM                  */
*                                                            */
*      IMS QCF for z/OS Version 3 Release 2                   */
*                                                            */
*      5697-N50                                               */
*                                                            */
*      Copyright IBM Corp. 2000, 2014                         */
*                                                            */
*      All Rights Reserved.                                   */
*      US Government Users Restricted Rights -                */
*      Use, duplication or disclosure restricted              */
*      by GSA ADP Schedule Contract with IBM Corp.            */
*                                                            */
***************************************************************/
* NOTES =                                                     */
* 1) THIS MEMBER IS USED AT SCI START UP                      */
*    CONTROL REGION                                           */
* 2) STORE THE MEMBER IN DATA SET POINTED BY PROCLIB DD       */
*    IN IMS CONTROL REGION START UP PROCEDURE                 */
* 3) ADD CQS AND/OR OM/RM/SCI TRACE LEVEL (TRCLEV) STATEMENTS*/
*    AS NEEDED BY YOUR INSTALLATION*/
***************************************************************/
*----------------------------------------------------------------*
*                                                                *
*     SCI BPE Initialization Proclib member                      *
*                                                                *
```

```
*-----------------------------------------------------------------*
LANG=ENU                            /* LANGUAGE FOR MESSAGES      */
                                    /* (ENU = U.S. ENGLISH)       */

#
# DEFINITIONS FOR BPE SYSTEM TRACES
#

TRCLEV=(*,HIGH,BPE)                 /* Default all traces to high */

# Note: Keep the following for compatibility with 6.1 BPE

TRCLEV=(STG,HIGH,BPE)               /* STORAGE TRACE              */
TRCLEV=(CBS,HIGH,BPE)               /* CONTROL BLK SRVCS TRACE    */
TRCLEV=(DISP,HIGH,BPE)              /* DISPATCHER TRACE           */
TRCLEV=(AWE,HIGH,BPE)               /* AWE SERVER TRACE           */
TRCLEV=(LATC,HIGH,BPE)              /* Latch Trace                */
TRCLEV=(SSRV,HIGH,BPE)              /* SYSTEM SERVICES TRACE      */
TRCLEV=(USRX,HIGH,BPE)              /* User Exit Services Trace   */

#
# DEFINITIONS FOR IQC TRACES
#

TRCLEV=(*,HIGH,IQC)                 /* Defalt all traces to high  */

#
# DEFINITIONS FOR CQS TRACES CAN BE ADDED HERE
#

#
# DEFINITIONS FOR OM/RM/SCI TRACES CAN BE ADDED HERE
#
```

SCIMBR=

SCI also requires its own configuration member in a partitioned data set.

The member name must have the following format:

```
CSLSIxxx
```

where *xxx* is a suffix value (see the SCIMBR parameter).

The *hlq*.SIQCSAMP (IQCSCI) member contains an example of a SCI initialization configuration.

```
*-----------------------------------------------------------------
*
* DESCRIPTIVE NAME = SAMPLE SCI CONFIGURATION MEMBER
*
* STATUS = VERSION 03  RELEASE 02  MODIFICATION LEVEL 00
*
* FUNCTION = SCI CONFIGURATION USED BY THE IMS SCI ADDRESS SPACE
*
*-----------------------------------------------------------------
*      Licensed Materials - Property of IBM
*
*      IMS QCF for z/OS Version 3 Release 2
*
*      5697-N50
*
*      Copyright IBM Corp. 2000, 2014
*
*      All Rights Reserved.
*      US Government Users Restricted Rights -
*      Use, duplication or disclosure restricted
*      by GSA ADP Schedule Contract with IBM Corp.
*
*-----------------------------------------------------------------
*****************************************************************/
* NOTES =                                                     */
*                                                             */
* 1) REVIEW THE IMS SCI INITIALIZATION DEFINITION TO          */
*    INSURE THAT THE DEFINITION IS CORRECT                    */
* 2) CHANGE THE PROC MEMBER NAME IQCSCI TO CSLSxxx         */
* 3) REVIEW ALL STATEMENTS                                    */
* 4) CHANGE QCFSCI TO YOUR INSTALLATION SCI NAME              */
* 5) CHANGE IPLEX TO YOUR INSTALLATION IMSPLEX NAME           */
```

```
*                                                        */
*-------------------------------------------------------------------
*-------------------------------------------------------------------*
*                                                                   *
* SCI Initialization Proclib Member.                                *
*                                                                   *
*-------------------------------------------------------------------*

ARMRST=N,
SCINAME=QCFSCI,
IMSPLEX(NAME=IPLEX)
```

2. Creating configuration members for message processing

These procedures describe how to allocate and initialize the data sets that are required for message processing in a nonshared-queues environment.

You must create the configuration members for the following types of message processing:

- Queue overflow protection processing
- Committed messages processing
- Uncommitted messages processing
- Threshold processing

Conditions for these tasks:

- These tasks are performed by the IQCSTRTJ sample job.
- Perform these tasks regardless of the overflow protection approach you select.
- If you are running nonshared queues, you must perform the configuration steps described in this section.
- If you are running shared queues, you can skip this section.

Sample members for these tasks can be found in the sample library *hlqual*.SIQCSAMP.

Allocating data sets for message processing

Allocate three partitioned data sets for the following parameters:

- Queue overflow protection data set - QSUN notification parameters
- Commit data set - QSN committed message processing parameters
- Uncommit data set - QSN uncommitted message processing parameters

You can customize sample JCL IQCSTRTJ in *hlqual*.SIQCSAMP to create these data sets and put the supplied sample members in the overflow protection data sets.

Member IQC#####, which is located in *hlqual*.SIQCSAMP, lists all samples, ordered by their functionality.

Specify DCB RECFM=FB and LRECL=80 (follow local standards).

Unless you specify otherwise, as described later in this section, the IMS Queue Control Facility assumes that there are separate data sets for each IMS on each MVS system.

If you allocate separate data sets for each IMS on each MVS system, the naming convention of these data sets is as follows:

Data set	Naming convention	Example
Queue overflow protection data set	*ovflhlq.mvsname.imsid*.NOTPARMS	IQC.MVS1.IMSA.NOTPARMS
Commit data set	*ovflhlq.mvsname.imsid*.COMMIT	IQC.MVS1.IMSA.COMMIT MIT

Data set	Naming convention	Example
Uncommit data set	*ovflhlq.mvsname.imsid*.UNC OMMIT	IQC.MVS1.IMSA.UNCOMMIT

where:

- *ovflhlq* can be up to 25 characters long. The default is IQC.
- *mvsname* is the MVS system name.
- *imsid* is the name of the IMS control region.

If your MVS system is named such that the name would cause an invalid data set name to be created, the following rules apply:

- If the MVS system name is less than 8 characters, an *S* will be added to the beginning portion of the data set name that contains the MVS system name.

  For example, if your MVS system name is 123456, and your IMSID is IMSA, the commit data set must be named IQC.S123456.IMSA.COMMIT.
- If the MVS system name is 8 characters, an *S* will be added to the beginning of the first 7 characters of the MVS system name and an additional qualifier, beginning with an *S* followed by the 8th character, will be created.

  For example, if your MVS system name is 7654321@ and your IMSID is IMSB, the commit data set must be named IQC.S7654321.S@.IMSB.UNCOMMIT.

You can also define these three data sets in a way that they are shared among 1) all IMS subsystems on a given MVS system, 2) one IMS subsystem on all MVS systems, or 3) all IMS subsystems on all MVS systems. In these cases, the following naming rules apply:

- If you want the IMS Queue Control Facility to share the data sets among all IMS subsystems, omit *imsid* as the middle-level qualifiers of the data sets names.
- If you want the IMS Queue Control Facility to share the data sets among all MVS systems, omit *mvsname* as the middle-level qualifiers of the data sets names.

To share these data sets among all IMS subsystems on a given MVS system, specify their names as follows:

```
ovflhlq.mvsname.NOTPARMS
ovflhlq.mvsname.COMMIT
ovflhlq.mvsname.UNCOMMIT
```

To share these data sets among a single IMS subsystem on all MVS systems, specify their names as follows:

```
ovflhlq.imsid.NOTPARMS
ovflhlq.imsid.COMMIT
ovflhlq.imsid.UNCOMMIT
```

To share these data sets among all IMS subsystems on all MVS systems, specify their names as follows:

```
ovflhlq.NOTPARMS
ovflhlq.COMMIT
ovflhlq.UNCOMMIT
```

To use these data sets, create and populate them with *ovflhlq*, as set in the REXX program for starting the TSO client (sample: *hlq*.SIQCEXEC(IQC31)), with or without *mvsname* and *imsid* as the middle-level qualifier, and suffix NOTPARMS, COMMIT, or UNCOMMIT.

If these data sets are to be shared among IMS subsystems or MVS systems, follow these additional steps:

1. Start the TSO client. The IMS Queue Control Facility Main Menu panel is displayed.
2. Select **Preferences**.

3. Select option **1** (Change QCF session preferences).
4. Select **N** for **Include MVSNAME in DSNames**, **Include IMSID in DSNames**, or both. The default is **Y** for either field.

Initializing the type 3 queue overflow protection data set

To initialize the queue overflow protection data set, use the sample control statement *hlqual*.SIQCSAMP(IQCQSUNP) (the IBM-supplied default for the QSUN notification processing parameters) as a member in the NOTPARMS data set.

Descriptions for these samples can be found in member IQC#####.

You can use the QSN panels to modify the values in this member or to create new members.

Initializing the commit data set for types 2 and 3 queue overflow protection

To initialize the commit data set, use the sample FUNCTION control statement *hlqual*.SIQCSAMP(IQCQSNAB) as a member of the commit data set.

Descriptions for these samples can be found in member IQC#####.

These control statements will be input to the IMS Queue Control Facility client that was started by IMS when IMS message queue utilization goes from QSN area A to area B.

You can use the QSN panels to modify the values in this member or to create new members.

Initializing the uncommit data set for type 2 queue overflow protection

To initialize the uncommit data set, use the sample control statement *hlqual*.SIQCSAMP(IQCQSNUN) (the IBM-supplied default for the uncommitted message processing parameters) as a member in the uncommit data set.

Descriptions for these samples can be found in member IQC#####.

You can use the QSN panels to modify the values in this member or to create new members.

Threshold processing

By default, threshold processing is set to specify that no action is taken when the threshold is crossed.

For details on advanced threshold processing configuration, see "Configuring threshold processing" on page 51.

3. Modifying the program properties table

Add an entry in the z/OS program properties table (PPT) for IMS Queue Control Facility.

Procedure

To add the IMS Queue Control Facility entry in the z/OS program properties table, perform the following steps:

1. Edit the SCHED*nn* member of your SYS1.PARMLIB data set.
2. Add the IMS Queue Control Facility entry.

   The following example shows a z/OS program properties table with this entry:

```
PPT PGMNAME(IQCINI0$)      /* PROGRAM NAME = IQCINI0$              */
        CANCEL             /* PROGRAM CAN BE CANCELED             */
        KEY(7)             /* PROTECT KEY ASSIGNED IS 7           */
        NOSWAP             /* PROGRAM IS NON-SWAPPABLE            */
        NOPRIV             /* PROGRAM IS NOT PRIVILEGED           */
        DSI                /* REQUIRES DATA SET INTEGRITY         */
        PASS               /* CANNOT BYPASS PASSWORD PROTECTION   */
        SYST               /* PROGRAM IS A SYSTEM TASK            */
```

```
               AFF(NONE)          /* NO CPU AFFINITY                    */
               NOPREF             /* NO PREFERRED STORAGE FRAMES        */
```

3. To activate the updated program properties table, either perform an initial program load (IPL) on the z/OS system, or issue the z/OS SET SCH command.

4. APF authorize the program library

You must APF authorize the IMS Queue Control Facility load library for all of the IMS Queue Control Facility components to function.

Before you begin

The name of the IMS Queue Control Facility load library data set is *hlq*.SIQCLINK.

Procedure

To establish APF authorization, perform the following steps:

1. Add *hlq*.SIQCLINK to the parmlib PROG*xx* member.

   You can add *hlq*.SIQCLINK dynamically by issuing the following operator command:

   ```
   SETPROG APF,ADD,DSN=hlq.SIQCLINK,VOL=vvvvvv
   ```

2. To ensure that the library is APF authorized, add the IMS Queue Control Facility program library to the z/OS APF list.

3. To activate the updated APF list, either perform an initial program load (IPL) on the z/OS system, or issue the z/OS **SETPROG** command.

5. IMS SYSGEN/PSBGEN requirements

IMS Queue Control Facility 3.2 requires that a program specification block (PSB) be defined and generated in all target IMS control regions that IMS Queue Control Facility will be accessing.

About this task

The purpose of the PSB is to call the message queue services. The PSB is generated using the PSBGEN utility and must be present in the IMS PSBLIB.

This PSB name is defined in the IMS sysgen macro MSGQUEUE (keyword MRQPSBN).

The default name for this PSB is MRQPSB, but you can change the name of the PSB according to your IMS generation set name.

For more details, see *IMS System Definition*.

The following example code shows the sample JCL for generating MRQPSB. If you specified another PSB name with the MSGQUEUE macro, you must substitute this name in the JCL in place of MRQPSB.

```
//**************************************************************/
//* JOB NAME = IQCPSB                                         */
//*                                                           */
//* DESCRIPTIVE NAME = NEW PSB AND ACB INSTALLATION JCL FOR   */
//*                    THE IMS/ESA QUEUE CONTROL FACILITY     */
//*                                                           */
//* STATUS = VERSION 03  RELEASE 02  MODIFICATION LEVEL 00    */
//*                                                           */
//* FUNCTION = GENERATE THE PSB AND ACB                       */
//*                                                           */
//*     QCF VERSION 3 RELEASE 2                               */
//*     LICENSED MATERIALS - PROPERTY OF IBM                  */
//*     ALL RIGHTS RESERVED.                                  */
//*     US GOVERNMENT USERS RESTRICTED RIGHTS -               */
//*     USE, DUPLICATION OR DISCLOSURE RESTRICTED             */
//*     BY GSA ADP SCHEDULE CONTRACT WITH IBM CORP.           */
//*                                                           */
```

```
//* NOTES =                                                  */
//* 1) REVIEW ALL STATEMENTS BEFORE SUBMITTING THIS JOB.      */
//*                                                          */
//*  APAR    DATE    ID   DESCRIPTION                        */
//*                                                          */
//**************************************************************/
//*
//IQCPSB    JOB (ACCOUNTINFO),NORMAL JOBCARD INFO
//*
//         EXEC PSBGEN,MBR=MRQPSB
//C.SYSIN  DD *
ALTPCB01 PCB  TYPE=TP,MODIFY=YES
         PSBGEN  LANG=ASSEM,PSBNAME=MRQPSB
         END
/*
//         EXEC ACBGEN
//G.SYSIN  DD *
         BUILD PSB=MRQPSB
/*
```

6. Configuring and starting the IMS Queue Control Facility server

You must configure IMS Queue Control Facility server address space JCL and submit the job to start an IMS Queue Control Facility server address space.

Topics:

Creating the server address space JCL

You can customize the sample member IQCSSERV in *hlq*.SIQCSAMP to create the server JCL.

To create the server address space JCL, perform the following steps:

1. Edit the sample member IQCSSERV in *hlq*.SIQCSAMP.

   The following figure shows a sample of the JCL that is required to start an IMS Queue Control Facility server address space.

```
//**************************************************************/
//* JOB NAME = IQCSSERV                                        */
//*                                                          */
//* DESCRIPTIVE NAME = SAMPLE PROCEDURE TO START              */
//*                    QCF SERVER                             */
//*                                                          */
//* STATUS = VERSION 03  RELEASE 02  MODIFICATION LEVEL 00    */
//*                                                          */
//* FUNCTION = RUN QCF SERVER                                 */
//**************************************************************/
//*                                                          */
//*      Licensed Materials - Property of IBM                */
//*                                                          */
//*      IMS QCF for z/OS Version 3 Release 2                */
//*                                                          */
//*      5697-N50                                            */
//*                                                          */
//*      Copyright IBM Corp. 2004, 2014                      */
//*                                                          */
//*      All Rights Reserved.                                */
//*      US Government Users Restricted Rights -             */
//*      Use, duplication or disclosure restricted           */
//*      by GSA ADP Schedule Contract with IBM Corp.         */
//*                                                          */
//**************************************************************/
//* NOTES =                                                  */
//* 1) REVIEW ALL STATEMENTS BEFORE SUBMITTING THIS JOB.      */
//* 2) CHANGE hlqual    FIELDS TO VALID HIGH LEVEL QUALIFIERS */
//*    THAT COMPLY WITH YOUR SITES NAMING STANDARDS.          */
//*    CHANGE imshlqual FIELDS TO VALID HIGH LEVEL QUALIFIERS */
//*    FOR IMS LOAD LIBRARY.                                  */
//*    CHANGE iqchlqual FIELDS TO VALID HIGH LEVEL QUALIFIERS */
```

```
//*      FOR QCF LOAD LIBRARY.                                */
//* 3) CHANGE IQCBACNF and IQCSERVS TO YOUR CONFIGURATION      */
//*      MEMBERS CREATED IN PROCLIB                            */
//*                                                            */
//*  APAR    DATE     ID    DESCRIPTION                        */
//*                                                            */
//****************************************************************/
//IQCSSERV  PROC
//*********************************************************************
//*  BRING UP QCF SERVER                                             *
//*********************************************************************
//STEP1    EXEC PGM=IQCINI0$,REGION=5M,TIME=1440,
//            PARM='BPEINIT=IQCZINI0,BPECFG=IQCBACNF,IQCCFG=IQCSERVI'
//STEPLIB  DD  DSN=qcfhlqual.SIQCLINK,DISP=SHR
//         DD  DSN=imshlqual.RESLIB,DISP=SHR
//PROCLIB  DD  DSN=qcfhlqual.PROCLIB,DISP=SHR
//SYSPRINT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
```

2. Replace the italicized items and member names in the sample JCL with values that match your environment's specifications.

3. Code the parameters and data definition statements in the following sections on the IMS Queue Control Facility server procedure EXEC PARM= statement.

Parameters for server address space startup procedure

BPEINIT=*bpe_user_product_initialization_parameters_module*

Specify the name of the BPE user product initialization parameters module. BPE address space initialization will use the initialization values defined in this module to configure a user product address space.

A setting of BPEINT=IQCZINIO contains the initialization values to create an IMS Queue Control Facility server address space.

BPECFG=*bpe_config_proclib_mbr*

Specify an 8-character name for an optional BPE configuration PROCLIB member to define various BPE attributes, such as trace table levels and user exit specifications. This parameter can only be specified as a runtime parameter.

If you want to change the default BPE attributes, you must create a BPE configuration PROCLIB member for your address spaces. Optionally, you can create different BPE configuration PROCLIB members for each address space.

This parameter is optional. If you do not specify a PROCLIB member for your BPE configuration, BPE uses the following default values for all of its parameters:

- No user exits
- A trace level of error
- US English as the language

The *hlq*.SIQCSAMP (IQCBACNF) member contains an example of a BPE configuration. You can use this sample member, or create your own member if you want to set different trace values or to define exits.

IQCCFG=*qcf_config_proclib_mbr*

Specify an 8-character name for your IMS Queue Control Facility server configuration PROCLIB member. This parameter must be specified as a run parameter.

The *hlq*.SIQCSAMP(IQCSERVI) member contains a sample IMS Queue Control Facility server configuration.

This parameter is optional.

If you do not specify a PROCLIB member, the IMS Queue Control Facility server uses default values for all parameters.

In the following example, the IQCCFG parameter is not specified. You are notified that IQCCFG is missing and that it is using a default value:

```
IQCCFG= NOT SPECIFIED ON STARTUP PARMS
DEFAULT BEING USED
```

If the default PLEXNAME does not match the SCI PLEXNAME, the server will be unable to connect to SCI and you will receive the following message:

```
QCF SERVER WAITING FOR SCI
```

Specifying data definition statements

Specify the following data definition (DD) statements to set up your IMS Queue Control Facility server configuration JCL stream.

PROCLIB DD
Specify this required parameter to indicate which partitioned data sets contain your BPE configuration PROCLIB member.

Specify the BPE configuration PROCLIB member as BPECFG=*parameter*.

Specify the IMS Queue Control Facility server configuration PROCLIB member as IQCCFG=*parameter*.

The PROCLIB DD also contains members of the re-queue JCL and the offload JCL for the IMS Queue Control facility START TASK command.

STEPLIB DD
Ensure that the IMS Queue Control Facility authorized libraries are included in the IMS Queue Control Facility server startup JCL.

Example BPE PROCLIB member

The IMS Queue Control Facility code that runs in the IMS Queue Control Facility server or batch address space uses Base Primitive Environment (BPE) services.

The following figure shows a sample IMS Queue Control Facility BPE configuration PROCLIB member:

```
**********************************************************************
*  SAMPLE CONFIGURATION FILE FOR BPE WITH QCF SERVER ADDRESS SPACE    *
**********************************************************************
*---------------------------------------------------------------------*
* Specify the language for messages.  Syntax is:                      *
*                                                                     *
*   LANG=language                                                     *
*                                                                     *
* language = 3-character code for the requested language.  Currently, *
*            only ENU (U.S. English) is supported.                    *
*---------------------------------------------------------------------*
LANG=ENU                                 /* Language for messages    */
                                         /* (ENU = U.S. English)     */
*---------------------------------------------------------------------*
* Trace table level definitions.  Syntax is:                          *
*                                                                     *
*   TRCLEV=(tablename,level,component)                       -or-     *
*   TRCLEV=(tablename,level,component,PAGES=#pages)                   *
*                                                                     *
* tablename = name of the trace table, or * to set default for all    *
* level     = tracing level: NONE, ERROR, LOW, MEDIUM, or HIGH        *
* component = owning component name (BPE or QCF)                      *
* #pages    = (optional) # of 4K pages to be allocated to this trace  *
*             table                                                   *
*---------------------------------------------------------------------*
#
# Definitions for BPE system traces
#
TRCLEV=(*,HIGH,BPE)                       /* Set default for all BPE  */
                                         /* traces to HIGH.  Uncomment */
                                         /* lines below and fill in   */
                                         /* level to change particular */
                                         /* table levels to other than */
```

```
                                          /* HIGH.                    */
#TRCLEV=(AWE,level,BPE)                    /* AWE server trace         */
#TRCLEV=(CBS,level,BPE)                    /* Control blk srvcs trace  */
#TRCLEV=(CMD,level,BPE)                    /* Command services trace   */
#TRCLEV=(DISP,level,BPE)                   /* Dispatcher trace         */
#TRCLEV=(HASH,level,BPE)                   /* Hash table trace         */
#TRCLEV=(LATC,level,BPE)                   /* Latch trace              */
#TRCLEV=(SSRV,level,BPE)                   /* System services trace    */
#TRCLEV=(STG,level,BPE)                    /* Storage trace            */
#TRCLEV=(USRX,level,BPE)                   /* User exit services trace */
#
# Definitions for QCF traces
#
TRCLEV=(*,HIGH,IQC)                        /* Set default for all QCF  */
                                          /* traces to HIGH.  Uncomment */
                                          /* lines below and fill in  */
                                          /* level to change particular */
                                          /* table levels to other than */
                                          /* HIGH.                    */
#TRCLEV=(ERR,level,IQC)                    /* QCF error trace          */
#TRCLEV=(INI,level,IQC)                    /* QCF initialization trace */
#TRCLEV=(MST,level,IQC)                    /* QCF master control trace */
#TRCLEV=(CSV,level,IQC)                    /* QCF common service trace */
#TRCLEV=(CVC,level,IQC)                    /* QCF conversation trace   */
#TRCLEV=(REQ,level,IQC)                    /* QCF request service trace */
#TRCLEV=(RSP,level,IQC)                    /* QCF response service trace */
#TRCLEV=(SAF,level,IQC)                    /* QCF SAF manager trace    */
#TRCLEV=(CMD,level,IQC)                    /* QCF command processor trace */
#TRCLEV=(SCI,level,IQC)                    /* QCF SCI trace            */
```

Example server configuration PROCLIB member

The IMS Queue Control Facility server requires a configuration member.

The following figure shows an example IMS Queue Control Facility server configuration PROCLIB member.

Note: For details of the configuration values for the server configuration PROCLIB member, see "Server configuration PROCLIB member reference" on page 55.

```
***********************************************************************
*   SAMPLE QCF SERVER CONFIGURATION PROCLIB MEMBER                    *
***********************************************************************
*---------------------------------------------------------------------*
* 1- to 5-character IMSPLEX(name).  This name represents the          *
* IMS Plex with which QCF is communicating and does not represent the *
* the IMS image.                                                      *
*---------------------------------------------------------------------*
IMSPLEX(IPLEX)      /* QCF Target Member Name = IPLEX                 */

*---------------------------------------------------------------------*
* Security can be turned on if the AUTH=Y parameter is added.         *
* The TSO user must be defined to the MVS of the server system.       *
* This is based on the MVS name, IMS name and function being          *
* performed.                                                          *
*---------------------------------------------------------------------*
AUTH(Y|N)           /* On: indicates that SAF call will be used to
                       Validate TSO requests by function.            */
```

7. Enabling and configuring the IMS Queue Control Facility extensions

Before you can use IMS Queue Control Facility to manage IMS message queues, you must enable the IMS Queue Control Facility extensions in all of the IMS control regions that access the IMS message queues.

For background information on the IMS Queue Control Facility extensions, refer to "IMS Queue Control Facility extensions overview" on page 49.

Topics:

- "Prerequisite checklist before configuring extensions" on page 32
- "Enabling the IMS Queue Control Facility extensions" on page 33
- "Example IMS Queue Control Facility extensions configuration" on page 34

- "Loading and unloading queues automatically" on page 37

Prerequisite checklist before configuring extensions

Ensure that the IMS SCI address space is active on the z/OS images on which IMS Queue Control Facility will be running.

Notes:

- IMS does not need to be SCI-enabled for IMS Queue Control Facility to work. If your current IMS environment does not use SCI, you can continue to use that environment without any changes. The IMS Queue Control Facility extensions use of SCI is independent of IMS.
- IMS only uses SCI for a SYSPLEX environment. To enable IMS for SCI , the DFSPB*xx* PROCLIB member must contain the keyword CSLG=.

The IMS Queue Control Facility extensions are enabled through the IMS Partner Product user exit interface (PPUE).

Ensure that you have completed the following steps before enabling and configuring IMS Queue Control Facility extensions:

1. Add the IMS Queue Control Facility load library to your JOBLIB or STEPLIB concatenation for the IMS control region JCL.

   The IMS Queue Control Facility load library must be concatenated in front of the IMS RESLIB, and the IMS Queue Control Facility load library must be APF authorized.

2. Specify a Partner Product user exit by completing either of the following steps:

   - Define IQCPPUE0 as a Partner Product user exit to the IMS DFSDF member, as follows:

     ```
     <SECTION=USER_EXITS>
       EXITDEF=(TYPE=PPUE,EXITS=(IQCPPUE0))
     ```

   - If your IMS system uses the IMS Tools Base Generic Partner exit routine, add the following statement to the GPR*xxxx*0 member in the IMS PROCLIB, where *xxxx* is the IMS ID.

     ```
     EXITDEF(TYPE(PARTNER) EXITNAME(IQCPPUE0)  LOADLIB(library_name))
     ```

     where *library_name* is the data set name of the IMS Queue Control Facility load library.

     For more information, see *Tools Base for z/OS IMS Tools Common Services User's Guide and Reference*.

3. Optional: To enable the QCF functions provided by IMS Tools Online System Interface (TOSI) by use of the TOIPARMS statement for UNLOAD, install IMS Tools Online System Interface and define FOIPPUE0 as an additional Partner Product user exit. If these functions are not required, you can skip this step.

   a. Install the IMS Tools Online System Interface component.

   b. Complete either of the following steps:

      - Define FOIPPUE0 as an additional Partner Product user exit to the IMS DFSDF member as follows:

        ```
        <SECTION=USER_EXITS>
          EXITDEF=(TYPE=PPUE,EXITS=(IQCPPUE0,FOIPPUE0))
        ```

        IQCPPUE0 and FOIPPUE0 can be specified in any order.

      - If your IMS system uses the IMS Tools Base Generic Partner exit routine, add the following statements to the GPR*xxxx*0 member in the IMS PROCLIB, where *xxxx* is the IMS ID.

        ```
        EXITDEF(TYPE(PARTNER) EXITNAME(IQCPPUE0)  LOADLIB(library_name))
        EXITDEF(TYPE(PARTNER) EXITNAME(FOIPPUE0)  LOADLIB(tosi_name))
        ```

        where *library_name* is the data set name of the IMS Queue Control Facility load library, and *tosi_name* is the data set name of the IMS Tools Online System Interface load library.

For more information, see *Tools Base for z/OS IMS Tools Common Services User's Guide and Reference*

Enabling the IMS Queue Control Facility extensions

You must set several parameters that apply to the IMS Queue Control Facility extensions and to the IMS Queue Control Facility Threshold/Queue Space Usage Notification components.

About this task

You specify these parameters in a PROCLIB member. This PROCLIB member is optional. All parameters have default values that apply if they are not coded, or if a PROCLIB member does not exist.

Note: The default values for queue space overflow protection are the parameters in IMS Queue Control Facility 3.2.

If you provide the IMS Queue Control Facility IMS extensions PROCLIB member, the member must be placed in a data set that is in the PROCLIB DD statement concatenation in your IMS control region startup JCL or PROC.

The IMS Queue Control Facility extensions PROCLIB member has one of the following two possible names:

- IQC*imsid*I

  The prefix, IQC, is followed by a 1- to 4–character IMS ID, which is followed by the letter *I*, and is padded with blanks on the right after the *I*, as needed.

- IQCIMS00

If you name the IMS Queue Control Facility IMS extensions PROCLIB member IQC*imsid*I, then the member applies only to the IMS system with the IMS ID of *imsid*.

For example, if you named a member IQCIMSAI, that member would be processed only by an IMS control region with the IMS ID of IMSA. If you named a member IQCISI, that member would be processed only by an IMS control region with an IMS ID of IS.

Restriction: If you already have your IMS ID set to IMS00, you must change that IMS ID to something else. Or, create an individual PROCLIB member for each IMS.

If you name the IMS Queue Control Facility IMS extensions PROCLIB member IQCIMS00, the member IQCIMS00 applies to all IMS control regions and the IMS Queue Control Facility extensions.

IMS Queue Control Facility extensions process PROCLIB member data in the following order:

1. An attempt is made to find a member named IQC*imsid*I.

   If member IQC*imsid*I is found, the parameters in this member are read and processed.

   A member that is associated with a specific IMSID overrides both the global member (IQCIMS00) and the default values.

2. If a member named IQC*imsid*I is not found, an attempt is made to find a member named IQCIMS00.

   If one is found, the parameters in that member are read and processed.

   Parameters that are specified in the global member override the default values.

   Note that IQCIMS00 is not read or processed at all if an IMSID-specific member was found.

3. Any parameter that is read but not specifically coded in the IMS Queue Control Facility extensions PROCLIB member takes on the default value.

Note: For details of the configuration values for the IMS Queue Control Facility extensions PROCLIB member, see "Extensions PROCLIB member reference" on page 56.

Example IMS Queue Control Facility extensions configuration

The *hlq*.SIQCSAMP (IQCIMSII) member contains a sample extensions configuration.

The following examples show the IMS Queue Control Facility extensions member. This example contains additional commented documentation.

Sample IMS Queue Control Facility PROCLIB member (part 1)

```
*************************************************************************
*   SAMPLE QCF IMS QUEUE CONTROL FACILITY EXTENSIONS PROCLIB MEMBER    *
*************************************************************************
*----------------------------------------------------------------------*
* 1- to 5-character IMSPLEX(name).  This name represents the           *
* IMS Plex with which IMS Queue Control Facility is communicating and  *
* does not represent the IMS image.                                    *
*----------------------------------------------------------------------*
IMSPLEX(IPLEX)      /* QCF Target Member Name = IPLEX */

*----------------------------------------------------------------------*
*  For QCF IMS extensions, start the SCI -  QCF extensions will        *
*  start the proc to initialize the Strucutred Call Interface.         *
*                                                                      *
*  1 to 8 character proclib member name.                               *
*----------------------------------------------------------------------*
SCIPROC=IQCSCI              /*START QCF SUPPLIED SCI PROC*/

*----------------------------------------------------------------------*
* On IMS COLDSTART - Perform the function requested in PROCNAME/       *
* JOBNAME /STARNAME                                                    *
*PROCNAME                                                              *
* 1- to 8-character proclib member name. IMS Queue Control Facility    *
* extensions will send this member name to the IMS Queue Control       *
* Facility server. If a server is not present, then IMS Queue          *
* Control Facility uses the jobname.                                   *
*JOBNAME                                                               *
* 1- to 8-character proclib member name.                               *
* IMS Queue Control Facility will submit the member name to the z/OS   *
* internal reader.                                                     *
*STARNAME                                                              *
* 1- to 8-character member name.                                       *
* IMS Queue Control Facility issues an internal IMS /STA REG command   *
* and the job is started via the IMS internal reader.                  *
* Specify "NOREQ" to request that no automatic reload be done.         *
*----------------------------------------------------------------------*
ONCOLDSTART(JOBNAME=NOREQ)  /* On cold start: automatic reload of
                               the messages will not occur */

*----------------------------------------------------------------------*
* On IMS WARMSTART - Perform the function requested in PROCNAME/       *
* JOBNAME /STARNAME                                                    *
*PROCNAME                                                              *
* 1- to 8-character proclib member name. IMS Queue Control Facility    *
* extensions will send this member name to the IMS Queue Control       *
* Facility server. If a server is not present, then IMS Queue          *
* Control Facility uses the jobname.                                   *
*JOBNAME                                                               *
* 1- to 8-character proclib member name.                               *
* IMS Queue Control Facility will submit the member name to the z/OS   *
* internal reader.                                                     *
*STARNAME                                                              *
* 1- to 8-character member name.                                       *
* IMS Queue Control Facility issues an internal IMS /STA REG command   *
* and the job is started via IMS internal reader.                      *
* Specify "NOREQ" to request that no automatic unload be done.         *
*----------------------------------------------------------------------*
ONWARMSTART(JOBNAME=NOREQ)  /* On warm start: automatic unload of
                               the dead letter queue will not occur*/
*----------------------------------------------------------------------*
```

Sample IMS Queue Control Facility PROCLIB member (part 2)

```
*----------------------------------------------------------------------*
* Queue Overflow Protection - IMS Queue Control Facility Unload of     *
* Queues. IMS Queue Control Facility will initiate the PROCNAME/       *
* JOBNAME when the AB threshold is crossed.                            *
```

```
*PROCNAME                                                   *
* 1- to 8-character proclib member name. IMS Queue Control Facility  *
* extensions will send this member name to the IMS Queue Control  *
* Facility server. If a server is not present, then IMS Queue    *
* Control Facility uses the jobname.                          *
*JOBNAME                                                     *
* 1- to 8-character batch job name. IMS Queue Control Facility   *
* extensions will update the overflow table with this name during  *
* initialization.                                            *
* IMS Queue Control Facility will submit the member name to the z/OS  *
* internal reader.                                           *
* If "NOUNLOAD" is specifed, IMS Queue Control Facility extensions  *
* will not submit the JCL to start the IMS Queue Control Facility     *
* default batch job.                                         *
*-------------------------------------------------------------------*
ONTHRESHOLDAB(JOBNAME=NOUNLOAD) /* On crossover of AB threshold:
                                   IMS Queue Control Facility will use
                                   the default batch job */

*-------------------------------------------------------------------*
* Queue Overflow Protection - IMS Queue Control Facility Unload of  *
* Queues. IMS Queue Control Facility will initiate the PROCNAME/    *
* JOBNAME when the BC threshold is crossed.                   *
*PROCNAME                                                     *
* 1- to 8-character proclib member name. IMS Queue Control Facility  *
* extensions will send this member name to the IMS Queue Control  *
* Facility server. If a server is not present, then IMS Queue    *
* Control Facility uses the jobname.                          *
*JOBNAME                                                     *
* 1- to 8-character batch job name. IMS Queue Control Facility   *
* extensions will update the overflow table with this name during  *
* initialization.                                            *
* IMS Queue Control Facility will submit the member name to the z/OS  *
* internal reader.                                           *
* If "NOUNLOAD" is specifed, IMS Queue Control Facility extensions  *
* will not submit the JCL to start the IMS Queue Control Facility     *
* default batch job.                                         *
*-------------------------------------------------------------------*
ONTHRESHOLDBC(JOBNAME=IQCOFBC1) /* On crossover of BC threshold: IMS
                                   Queue Control Facility will use
                                   the specified batch job */
*-------------------------------------------------------------------*
```

Sample IMS Queue Control Facility PROCLIB member (part 3)

```
*-------------------------------------------------------------------*
* Queue Overflow Protection - IMS Queue Control Facility Unload of  *
* Queues. IMS Queue Control Facility will initiate the PROCNAME/    *
* JOBNAME when the CD threshold is crossed.                   *
*PROCNAME                                                     *
* 1- to 8-character proclib member name. IMS Queue Control Facility  *
* extensions will send this member name to the IMS Queue Control  *
* Facility server. If a server is not present, then IMS Queue    *
* Control Facility uses the jobname.                          *
*JOBNAME                                                     *
* 1- to 8-character batch job name. IMS Queue Control Facility   *
* extensions will update the overflow table with this name during  *
* initialization.                                            *
* IMS Queue Control Facility will submit the member name to the z/OS  *
* internal reader.                                           *
* If "NOUNLOAD" is specifed, IMS Queue Control Facility extensions  *
* will not submit the JCL to start the IMS Queue Control Facility     *
* default batch job.                                         *
*-------------------------------------------------------------------*
ONTHRESHOLDCD(PROCNAME=IQCOLCD3) /* On crossover of CD threshold:
                                    IMS Queue Control Facility
                                    will send the specified PROCNAME
                                    to the server*/
*-------------------------------------------------------------------*
* Queue Overflow Protection - IMS Queue Control Facility Unload of  *
* Queues. IMS Queue Control Facility will initiate the PROCNAME/    *
* JOBNAME when the UT threshold is crossed.                   *
*PROCNAME                                                     *
* 1- to 8-character proclib member name. IMS Queue Control Facility  *
* extensions will send this member name to the IMS Queue Control  *
* Facility server. If a server is not present, then IMS Queue    *
* Control Facility uses the jobname.                          *
*JOBNAME                                                     *
* 1- to 8-character batch job name. IMS Queue Control Facility   *
```

```
* extensions will update the overflow table with this name during  *
* initialization.                                                  *
* IMS Queue Control Facility will submit the member name to the z/OS *
* internal reader.                                                 *
* If "NOUNLOAD" is specifed, IMS Queue Control Facility extensions  *
* will not submit the JCL to start the IMS Queue Control Facility    *
* default batch job.                                               *
*------------------------------------------------------------------*
ONTHRESHOLDUT(JOBNAME=NOUNLOAD)  /* On cross user set threshold:
                                    automatic off-load the messages
                                    on the queue */
*------------------------------------------------------------------*
*------------------------------------------------------------------*
* What action to take if an abend occurs under IMS Queue Control   *
* Facility in the IMS control region:  DUMP, NODUMP                *
*------------------------------------------------------------------*
ONABEND(DUMP)         /* On abend: take an SDUMP and then terminate
                         IMS Queue Control Facility processing in the
                         IMS address space  */

*------------------------------------------------------------------*
* What action to take if a non-abend error occurs during IMS Queue *
* Control Facility initialization in the IMS control               *
* region:   TERM or ABEND                                          *
*------------------------------------------------------------------*
ONINITERR(TERM)       /* On initialization error:
                         terminate IMS Queue Control Facility control
                         region processing and let IMS continue.  */
```

Sample IMS Queue Control Facility PROCLIB member (part 4)

```
*------------------------------------------------------------------*
*Queue Overflow Protection - QSUN diagnostics/prevention           *
* The queue space can be partitioned into 10 different Areas. Areas *
* target specific queue space usage (by IMS process).              *
* Areas are identified by an ID (ids must be unique).              *
* Each Area is defined as a percentage of the queue space (TOTAL). *
* Actions in an Area are defined as a percentage of the Area (USED).*
* The effective queue usage for an AREA ACTION is defined as:      *
*    Queue Usage = (TOTAL % * USED %)                              *
* Total percentage can be from 00-99 (00 disables the AREA).       *
* Used percentage can be from 00-99 (00 disables the AREA).        *
* ACTIONS subtypes are of four types:                              *
*    - CSTOP/CSTART Most probable cause of the excessive queue usage *
*    _ OSTOP/OSTOP  Alternative cause of the excessive queue usage  *
* The following ACTIONS can be specified:                          *
*    - NONE                                                        *
*    - WTO                                                         *
*    - WAIT                                                        *
*    - STOP                                                        *
*    - ABEND                                                       *
* The following ACTIONS can be taken against these IMS types:      *
*    - ALL                                                         *
*    - APPL (application)                                          *
*    - DC   (btam/vtam)                                            *
*    - OTMA (otma)                                                 *
*    - APPC (lu 6.2 / appc)                                        *
*    - MSC  (msc)                                                  *
*------------------------------------------------------------------*
AREA=(ID=AREA0001,                    Notify when a specific
      PERCENT=(TOTAL=20,USED=25),     process (ALL) uses 05% of queue
      TYPE=(ALL),
      CSTOPACTION=WTO,
      CSTARTACTION=WTO,
      OSTOPACTION=WTO,
      OSTARTACTION=WTO),
      NOTIFY=(TYPE=TSO,JNAME=member,USERID=tsoid,TEXT=(AREA0001 notification))
AREA=(ID=AREA0002,                    Wait a specific process
      PERCENT=(TOTAL=25,USED=40),     (OTMA/MSC) that uses
      TYPE=(OTMA,MSC),                10% of queue
      CSTOPACTION=WTO,
      CSTARTACTION=WAIT,
      OSTOPACTION=WTO,
      OSTARTACTION=WTO)
AREA=(ID=AREA0003,                    Stop a specific process
      PERCENT=(TOTAL=30,USED=50),     (DC/Dependent Region)
      TYPE=(APPL,DC),                 that uses 15% of queue
      CSTOPACTION=WTO,
```

```
        CSTARTACTION=STOP,
        OSTOPACTION=WTO,
        OSTARTACTION=WTO)
 AREA=(ID=AREA0004,                    Abend a specific process
        PERCENT=(TOTAL=40,USED=50),    (ALL)
        TYPE=(ALL),                    that uses 20% of queue
        CSTOPACTION=WTO,
        CSTARTACTION=ABEND,
        OSTOPACTION=WTO,
        OSTARTACTION=WTO)
```

Note: Areas can be used concurrently with failsafes.

Sample IMS Queue Control Facility PROCLIB member (part 5)

```
*--------------------------------------------------------------------------*
*Queue Overflow Protection - QSUN diagnostics/prevention           *
* The queue space can be partitioned into 10 different Failsafes.   *
* Failsafes target the entire queue space usage of all IMS processes. *
* Failsafes are identified by an ID, which must be unique.          *
* Each failsafe is defined as a percentage of the queue space (TOTAL). *
* Actions in a failsafe are defined as a percentage of the total    *
* queue space used by a process (USED).                            *
* A FAILSAFE partition action will only be taken when a process has *
* exceeded the USED percentage.                                    *
* The effective queue usage for a FAILSAFE ACTION is defined as:    *
*   Queue Usage = (TOTAL % + USED %)                               *
* Total percentage can be from 00-99 (00 disables the failsafe)     *
* Used  percentage can be from 00-99 (00 performs the action immediately *
* The following actions can be specified:                          *
*    - NONE                                                         *
*    - WTO                                                          *
*    - WAIT                                                         *
*    - STOP                                                         *
*    - ABEND                                                        *
* Actions are against all IMS types                                *
*--------------------------------------------------------------------------*
FAILSAFE=(ID=FAIL0001,                     Total queue usage is 55%;
          PERCENT=(TOTAL=55,USED=05),      do notify of any process
          ACTION=WTO)                      that uses more than 5% of queue
          NOTIFY=(TYPE=TSO,JNAME=member,USERID=tsoid,TEXT=(FAIL0001 notification))
FAILSAFE=(ID=FAIL0002,                     Total queue usage is 60%;
          PERCENT=(TOTAL=60,USED=04),      do notify of any process
          ACTION=WTO)                      that uses more than 4% of queue
FAILSAFE=(ID=FAIL0003,                     Total queue usage is 65%;
          PERCENT=(TOTAL=65,USED=03),      do notify of any process
          ACTION=WTO)                      that uses more than 3% of queue
FAILSAFE=(ID=FAIL0004,                     Total queue usage is 70%;
          PERCENT=(TOTAL=70,USED=02),      do notify of any process
          ACTION=WTO)                      that uses more than 2% of queue
FAILSAFE=(ID=FAIL0005,                     Total queue usage is 75%;
          PERCENT=(TOTAL=75,USED=01),      do wait any process
          ACTION=WAIT)                     that uses more than 1% of queue
FAILSAFE=(ID=FAIL0006,                     Total queue usage is 80%, do
          PERCENT=(TOTAL=80,USED=00),      stop everything.
          ACTION=STOP)
```

Note: Failsafes can be used concurrently with areas.

In the configuration definition, you can also define thresholds of queue usage for which unloads can be initiated.

Loading and unloading queues automatically

IMS Queue Control Facility IMS extensions allow you to specify a PROCLIB member that contains the IMS Queue Control Facility control region options.

About this task

You can use any of the following three methods to configure IMS Queue Control Facility to perform automatic queue loads and unloads:

* ONTHRESHOLD*xx*

- ONCOLDSTART
- ONWARMSTART

Ensure that your IMS Queue Control Facility IMS extensions PROCLIB member is in your PROCLIB.

Your IMS Queue Control Facility IMS extensions PROCLIB member contains the member name that must be placed in a data set that is in the PROCLIB DD statement concatenation in your server region startup JCL or PROC.

Determine the name of the PROCLIB that contains the IMS Queue Control Facility control region options.

Defining threshold jobs

Use the SBMP*xx* parameter on the QCFIN function. This parameter identifies the threshold for which the job is running.

You must set the *xx* value of this parameter to either AB, BC, CD, or UT to define a threshold job that is to be scheduled.

If you specify this parameter for each job, only the job that is processing the highest threshold is allowed to run in order to eliminate message queue thrashing; the lower threshold jobs are terminated with AIB reason code E000.

If multiple jobs are scheduled at the same time and this parameter is not specified for each job, it is unpredictable which job will run, and unexpected results can occur.

Define this parameter for the QCFIN parameters that are being supplied to threshold jobs AB, BC, CD, and UT.

- For the threshold AB unload job, specify the function statement SBMP*xx* as shown in this example:

```
FUNCTION UNLOAD QUEUETYPE=(ALL),SBMPAB
```

- For the threshold BC unload job, specify the function statement SBMP*xx* as shown in this example:

```
FUNCTION UNLOAD QUEUETYPE=(ALL),SBMPBC
```

- For the threshold CD unload job, specify the function statement SBMP*xx* as shown in this example:

```
FUNCTION UNLOAD QUEUETYPE=(ALL),SBMPCD
```

- For the threshold UT unload job, specify the function statement SBMP*xx* as shown in this example:

```
FUNCTION UNLOAD QUEUETYPE=(ALL),SBMPUT
```

The following example of an ONTHRESHOLDAB(*procname*) PROCLIB member forces an automatic browse of committed Queue Manager records.

```
***************************************************************
*  QCF IMS CONTROL REGION EXTENSIONS PROCLIB MEMBER AUTOAB   *
*  INPUT TO QCF SERVER                                       *
***************************************************************
*-------------------------------------------------------------*
* 1-to 4-character IMS ID name.                               *
*-------------------------------------------------------------*
IMS(IMS1)                      /* IMS name = IMS1 */
*-------------------------------------------------------------*
* Define the data set and member that contains the QCF input *
* command stream. 1-to 44-character for PDS name             *
* 1-to 8-character for member name                           *
*-------------------------------------------------------------*
QCFIN(DSN(USER.PRIVATE.PROCLIB),MEM(AQCFABIN)
*-------------------------------------------------------------*
* Define 1-to 44-character data set name to be used for the  *
* QCF print output                                           *
*-------------------------------------------------------------*
QCFPRINT(USRT002.QCF31AB.QCFPRINT)
*-------------------------------------------------------------*
* Define 1-to 44-character data set name to be used for the  *
* QCF unload output                                          *
```

```
*------------------------------------------------------------*
BROWSE(USRT002.QCF31AB.UNLOAD)
```

The following example of an ONTHRESHOLDBC(*procname*) PROCLIB member forces an automatic unload of committed Queue Manager records.

```
**************************************************************************
*  QCF IMS CONTROL REGION EXTENSIONS PROCLIB MEMBER AUTOBC        *
*  INPUT TO QCF SERVER                                           *
**************************************************************************
*----------------------------------------------------------------------*
* 1- to 4-character IMS ID name.                                *
*----------------------------------------------------------------------*
IMS(IMS1)                      /* IMS name = IMS1 */
*----------------------------------------------------------------------*
* Define the data set and member that contains the QCF input    *
* command stream. 1-to 44-character for PDS name                *
* 1 to 8 character for member name                              *
*----------------------------------------------------------------------*
QCFIN(DSN(USER.PRIVATE.PROCLIB),MEM(AQCFBCIN)
*----------------------------------------------------------------------*
* Define 1-to 44-character data set name to be used for the     *
* QCF print output                                              *
*----------------------------------------------------------------------*
QCFPRINT(USRT002.QCF31BC.QCFPRINT)
*----------------------------------------------------------------------*
* Define 1-to 44-character data set name to be used for the     *
* QCF unload output                                             *
*----------------------------------------------------------------------*
UNLOAD(USRT002.QCF31BC.UNLOAD)
```

Note: For details of the configuration values for the IMS Queue Control Facility extensions ONTHRESHOLD*nn* PROCLIB member, see "Extensions ONTHRESHOLDnn PROCLIB member reference" on page 73.

8. Implementing security for TSO users and batch jobs

The SAF server component and the destination control table provide security for the TSO client and batch job users.

The destination control table provides destination-specific security control for IMS Queue Control Facility. The SAF server component control provides security for the total environment and systems, including IMS Queue Control Facility.

Implementing SAF security for TSO users and batch jobs

The SAF server component provides security for the TSO client and batch job users. A single SAF call replaces the previous system of security that issued a call each time a function was processed by the server.

About this task

SAF security is based on the MVS SYSNAME, the IMS name, and the function that is performed.

The functions QUERY, BROWSE, LOAD, UNLOAD, and RECOVER are protected when used from the client application or in a batch job. The server commands are not QCF SAF protected. The RECOVER function is used only in a shared queue environment to recover messages from the COLD queue after the control region breakdown and restart. Procedures RECOVERDM, RECOVERAB, and REPROCESS are QCF SAF protected only for the LOAD step. LOAD is the final step in the procedures that operate on the live queues.

Conditions for implementing SAF security:

- Ensure that the TSO user is defined to the MVS of the server system.
- You can turn off security for QCF TSO application clients by specifying the AUTH=N parameter in the server IQCCFG member.
- If RACF® restrictions are set, QCF batch jobs are protected automatically. The AUTH parameter in the Server IQCCFG member has no impact on the QCF batch jobs.

- Security is linked to the USERID on the JOB statement of the job step startup JCL through the RACF started group, the ICHRIN00 PARMLIB control block, or the equivalent security system function.
- You must set up RACF or an equivalent security product for USERID access to the IMS message queues.

When a user selects an IMS system, authorization of that system is checked by a SAF call, as shown in the following table:

Table 1. SAF system authorization

CLASS	FACILITY
Resource name:	`IQC.MVS.IMS`

In the table, *MVS* is the MVS system ID, and *IMS* is the IMS ID.

Grant UPDATE access for full access to all menu options, or READ access to restrict you to queues that contain only queries or filters and to the list of destinations.

When a user selects a particular function, authorization of that function is checked with a SAF call, as shown in the following table:

Table 2. SAF function authorization

CLASS	FACILITY
Resource name:	`IQC.MVS.IMS.FUNCTION`

In the table, *FUNCTION* is any of the following values:

- BROWSE
- RECOVER
- QUERY
- LOAD
- UNLOAD

Creating RACF authorizations

You must define RACF security authorizations for IMS Queue Control Facility to work with IMS and MVS.

Procedure

To set up RACF authorizations, complete the following steps:

1. Define the facility for IMS Queue Control Facility for MVS and for IMS:

```
RDEFINE FACILITY IQC.MVSID.IMSID (for every IMS and MVS system)
or
RDEFINE FACILITY IQC.*.*          (for all MVS and IMS subsystems
```

2. Define the facility for the necessary functions for MVS and IMS subsystems:

```
RDEFINE FACILITY  IQC.MVSID.IMSID.LOAD
RDEFINE FACILITY  IQC.MVSID.IMSID.UNLOAD
RDEFINE FACILITY  IQC.MVSID.IMSID.QUERY
RDEFINE FACILITY  IQC.MVSID.IMSID.BROWSE
RDEFINE FACILITY  IQC.MVSID.IMSID.RECOVER
```

   You can specify MVSID and IMSID as an asterisk (*) to be valid for all MVS and IMS subsystems.

3. Optional: To allow user USRT001 to access IMS and MVS, specify the following rule:

```
PERMIT IQC.MVSID.IMSID CLASS(FACILITY) ID(USRT001) ACCESS(UPDATE) GENERIC
```

4. Optional: To allow user USRT001 to run the load function, specify the following rule:

```
PERMIT IQC.MVSID.IMSID.LOAD CLASS(FACILITY) ID(USRT001) ACCESS(UPDATE) GENERIC
```

RACF authorization examples

The following example of RACF security implementation uses an asterisk (*) instead of MVS IDs or IMS
IDs:

```
RDEFINE FACILITY IQC.*.*           UACC(NONE) AUDIT(NONE)
RDEFINE FACILITY IQC.*.*.BROWSE    UACC(NONE) AUDIT(NONE)
RDEFINE FACILITY IQC.*.*.QUERY     UACC(NONE) AUDIT(NONE)
RDEFINE FACILITY IQC.*.*.LOAD      UACC(NONE) AUDIT(NONE)
RDEFINE FACILITY IQC.*.*.UNLOAD    UACC(NONE) AUDIT(NONE)
RDEFINE FACILITY IQC.*.*.RECOVER   UACC(NONE) AUDIT(NONE)

PERMIT IQC.*.*        CLASS(FACILITY) ID(USRT001) ACCESS(UPDATE) GENERIC
PERMIT IQC.*.*.QUERY  CLASS(FACILITY) ID(USRT001) ACCESS(UPDATE) GENERIC
PERMIT IQC.*.*.BROWSE CLASS(FACILITY) ID(USRT001) ACCESS(UPDATE) GENERIC
PERMIT IQC.*.*.LOAD   CLASS(FACILITY) ID(USRT001) ACCESS(UPDATE) GENERIC
PERMIT IQC.*.*.UNLOAD CLASS(FACILITY) ID(USRT001) ACCESS(UPDATE) GENERIC

PERMIT IQC.*.*        CLASS(FACILITY) ID(USRT002) ACCESS(UPDATE) GENERIC
PERMIT IQC.*.*.QUERY  CLASS(FACILITY) ID(USRT002) ACCESS(UPDATE) GENERIC
PERMIT IQC.*.*.BROWSE CLASS(FACILITY) ID(USRT002) ACCESS(UPDATE) GENERIC
PERMIT IQC.*.*.LOAD   CLASS(FACILITY) ID(USRT002) ACCESS(UPDATE) GENERIC

PERMIT IQC.*.*        CLASS(FACILITY) ID(USRT003) ACCESS(UPDATE) GENERIC
PERMIT IQC.*.*.QUERY  CLASS(FACILITY) ID(USRT003) ACCESS(UPDATE) GENERIC
PERMIT IQC.*.*.BROWSE CLASS(FACILITY) ID(USRT003) ACCESS(READ)   GENERI
PERMIT IQC.*.*.UNLOAD CLASS(FACILITY) ID(USRT003) ACCESS(UPDATE) GENERIC

PERMIT IQC.*.*        CLASS(FACILITY) ID(USRT004) ACCESS(UPDATE) GENERIC
PERMIT IQC.*.*.QUERY  CLASS(FACILITY) ID(USRT004) ACCESS(UPDATE) GENERIC
PERMIT IQC.*.*.BROWSE CLASS(FACILITY) ID(USRT004) ACCESS(READ)   GENERIC

PERMIT IQC.*.*        CLASS(FACILITY) ID(USRT005) ACCESS(UPDATE) GENERIC
PERMIT IQC.*.*.QUERY  CLASS(FACILITY) ID(USRT005) ACCESS(UPDATE) GENERIC
```

The settings in the previous example cause the following authorizations:

- USRT001 can run all menu options.
- USRT002 cannot delete messages.
- USRT003 cannot load messages.
- USRT004 has QUERY and BROWSE access only.
- USRT005 has QUERY access only.

You can define resources IQC.MVSID.IMISID and IQC.MVSID.IMSID.*FUNCTION* (all possible functions
that run under IMS Queue Control Facility control processing). When users have READ or UPDATE access
to these resources, they can run the function.

After IMS Queue Control Facility starts TSO panels and selects a server, a security check is performed to
determine whether the user is authorized to access IQC.MVSID.IMSID.

If the user selects a function, IMS Queue Control Facility performs a security check to determine whether
the user is authorized to use that function (IQC.MVSID.IMSID.*FUNCTION*).

If data sets are not authorized, they cannot be allocated and used. Follow these authorization rules when
you create access authorizations:

- All data sets that are allocated when IMS Queue Control Facility starts must be authorized for the user
  (set *usrhlq* in the IQC starting script).
- All data sets whose names are specified by the user, such as SCRAPLOG or COPY, must be authorized
  for that user.
- All data sets that are passed to the server must be authorized for the server user ID.

The following example shows how SAF security might be implemented for TSO users:

```
RDEFINE FACILITY IQC.*.*.S
ELECT UACC(NONE) AUDIT(NONE)
RDEFINE FACILITY IQC.MVSB.IMS7.QUERY UACC(NONE) AUDIT(NONE)
RDEFINE FACILITY IQC.MVSA.*.* UACC(NONE) AUDIT(NONE)
RDEFINE FACILITY IQC.*.IMRG.* UACC(NONE) AUDIT(NONE)
PERMIT IQC.*.*.LOAD CLASS(FACILITY) ID(IQCREAD) ACCESS(READ) GENERIC
TSO CONNECT TSOUSER GROUP(IQCREAD)
```

Implementing destination control security for TSO client and batch job users

The destination control table provides security for the TSO client and batch job users.

About this task

By default, users have access to all messages in all control regions for the function level QUERY. Destination control changes the default so that users have access to the message number but not the message content.

The destination control table defines the authorization for TSO and batch jobs users to run IMS Queue Control Facility functions on specific destination types and names within a specific IMS and IMSplex. The internal destination control table is created by using the ACCESS control statements.

The destination control table is valid within the IMSplex and is linked with the name IQC*plexname* in *hlq*.SIQCLINK.

- If the IQC*plexname* load module is not found in *hlq*.SIQCLINK, destination control is not performed and only the RACF security, if configured, is in effect.
- If the IQC*plexname* load module is found in *hlq*.SIQCLINK, destination control security is performed on each message to verify that the USERID on the JOB or TSO client user ID statement is properly authorized to run the function for the message in the IMSplex and in the IMS.

ACCESS control statements are used to define the destination control table. ACCESS control statements are created for IMS and USERID in an IMSplex. If the destination table exists, but no valid entry is found for IMS and USERID, the user ID is not authorized to perform any functions within an IMS.

Procedure

To implement destination control security for TSO client and batch job users:

1. Modify the ACCESS control statements in the destination control sample member IQCDSTCI.

   IMSID=*imsid*
   The IMS ID.

   USERID=*userid*
   The TSO user ID.

   INCLUDE(*subparameters*)
   Defines the functions and message types and names that are allowed for the user.

   FUNCTION=(B,U,L,R,A)
   The function type: B for BROWSE, U for UNLOAD, L for LOAD, R for RECOVER, and A for ALL.

   DESTYPE=(LT,TR,APPC,OTMA,ALL)
   The destination type: LT for LTERM, TR for transaction, APPC for APPC/MVS Advanced Program-to-Program Communication, OTMA for Open Transaction Manager Access, or ALL for all destination types.

   DESNAME=(*destination*)
   The name of the destination.

   EXCLUDE(*subparameters*)
   Defines the functions and messages types and names that are not allowed for the user.

FUNCTION=(B,U,L,R,A)
  The function type: B for BROWSE, U for UNLOAD, L for LOAD, R for RECOVER, and A for ALL.

DESTYPE=(LT,TR,APPC,OTMA,ALL)
  The destination type: LT for LTERM, TR for transaction, APPC for APPC/MVS Advanced Program-to-Program Communication, OTMA for Open Transaction Manager Access, or ALL for all destination types.

DESNAME=(*destination*)
  The name of the destination.

2. Modify the sample member IQCSYNTY.
3. Submit the modified IQCSYNTY sample JCL.

Example

In the following example, the destination control table is created by using input files:

```
ACCESS(IMSID=*,USERID=*)

ACCESS(IMSID=SYS3,USERID=USRT002,
     EXCLUDE(FUNCTION(B,L,U),DESTYPE(ALL),DESNAME(E*))
     EXCLUDE(FUNCTION(B,L,U),DESTYPE(ALL),DESNAME(C*)))

ACCESS(IMSID=****,USERID=USRT003,
     INCLUDE(FUNCTION(B,L),DESTYPE(LT,TR,APPC,OTMA),DESNAME(E*))
     INCLUDE(FUNCTION(A),DESTYPE(LT,OTMA),DESNAME(A*))

        EXCLUDE(FUNCTION(B,L),DESTYPE(LT,TR,APPC),DESNAME(ER1*))
        EXCLUDE(FUNCTION(B,L,U),DESTYPE(APPC,OTMA),DESNAME(ER2*)))
```

In this example:

- USRT002 is denied access in the IMS=SYS3 for functions BROWSE, LOAD, and UNLOAD, for all destination types, and for destination names that begin with E and C.
- USRT003 is granted access (included) in all IMS systems for functions BROWSE and LOAD, for destination types LT, TR, APPC, and OTMA, and for destination names that begin with E.
- USRT003 is granted access (included) in all IMS systems for ALL functions, for destination types LT and OTMA, and for destination names that begin with A.
- USRT003 is denied access to destinations that begin with ER1 for functions BROWSE and LOAD, for destination types LT, TR, and for APPC.
- USRT003 is denied access to destinations that begin with ER2 for functions BROWSE, LOAD, and UNLOAD, and for destination types APPC and OTMA.

9. Starting the TSO client (ISPF interface)

IMS Queue Control Facility requests originate from the TSO client (ISPF interface) and flow to the IMS Queue Control Facility server to be distributed to the appropriate IMS Queue Control Facility extensions for further processing.

About this task

The TSO client has the ability to join a specific IMS Queue Control Facility group and to locate all of the IMS Queue Control Facility servers with which it can communicate.

If no IMS Queue Control Facility server is selected when you begin to use the interface, the available IMS Queue Control Facility servers are presented as a list of potential servers.

The TSO client communicates with the IMS Queue Control Facility server by using IMS SCI services.

Procedure

To start the IMS Queue Control Facility client interface, complete the following steps:

1. Log on to TSO and specify a minimum REGION of (6144).
2. Select ISPF option **6**, and issue the following command on the TSO command line:

```
Ex 'hlq.SIQCEXEC(IQC31)'
```

Requirement: Both the IMS Queue Control Facility server address spaces and the IMS Queue Control Facility TSO/ISPF clients must use the same SCI IMSPLEX name.

Requirement: The SCI IMSPLEX name must be the same as the name that is specified in the SCI initialization PROCLIB member. See the IMSPLEX(NAME=*name*) configuration statement.

Starting the ISPF user interface

You can use the ISPF user interface to select functions and selection criteria for completing numerous tasks.

For example, you can use the ISPF interface to query messages, unload and load messages onto the queues, and release or end tasks on the wait queue.

To start the ISPF user interface, perform the following steps:

1. Set up a command in your user CLIST library that allows you to access the panels.

   A sample command is supplied in SMP library SIQCEXEC in member IQC31.

   If you prefer, you can rename this member in your user CLIST library.

   - If IQC31 is called with TSO IQC31 and not with TSO *hlq*.SIQCEXEC(IQC31), MLIB, PLIB, LINK, and CEXE must point to the correct data sets.
   - MLIB, PLIB, LINK, and CEXE must point to the correct data sets if the HLQ for TSO *hlq*.SIQCEXEC(IQC31) is different for any of the passed data sets.
2. Edit and modify member IQC31 according to the instructions inside the member, then save your changes.
3. If you are running with security turned on, apply security profiles as shown in the following examples.

   If you are running with security turned off, this step is optional.

Example security settings

In the following example, the installation verification procedure user has READ access to the function level of QUERY.

Table 3. Read access

CLASS	FACILITY
PROFILE	IQC.*MVS*.*IMS*.QUERY

In the following example, the installation verification procedure user has UPDATE or READ access to IMS user level security.

Table 4. Update or read access

CLASS	FACILITY
PROFILE	IQC.*MVS*.*IMS*.

In the table, *MVS* is the MVS system ID, and *IMS* is the IMSID.

Ensure that the target IMS control region has started.

Example member IQC31

```
/* REXX ************************ rexx ******************************/
/*                                                                */
/*                 M O D U L E   P R O L O G U E                  */
/*                                                                */
/******************************************************************/
/*                                                                */
/*        Name:  IQC                                              */
/*                                                                */
/* Description:  Start up rexx for    ISPF Dialog subsystem.      */
/*                                                                */
/*************************************************************@SCPYRT*/
/*        Licensed Materials - Property of IBM                   */
/*                                                                */
/*        IMS QCF for z/OS Version 3 Release 2                    */
/*                                                                */
/*        5697-N50                                               */
/*                                                                */
/*        Copyright IBM Corp. 2000, 2014                          */
/*                                                                */
/*        All Rights Reserved.                                   */
/*        US Government Users Restricted Rights -                */
/*        Use, duplication or disclosure restricted             */
/*        by GSA ADP Schedule Contract with IBM Corp.           */
/*                                                                */
/*************************************************************@ECPYRT*/
/*                                                                */
/*        Status:  IMS Queue Control Facility                    */
/*                                                                */
/* Module type:  rexx                                            */
/*                                                                */
/*     Changes:                                                  */
/* APAR...   ID  PREREQ DATE....   DESCRIPTION......................*/
/* PK90862           09/16/2009 Improve comments on parameters    */
/*                                                                */
/*************************************************************@ECPYRT*/
/*                                                                */
/* Warning! When copying REXX programs to your CLIST.CLIST ensure */
/*          that sequence numbers are not inserted in columns 73-80 */
/*          of your new CLIST.CLIST member.                       */
/*                                                                */
/* This REXX is executed by TSO users to invoke the QCF ISPF feature. */
/*                                                                */
/* Make all customization and use :                              */
/* ex 'smphlq.SIQCEXEC(IQC31)'   to start the script             */
/*                                                                */
/* Change smphlq to your local hlq and the SIQC prefixed qualifiers */
/* to your local values if you do not use IBM SMP/E naming standards. */
/*                                                                */
/* The "USRHLQ" parameter is used to set the hlq for the temporary */
/* data sets. In this sample the parameter is set to "IQC". You can */
/* change this to some other value, or to userid(). Make sure the QCF */
/* server has authority to read and write these data sets.       */
/*                                                                */
/* The "OVFLHLQ" parameter is used to set the hlq for the permanent */
/* data sets, containing overflow protection tables ..COMMIT     */
/* ..UNCOMMIT.In this sample the parameter is set to "IQC". You can */
/* change this to some other value, as required by your policies. */
/* OVFLHLQ should not exceed 25 chars, QCF builds the data set names */
/* by adding '.OSNAME.IMSID.COMMIT' '.OSNAME.IMSID.UNCOMMIT'      */
/* and expects to find these data sets catalogged and set up with info*/
/* This is how the data sets should be named in your installation. */
/* OSNAME is the name of the operating system installation. The name */
/* shows under &SYSNAME in the output of D SYMBOLS command.        */
/* If the &SYSNAME starts with a number - it is prefixed by 'S' to */
/* be consistent with data set naming rules.                     */
/*                                                                */
/* The "IMSPLEX" parameter is be used to set the name of IMSPLEX  */
/* Make sure the control region, the QCF server and the SCI are using */
/* the same plex name.                                           */
/*                                                                */
/* The "SPACE" parameter can be used to cange the default primary and */
/* secondary space allocation for BROWSE/BROWSES/UNLOAD work data sets*/
/* This is the only optional paremeter. If not set in the sample - */
/* the default space allocation is used.                         */
/*                                                                */
/* The "SCILOAD" parameter is used to set the name of your SCI    */
/* load library ( IMS reslib)                                    */
/*                                                                */
/* The "N" parameter is set to "Y" if a diagnostic trace is requested */
```

```
/* by IBM service personnel.                                       */
/*                                                                  */
/* The "DUMVOL" parameter must be changed to a valid VOLSER for     */
/* temporary work data sets in a non-SMS managed shop.              */
/*                                                                  */
/* You will also have to copy 'smphlq.SIQCEXEC(IQCC00)' into your   */
/* CLIST.CLIST data set.                                            */
/*                                                                  */
/*                                                                  */
/********************************************************************/
/*                                         */
/*   GET the HLQ of the source dataset */
/*                                         */
trace off
PARSE SOURCE . . . . HLQ .
HLQ = LEFT(HLQ,length(HLQ)-(INDEX(REVERSE(HLQ),'.')))

MLIB = HLQ||".SIQCMENU" /* <== change this for your site if required */
PLIB = HLQ||".SIQCPENU" /* <== change this for your site if required */
LINK = HLQ||".SIQCLINK" /* <== change this for your site if required */
CEXE = HLQ||".SIQCEXEC" /* <== change this for your site if required */
TLIB = HLQ||".SIQCTENU" /* <== change this for your site if required */
USRHLQ = "IQC"          /* <== change this for your site if required */
OVFLHLQ = "IQC"         /* <== change this for your site if required */
                        /*     up to 25 characters                   */
IMSPLEX = "PLEX1"       /* <== change this for your IMSPLEX name    */
SPACE   = "(5 5)"       /* <== change this for BROWSE/BROWSES/UNLOAD */
                        /*     primary and secondary space allocation*/
SCILOAD = "imshlq.RESLIB" /* <== SCI STEPLIB name                   */
If LENGTH(OVFLHLQ) > 25
  THEN DO
  SAY ' Invalid OVFLHLQ length ( >25) '
  EXIT
  END

IF LENGTH(USRHLQ) > 8
  THEN DO
  SAY ' Invalid USRHLQ  length ( >8 ) '
  EXIT
  END
/********************************************************************/
/*   You can use the following script to decrease the parameter     */
/*   string in case your qualifiers are long and the parameter      */
/*   becomes longer than 250 bytes                                  */
/*   Uncomment next lines and delete Address TSO command            */
/* PRG  = CEXE||'(IQCC00)'                                          */
/* VR1  = MLIB                                                      */
/* VR2  = PLIB                                                      */
/* VR3  = LINK                                                      */
/* VR4  = CEXE                                                      */
/* VR5  = TLIB                                                      */
/* VR6  = 'Y'                                                       */
/* VR7  = 'DUMVOL'                                                  */
/* VR8  = USRHLQ                                                    */
/* VR9  = OVFLHLQ                                                   */
/* VR10 = IMSPLEX                                                   */
/* VR11 = SCILOAD                                                   */
/* VR12 = SPACE                                                     */
/* "EXEC '"PRG"' '"VR1 VR2 VR3 VR4 VR5 VR6 VR7 VR8 VR9 VR10 VR11    */
/*                VR12 "'"                                          */
/********************************************************************/
Address TSO
CALL IQCC00,
MLIB,
PLIB,
LINK,
CEXE,
TLIB,
N,
DUMVOL,
USRHLQ,
OVFLHLQ,
IMSPLEX,
SCILOAD,
SPACE
```

10. Starting the overflow protection test tool

IMS Queue Control Facility provides an overflow protection test tool that consists of a batch message processing (BMP) program and a message processing program (MPP) that are used to generate messages on the IMS Queue Manager queues.

The overflow protection test tool allows you to populate the Queue Manager queues to a level of utilization that forces the crossing of the user-defined thresholds and trigger the corresponding defined actions for that threshold.

Topics:

-
-

Using the BMP test tool

Before you begin

To prepare the BMP test tool, perform the following procedures:

1. Compile the sample IQCSPSB0 PSB (refer to member IQCSPSB0 in *hlq*.SIQCSAMP) and add it to the IMS PSBLIB.

   PSB example:

   ```
   PCB    TYPE=TP,MODIFY=YES
   PSBGEN LANG=ASSEM,PSBNAME=IQCSPSB0
   END
   ```

2. Perform an ACBGEN build for the PSB and add it to the IMS ACBLIB.

3. Customize the BMP sample JCL (refer to member IQCSBMP in *.hlq*.SIQCSAMP).

Procedure

You can access the BMP test tool from the main IMS Queue Control Facility TSO panel using option **5** (for type 2 overflow protection) or option **6** (for type 3 overflow protection):

1. From panel IQCP00, select option **5** (type 2) or option **6** (type 3) and press Enter.

   The Queue Overflow Parameters Sub-menu panel is displayed (IQCP70).

2. Select option **6 - Space utilization notification testing tool**.

3. In panel IQCPIVP1, provide the data set name, member name for the BMP, and specify Y for **Execute the job**, and press Enter.

4. In the ISPF Edit Panel, edit the BMP JCL if necessary.

   When finished, press End.

5. In the ISPF Browse Panel, review modifications (if any) made to the BMP JCL.

   When finished, press End.

6. In panel IQCPIVP2, provide the following message details, and press Enter.

   - Destination's Name(s)

   - Number of Destination(s)

   - Type of messages (short/long) messages to be inserted to each destination

   - Type of messages (committed/uncommitted) messages to be inserted to each destination

   - Number of Messages or Percent of Queue

7. In the ISPF Edit Panel, edit the BMP JCL if necessary.

   When finished, press End.

8. In panel IQCPIVP3, press Enter to start the BMP test.

Using the MPP test tool

Before you begin

To prepare the MPP test tool, perform the following procedures:

1. Compile the sample IQCSPSB1 PSB (refer to member IQCSPSB1 in *hlq*.SIQCSAMP) and add it to the IMS PSBLIB.

   PSB example:

   ```
   PCB TYPE=TP,MODIFY=YES
   PSBGEN LANG=ASSEM,PSBNAME=IQCSPSB1
   END
   ```

2. Perform an ACBGEN build for the PSB and add it to the IMS ACBLIB.
3. Define the transaction IQCT1 to IMS.
4. Link-edit the IMS Queue Control Facility application module IQCSMPP1 with a new name IQCSPSB1.
5. Build the MFS format for this application (use member IQCSMFS in the IMS Queue Control Facility sample library).

Procedure

The MPP test tool is an IMS application program that is associated with the transaction IQCT1:

1. Use MFS to access the application program:

   ```
   /FORMAT IQCT1
   ```

2. Use the MFS screen to define the requested destinations.

Chapter 4. Supplemental setup information

The topics in this reference section provide supplemental information for the concepts used in the IMS Queue Control Facility setup procedures.

Topics:

- "IMS Queue Control Facility extensions overview" on page 49
- "Considerations for creating the SCI address space" on page 49
- "Configuring threshold processing" on page 51
- "Overriding buffer definitions for HIOP" on page 52

IMS Queue Control Facility extensions overview

IMS Queue Control Facility extensions run in the IMS control region. Extensions are responsible for manipulating IMS message queues and transmitting the output data to the IMS Queue Control Facility server by way of the IMS structured call interface (IMS SCI).

The IMS Queue Control Facility extensions provide the following capabilities:

- Allow you to manipulate the message queues
- Provide queue overflow protection
- Provide automatic load and unload

Extensions are loaded into the IMS control region during IMS initialization.

Extensions register with the IMS SCI, and then await notifications from the IMS Queue Control Facility server address spaces about IMS message queues to be manipulated.

The IMS SCI component in each LPAR is used to communicate among different IMS address spaces and IMS Queue Control Facility address spaces.

This background topic supplements the following setup procedure: "7. Enabling and configuring the IMS Queue Control Facility extensions" on page 31.

Considerations for creating the SCI address space

There are several key concepts that you must consider when creating the SCI address space that is required by the IMS Queue Control Facility extensions to communicate to the IMS Queue Control Facility server.

- Within any given logical partition (LPAR), any IMS Queue Control Facility extensions member requires the IMS Structured Call Interface (SCI) facility for communication between the IMS Queue Control Facility extension member and the IMS Queue Control Facility server.
- The IMS Queue Control Facility extensions member does not require that your IMS system have the SCI address space enabled for the IMS system; it only requires that the SCI be enabled for the extensions member.
- If you already have the IMS system enabled for the SCI to support the IMS Operation Manager or IMS Resource Manager facilities, you do not need to enable a second SCI for the IMS Queue Control Facility extensions.

  The IMS Queue Control Facility extensions member can share the same SCI that is already enabled by the IMS system.

  This configuration is accomplished by specifying the IMS Queue Control Facility IMSPLEX(*name*) parameter, where *name* matches the name specified in the SCI IMSPLEX(NAME=*name*) parameter.

- Even when the current IMS system in the LPAR does not have the SCI enabled for the IMS system, there might be a time when the SCI must be enabled by the IMS system to support other IMS functions (for example, IMS Operations Manager or IMS Resource Manager).

  If these IMS functions are added at some later date, you can use the SCI that has been enabled by the IMS Queue Control Facility extensions that resides in the same LPAR.
- If only the IMS Queue Control Facility extensions is enabling SCI and not the IMS system, you must consider the name assignment of the IMS Queue Control Facility IMSPLEX(*name*) parameter, which is also one of the parameters defined by the SCI IMSPLEX(NAME=*name*) parameter and IMS system IMSPLEX=*name* parameter.

  For the IMS system, this parameter is defined in the Common Services Layer section of the DFSDF*xxx* member.
- Since the IMS system and the IMS Queue Control Facility extensions can share the same SCI in the same LPAR, you might want to set the IMS Queue Control Facility IMSPLEX(*name*) parameter to a name that defines the IMSplex, rather than use a name that defines the IMS Queue Control Facility extensions itself.

  For example, instead of the name IMS Queue Control Facility IMSPLEX(**QCF1**), use a name similar to IMS Queue Control Facility IMSPLEX(**IPLX1**).

This background topic supplements the following setup procedure: "1. Creating the SCI address space" on page 19.

Configuring more than one SCI address space per LPAR

You can have more than one SCI address space per LPAR, for example, one SCI for the IMS Queue Control Facility extensions and another SCI for the IMS system. This configuration is not recommended, but it is supported.

If you do choose to have a separate SCI address space for the IMS Queue Control Facility extensions and a second SCI address space for the IMS system you would need to define the IMSPLEX parameters as follows:

- SCI for the IMS Queue Control Facility extensions

  Use the same value for *name1* in the IMS Queue Control Facility extensions IMSPLEX(*name1*) parameter and the SCI address space IMSPLEX(NAME=*name1*) parameter.

  For example: IMSPLEX(QCF1) and IMSPLEX(NAME=QCF1)
- SCI for the IMS system

  Use the same value for *name2* in the IMS system IMSPLEX(*name2*) parameter and the SCI address space IMSPLEX(NAME=*name2*) parameter.

  For example: IMSPLEX(IPLX1) and IMSPLEX(NAME=IPLX1)

Starting up the SCI address space

When supporting IMS Queue Control Facility extensions, the SCI address space can be started before or after the IMS system is started, and before or after the IMS Queue Control Facility server address space is started. Refer to "6. Configuring and starting the IMS Queue Control Facility server" on page 28.

However, if the IMS system requires the SCI for other functions, such as IMS Operations Manager or IMS Resource Manager, it is recommended that the SCI address space be started before the IMS system, the IMS Queue Control Facility extensions, and the IMS Queue Control Facility server.

Because SCI is shipped with IMS, you need to create only the SCI address space startup JCL and the SCI/BPE configuration members.

Connecting IMS Queue Control Facility to SCI

Both IMS Queue Control Facility extensions and the IMS Queue Control Facility server connect to the Structured Call Interface (SCI) facility. Both the extensions and the sever must have completed connection processing for IMS Queue Control Facility to be fully connected to the SCI.

IMS Queue Control Facility is connected to the SCI and all IMS Queue Control Facility functions are fully functional only when the following message is displayed:

```
IQC6001I QCF EXTENSION ENABLED
```

IMS Queue Control Facility extensions allow you to start the SCI job and to specify the SCI job name that is started by IMS Queue Control Facility extensions when the SCIPROG= statement is specified in the IMS Queue Control Facility extensions configuration statements.

See "7. Enabling and configuring the IMS Queue Control Facility extensions" on page 31.

If message IQC6900I QCF IMS EXTENSION WAITING FOR SCI: (PLEX NAME: *plex_name*) is issued, the following conditions might be possible:

- The SCI was not started
- The SCI job failed after you specified the SCI job name in the IMS Queue Control Facility extensions configuration

  In this case, you must start the SCI job with an MVS start command, or submit the PROC from the PROCLIB to allow IMS Queue Control Facility extensions and the IMS Queue Control Facility server to connect to the SCI.

If message IQC6915I NO SCI PROC SPECIFIED is issued, then you must start the SCI job with an MVS start command or submit the PROC from the PROCLIB.

If the SCI terminates and is restarted (with an MVS start command or by submitting the PROC from the PROCLIB), the IMS Queue Control Facility server and IMS Queue Control Facility extensions will automatically reconnect.

Refer to (and modify if required) the SCI start JCL procedure (IQCSCI), the SCI BPE configuration file (BPECONF1), and the SCI INIT configuration file (CSLSI000) in the IMS Queue Control Facility *hlq*.SIQCSAMP data set.

Configuring threshold processing

When configuring threshold processing, there are two mechanisms that allow you to perform an action such as unloading all or selected messages from the IMS Message Queue when a threshold is crossed.

In the method descriptions that follow, the variable *nn* = (AB, or BC, or CD, or UT) for the ONTHRESHOLD*nn* parameter represents the threshold boundaries on the IMS Queue Control Facility extensions PROCLIB member ONTHRESHOLD*nn* statement.

- If the IMS Queue Control Facility server has been enabled and the ONTHRESHOLD*nn* PROCNAME=*procname* statement is specified in the IMS Queue Control Facility extensions member for this threshold, then the IMS Queue Control Facility extensions will send to the IMS Queue Control Facility server that PROCLIB member when the threshold is crossed.

  The IMS Queue Control Facility server processes the control statements in the *procname* and starts an IMS Queue Control Facility server batch job to perform requested action.

- If the IMS Queue Control Facility server is not enabled or the ONTHRESHOLD*nn* PROCNAME=*procname* statement is not specified in the IMS Queue Control Facility extensions member for this threshold, and the ONTHRESHOLD*nn* JOBNAME=*jobname* statement in the IMS Queue Control Facility extensions member is specified, the IMS Queue Control Facility extensions issues the MVS START command for the *jobname* when the threshold is crossed.

For proper operation, there are four sample JOBNAME members (IQCPRCAB, IQCPRCBC, IQCPRCCD, and IQCPRCUT) in the *hlq*.SIQCSAMP data set which contain the JCL that you must modify to adhere to

installation naming convention. These four members must be copied to a user PROCLIB and modified to meet installation standards and to perform the required function and message selection.

Each of these four sample JOBNAME member procedures contain the following DD statement:

```
//QCFIN    DD  DISP=SHR,DSN=IQC.mvsid.imsid.COMMIT(IQCQSNnn)
```

where IQCQSNnn supplies the control statements that define the functions and action to be taken when the threshold is crossed.

These four sample PROCLIB members (IQCQSNAB, IQCQSNBC, IQCQSNCD and IQCQSNUT) in the *hlq*.SIQCSAMP data set must be copied to the installation QCFIN data set to ensure that the JOBNAME procedure JCL is able to process them.

The JOBNAME member and the matching QCFIN member are listed below. IQCQSNnn contains the control statements that are referenced by the JOBNAME IQCPRCnn.

Edit *hlq*.SIQCSAMP(IQCPRCAB). This sample JCL for the IMS Queue Control Facility server is started by IMS when IMS message queue utilization goes from QSN area A to B.

Sample data set and member names:

- *hlq*.SIQCSAMP(IQCQSNAB) and *hlq*.SIQCSAMP(IQCPRCAB) (for crossing the threshold from area A to B)
- *hlq*.SIQCSAMP(IQCQSNBC) and *hlq*.SIQCSAMP(IQCPRCBC) (for crossing the threshold from area B to C)
- *hlq*.SIQCSAMP(IQCQSNCD) and *hlq*.SIQCSAMP(IQCPRCCD) (for crossing the threshold for area C to D)
- *hlq*.SIQCSAMP(IQCQSNUT) and *hlq*.SIQCSAMP(IQCPRCUT) (for crossing the threshold that you set)

The four members (IQCQSNAB, IQCQSNBC, IQCQSNCD and IQCQSNUT), that are shipped as samples control cards, can also be used as the QCFIN data set members for the four PROCNAME members.

When defining the IMS Queue Control Facility extension PROCLIB member ONTHERESHOLDnn statement, you can specify for QCFIN(DSN(*user_proclib*),MEM(*member_name*) the following QCFIN(DSN(USER.PRIVATE.PROCLIB),MEM(IQCQSNnn), where *nn* represents the AB, BC, CD, or UT threshold being crossed.

Here is an example of a PROCNAME member that is using the same QCFIN member (IQCQSNnn) as used by the JOBNAME defined above.

```
IMS(IMS1)
QCFIN(DSN(USER.PRIVATE.PROCLIB),MEM(IQCQSNnn))
QCFPRINT(QCFTEST.BATCH.QCFPRINT)
UNLOAD(QCFTEST.BATCH.UNLOAD)
```

This background topic supplements the following setup procedure: "2. Creating configuration members for message processing" on page 24.

Chapter 5, "Server and extension reference," on page 55 provides supplemental detail on PROCLIB members used during the setup procedures.

Overriding buffer definitions for HIOP

Each time you run the IMS Queue Control Facility functions by using ISPF panels or issuing JCL job streams, a buffer larger than 32 KB is required from the IMS high input/output pool (HIOP) for use by the IMS Queue Control Facility extensions, which run within the IMS control region. This HIOP buffer is released when the IMS Queue Control Facility functions end processing. It is therefore recommended that you define a 64 KB HIOP buffer from a performance point of view because the use of oversized buffers is costly and IMS recommends avoiding oversized traffic.

While the IMS control region is being started, IMS Queue Control Facility checks whether a 64 KB HIOP buffer is defined. If not, it issues warning message IQC6221W, and the startup processing continues.

If IQC6221W is issued, you might want to override HIOP pool definitions by either of the following methods and then restart IMS to make the changes effective:

- If the DFSSPM*xx* member of the IMS PROCLIB data set does not exist, or if DFSSPM*xx* exists but no HIOP buffers are not defined, add all of the following lines to DFSSPM*xx*:

```
FPL=HIOP,(256,64,32,N),(512,64,32,N),(1024,32,16,N)
FPL=HIOP,(2048,32,16,N),(4069,16,8,N),(8192,8,4,N)
FPL=HIOP,(16K,4,2,N),(32K,4,2,N),(64K,4,2,N)
```

- If the DFSSPM*xx* member of the IMS PROCLIB data set exists and an HIOP buffer of a size other than 64 KB is defined, add the following line to add a 64 KB HIOP buffer:

```
FPL=HIOP,(64K,4,2,N)
```

For more information about the DFSSPM*xx* member of the IMS PROCLIB data set, see *IMS System Definition*.

Note: You can run IMS Queue Control Facility functions without a 64 KB HIOP buffer defined. If performance tuning of IMS Queue Control Facility functions is not necessary, you can ignore the warning messages.

Chapter 5. Server and extension reference

The topics in this reference section provide supplemental information for the PROLIB members used in the IMS Queue Control Facility setup procedures.

Refer to Chapter 3, "Setting up IMS Queue Control Facility," on page 19 for complete details on using these PROCLIB members during IMS Queue Control Facility setup.

Topics:

- "Server configuration PROCLIB member reference" on page 55
- "Extensions PROCLIB member reference" on page 56
- "Extensions ONTHRESHOLDnn PROCLIB member reference" on page 73

Server configuration PROCLIB member reference

This topic describes the syntax and parameters for the IMS Queue Control Facility server configuration PROCLIB member.

Note: For an explanation of the PROCLIB member format, usage, and configuration, see "6. Configuring and starting the IMS Queue Control Facility server" on page 28.

```
      ┌─ IMSPLEX(IPLEX) ─┐   ┌─ AUTH(N) ─┐   ┌─ WAITTIME=60 ──┐
►►─────┤                  ├───┤           ├───┤                ├──►◄
       └─ IMSPLEX( name ) ┘   └─ AUTH(Y) ─┘   └─ WAITTIME= nnn ┘
```
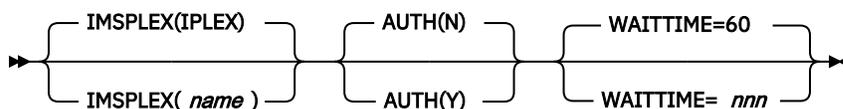
IMSPLEX(IPLEX | *name*)

Use the IMSPLEX parameter to specify a required IMSPLEX value that is a 1- to 5-character CSLPLEX name to which IMS Queue Control Facility communicates. This IMSPLEX name is used for communications between the IMS Queue Control Facility address spaces and the IMS Queue Control Facility extensions. If the current IMS systems do not use SCI, then do not change them for IMS Queue Control Facility.

Both the IMS Queue Control Facility server address spaces and the IMS subsystems that are manipulating IMS message queues must use the same SCI IMSPLEX(*plxnm*) name. (See the Restriction for providing a SCI IMSPLEX(*plxnm*) name.)

The value *name* must be the same name that is specified in the SCI initialization PROCLIB member, IMSPLEX(NAME=*plxnm*).

If you omit IMSPLEX or if no IMS Queue Control Facility extensions PROCLIB member exists, then the default name is set to IPLEX.

Restriction: To allow the *name* value for the IMS Queue Control Facility IMSPLEX statement to be defined and/or modified from the IMS Queue Control Facility TSO session, IMS Queue Control Facility has placed the following restriction on the SCI IMSPLEX(*plxnm*) parameter which is also used by other IMS components, including Resource Manager, Operations Manager, and IMS Connect.

This restriction states that the SCI IMSPLEX(*plxnm*) parameter must follow the MVS naming convention This convention consists of uppercase alphabetic characters (A through Z), special characters ($, #, and @), and numeric values (0 through 9). The first character of the name cannot be a numeric value. This MVS naming convention is not enforced by IMS Queue Control Facility. If the recommended MVS naming convention is not followed, you will not be able to specify a non-MVS naming convention name from the IMS Queue Control Facility TSO session.

This restriction might require modification to the IMSPLEX parameter for the SCI and those IMS components or other services that use the SCI interface. This restriction would only require modification to other IMS components and services that use the SCI if the IMSPLEX name does not meet the MVS naming convention.

Note: IMS Queue Control Facility has a member name of *serverid*, where *serverid* is the PROC name of the IMS Queue Control Facility server.

If the IMS SCI is terminated, message IQC7016I SERVER DISCONNECTED FROM SCI is issued. After the SCI has been restarted, the server automatically reregisters to the SCI, and message IQC7015I SERVER RECONNECTED TO SCI is issued.

AUTH(N | Y)

The AUTH parameter specifies the optional security option for IMS Queue Control Facility TSO users. Specify either character Y or N. The default is N.

The server provides security for TSO users based on the MVS name, IMS name, and the function that is being performed. The TSO user must be defined to the MVS of the server system. You can turn off security by specifying AUTH=N, when no SAF calls will be made.

WAITTIME=*nnn*

The WAITTIME parameter specifies the maximum time in seconds that the client is allowed to wait for the server to complete the entire function and respond back to the client with all messages that meet the criteria.

This parameter also represents the time in seconds that the server waits for the Control Region to return each single message that satisfies the criteria of the function request.

The server uses the same value as the client because there might be one and only one message that satisfies the complete request and that message might be the very last message queued.

The range of valid values is 7 to 999.

If this parameter is not specified, the default wait time is 60 seconds.

If this parameter is set to a value less than or equal to 6, it is set to the default value of 60 seconds.

The following example shows the server sample startup member with a WAITTIME of 24 seconds:

```
IMSPLEX=PLEX1
AUTH=N
WAITTIME=24
```

Extensions PROCLIB member reference

This topic describes the syntax and parameters for the IMS Queue Control Facility extensions PROCLIB member control statements.

Note: For an explanation of the PROCLIB member format, usage, and configuration, see "7. Enabling and configuring the IMS Queue Control Facility extensions" on page 31.

The **JOBNAME**=*jobname* parameter (for ONCOLDSTART, ONWARMSTART, ONTHRESHOLDAB, ONTHRESHOLDBC, ONTHRESHOLDCD, and ONTHRESHOLDUT) defines the name of a customer-supplied PROCLIB member for which the IMS Queue Control Facility extension issues an MVS **START** command.

The **PROCNAME**=*procname* parameter (for ONCOLDSTART, ONWARMSTART, ONTHRESHOLDAB, ONTHRESHOLDBC, ONTHRESHOLDCD, and ONTHRESHOLDUT) defines the name of a customer-supplied PROCLIB member which the IMS Queue Control Facility extension sends to the IMS Queue Control Facility server.

The **STARNAME**=*regionname* parameter (for ONCOLDSTART and ONWARMSTART) defines the name of a customer-supplied member for which the IMS Queue Control Facility extension issues an internal **/STA REG** command.

When IMS Queue Control Facility processes these parameters, the following results are possible:

- JOBNAME is used first, and a server is required
- If no server is available for JOBNAME, then PROCNAME is used
- If no PROCNAME was specified, then STARNAME is used
- STARNAME is only used when both JOBNAME and STARNAME are specified, but no server is available

- If PROCNAME and STARNAME are both specified, STARNAME is never used

When specifying control statements, use an equal sign (=), parentheses, or both to separate a control statement and its keyword parameters. For example, you can specify the WTODESTINATION statement in any of the following formats:

```
WTODESTINATION(WTO)
WTODESTINATION=WTO
WTODESTINATION=(WTO)
```

Control statements:

ACTIONPOOL control statement

Syntax

```
▶▶── ACTIONPOOL=( ──┬── MAXPOOL=1 ──┬── ) ──◄
                    │                │
                    └── MAXPOOL= nnn ┘
```

Keywords

MAXPOOL=nnn

The MAXPOOL parameter specifies the maximum number of QSUN cell pools. The valid range is 1 to 100. The default is 1.

One cell pool can take QSUN processing (ACTION=WAIT, ACTION=STOP, ACTION=WTO, and so on) about 100 times.

If all cell pools are used up, the messages are passed to IMS without QSUN processing. To activate QSUN processing again, IMS must be recycled.

AREA control statement

Syntax
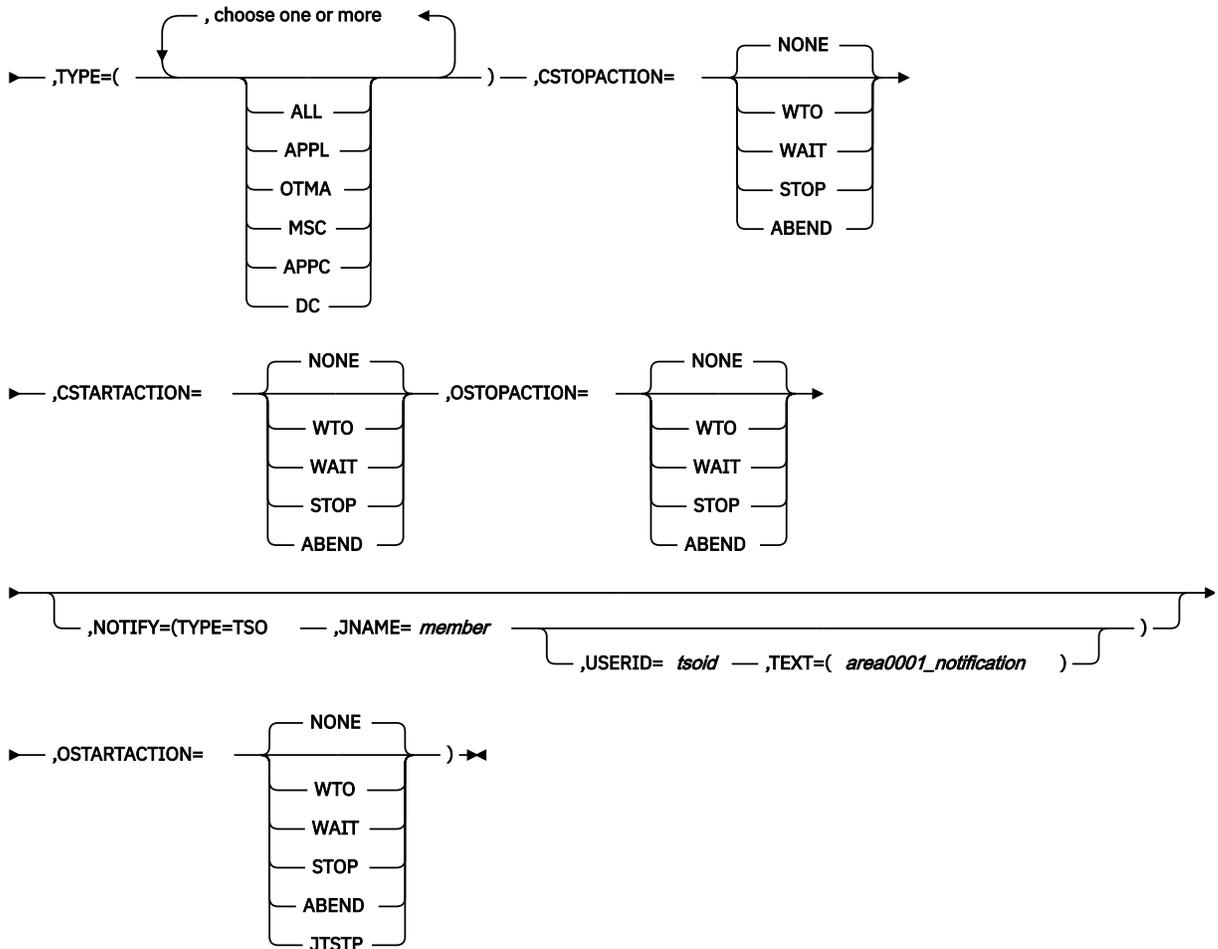
Usage notes

Up to ten AREA statements can be specified. All additional AREA statements after the first ten are ignored.

AREA statements must be in ascending order by their total percentage.

Note: If you use the TSO client to update QSUN, the client automatically puts the AREA cards in the correct sequence.

For information on configuring message queue overflow protection using AREA, see Chapter 9, "Message queue overflow protection," on page 85

For information about recovering from actions taken by AREA statements, see "Troubleshooting reference for AREA and FAILSAFE actions" on page 439.

Keywords

ID=*name*
The ID parameter specifies a 1- to 8-character ID used to identify the area. This name must be unique. Duplicate IDs will cause unexpected results.

A valid name consists of uppercase alphabetic characters (A through Z), special characters ($, #, or @), and numeric values (0 through 9). The first character of the name cannot be a numeric value.

PERCENT=(TOTAL=*nn*,USED=*nn*)
The required PERCENT parameter specifies a percentage value of 00 - 99.

The value for TOTAL specifies how much of the total queue will be looked at for analysis. A TOTAL value of 00 disables processing of the AREA statement.

The value for USED specifies how much of the area a process can use before implementing the specified action. A USED value of 00 disables processing of the AREA statement.

TYPE=*(type,type,...)*
The TYPE parameter specifies one or more source and destination types. Values are: APPL (dependent region application program), DC, APPC, OTMA, MSC, or ALL.

xxxxx*ACTION=*action
The following keywords are available for variable *action* in *xxxxx*ACTION=*action* (for example, CSTOPACTION=*action*):

NONE
Takes no action.

WTO
Issues IQC68*nn* messages as write-to-operator (WTO). WTO messages are also written to the job log of the control region or the dependent region.

WAIT
Waits for a user action.

You must use the TSO/ISPF Display Waited Tasks panel to end the wait.

In addition, IMS Queue Control Facility issues IQC68*nn* messages as write-to-operator (WTO). WTO messages are also written to the job log of the control region or the dependent region.

STOP
Conducts a Stop action, which can be one of the following:

- For TYPE=APPL (applications), issues an A7 status code to the application
- For TYPE=APPC (LU 6.2 /APPC), issues a DFS0777 message to the MTO and deallocates the input terminal
- For TYPE=DC (Data communications input terminal, BTAM / VTAM®), issues a DFS074 message to the input terminal
- For TYPE=OTMA, issues a NAK message to the client
- For TYPE=MSC, issues a DFS1945 message to the MTO, issues a DSF2140 message to the MTO in the partner IMS system, and stops the sending path MSNAME

In addition, IMS Queue Control Facility issues IQC68*nn* messages as write-to-operator (WTO). WTO messages are also written to the job log of the control region or the dependent region.

ABEND
Terminates the application in the dependent region abnormally with abend code U405.

In addition, IMS Queue Control Facility issues IQC68*nn* messages as write-to-operator (WTO). WTO messages are also written to the job log of the dependent region.

ABEND is applicable to TYPE=APPL only; this keyword is treated as STOP for other destination types.

JTSTP
Returns an A7 status code to an IMS process when the Queue Manager resources that are allocated to a destination have exceeded the AREA statement specifications.

The following IMS messages are sent to different IMS input devices or MTO:

- VTAM: DFS074 message

- APPC: DFS0777 message
- OTMA: NAK message
- BTAM: DFS074 message
- MSC: DFS1945 message

CSTOPACTION=*action*

The CSTOPACTION parameter specifies the action to be taken if the source process that sends messages and causes excessive queue usage is in a stopped state. This parameter additionally specifies the same action to be taken for the destination when the source process has caused an excessive message count on the queue. The parameter specifies one of the following actions to be taken against the source and destination processes: NONE, ABEND, WTO, STOP, or WAIT. The default is NONE.

Example sources: a PST (partition specification table - an IMS control block that represents a dependent region) or a CLB (communication line block - an IMS control block that represents a VTAM node or a BTAM line).

Example destinations: an SMB (scheduler message block - an IMS control block that represents a transaction) or CNT (communication name table - an IMS control block that represents a logical terminal).

Table 5. Valid source analysis actions for CSTOPACTION

TYPE (cause)	AREA statement	Valid actions (taken against the source)
APPL	CSTOPACTION	Not applicable (see note)
DC	CSTOPACTION	Not applicable (see note)
APPC	CSTOPACTION	Not applicable (see note)
OTMA	CSTOPACTION	Not applicable (see note)
MSC	CSTOPACTION	Not applicable (see note)

Note: If a message buffer has been assigned, then the inputting (causing) sources cannot be in a stopped state. Therefore, CSTOPACTION is not applicable.

Table 6. Valid destination analysis actions for CSTOPACTION

TYPE (cause)	AREA statement	Valid actions (taken against the destination)
APPL	CSTOPACTION	NONE, WTO (see note)
DC	CSTOPACTION	NONE, WTO (see note)
APPC	CSTOPACTION	NONE, WTO (see note)
OTMA	CSTOPACTION	NONE, WTO (see note)
MSC	CSTOPACTION	NONE, WTO (see note)

Note: When a destination is either started or stopped, the only appropriate action is to issue a WTO to notify the administrator that a destination has exceeded the threshold specified by the AREA statement.

The ABEND, STOP, and WAIT actions, if specified, default to WTO.

CSTARTACTION=*action*

The CSTARTACTION parameter specifies the action to be taken if the source process that sends messages and causes excessive queue usage is in a started state. This parameter additionally specifies the same action to be taken for the destination when the source process has caused an excessive message count on the queue. The parameter specifies one of the following actions to be

taken against the source and destination processes: NONE, ABEND, WTO, STOP, or WAIT. The default is NONE.

Example sources: a PST (partition specification table - an IMS control block that represents a dependent region) or a CLB (communication line block - an IMS control block that represents a VTAM node or a BTAM line).

Example destinations: an SMB (scheduler message block - an IMS control block that represents a transaction) or CNT (communication name table - an IMS control block that represents a logical terminal).

Table 7. Valid source analysis actions for CSTARTACTION

TYPE (cause)	AREA statement	Valid actions (taken against the source)
APPL	CSTARTACTION	NONE, ABEND, WTO, STOP, WAIT
DC	CSTARTACTION	NONE, ABEND, WTO, STOP, WAIT
APPC	CSTARTACTION	NONE, ABEND, WTO, STOP, WAIT
OTMA	CSTARTACTION	NONE, ABEND, WTO, STOP, WAIT
MSC	CSTARTACTION	NONE, ABEND, WTO, STOP, WAIT

Table 8. Valid destination analysis actions for CSTARTACTION

TYPE (cause)	AREA statement	Valid actions (taken against the destination)
APPL	CSTARTACTION	NONE, WTO (see note)
DC	CSTARTACTION	NONE, WTO (see note)
APPC	CSTARTACTION	NONE, WTO (see note)
OTMA	CSTARTACTION	NONE, WTO (see note)
MSC	CSTARTACTION	NONE, WTO (see note)

Note: When a destination is either started or stopped, the only appropriate action is to issue a WTO to notify the administrator that a destination has exceeded the threshold specified by the AREA statement.

The ABEND, STOP, and WAIT actions, if specified, default to WTO.

NOTIFY=(TYPE=TSO,JNAME=*member*,USERID=*tsoid*,TEXT=(*area0001_notification*)
The optional NOTIFY parameter sends notifications to the TSO user IDs when the AREA or FAILSAFE parameters are triggered.

Multiple NOTIFY parameters can be added to each AREA control statement. The NOTIFY parameter sets a job to be started when the ACTION for the AREA is taken. The job sends notifications to TSO user.

TYPE=TSO
This required subparameter is the type of the started JCL procedure. It is not checked and it can be used to perform other services.

JNAME=*jobname*
This required subparameter is the name of a procedure in the PROCLIB DD concatenation for the control region. This procedure is started before the requested action for the AREA. The procedure is not started if the requested action is NONE.

USERID=*tsoid*
This optional subparameter, set in the started procedure, is the designated TSO user ID that receives the notifications. This subparameter is informational only.

TEXT=*area0001_notification*

> This optional subparameter, set in the started procedure, sends text to the TSO user ID. This subparameter is informational only.

OSTOPACTION=*action*

> The OSTOPACTION parameter identifies the action to take if another (other) source process, that is found to be responsible for causing the queue to exceed the queue usage specified by the AREA statement, is in a stopped state. The parameter specifies one of the following actions to be taken against the source process: NONE, ABEND, WTO, STOP, or WAIT. The default is NONE.
>
> Example sources: a PST (partition specification table - an IMS control block that represents a dependent region) or a CLB (communication line block - an IMS control block that represents a VTAM node or a BTAM line).
>
> Example destinations: an SMB (scheduler message block - an IMS control block that represents a transaction) or CNT (communication name table - an IMS control block that represents a logical terminal).

Table 9. Valid source analysis actions for OSTOPACTION

TYPE (other)	AREA statement	Valid actions (taken against the source)
APPL	OSTOPACTION	Not applicable (see note)
DC	OSTOPACTION	Not applicable (see note)
APPC	OSTOPACTION	Not applicable (see note)
OTMA	OSTOPACTION	Not applicable (see note)
MSC	OSTOPACTION	Not applicable (see note)

Note: If a message buffer has been assigned, then the inputting (other) sources cannot be in a stopped state. Therefore, OSTOPACTION is not applicable.

Table 10. Valid destination analysis actions for OSTOPACTION

TYPE (cause)	AREA statement	Valid actions (taken against the destination)
APPL	OSTOPACTION	NONE, WTO (see note)
DC	OSTOPACTION	NONE, WTO (see note)
APPC	OSTOPACTION	NONE, WTO (see note)
OTMA	OSTOPACTION	NONE, WTO (see note)
MSC	OSTOPACTION	NONE, WTO (see note)

Note: When a destination is either started or stopped, the only appropriate action is to issue a WTO to notify the administrator that a destination has exceeded the threshold specified by the AREA statement.

The ABEND, STOP, and WAIT actions, if specified, default to WTO.

OSTARTACTION=*action*

> The OSTARTACTION parameter identifies the action to take if another (other) source process, that is found to be responsible causing the queue to exceed the queue usage specified by the AREA statement (even if by only one message), is in a started state. The parameter specifies one of the following actions to be taken: NONE, ABEND, WTO, STOP, JTSTP, or WAIT. The default is NONE.
>
> Example sources: a PST (partition specification table - an IMS control block that represents a dependent region) or a CLB (communication line block - an IMS control block that represents a VTAM node or a BTAM line).

Example destinations: an SMB (scheduler message block - an IMS control block that represents a transaction) or CNT (communication name table - an IMS control block that represents a logical terminal).

Table 11. Valid source analysis actions for OSTARTACTION

TYPE (other)	AREA statement	Valid actions (taken against the source)
APPL	OSTARTACTION	NONE, WTO, JTSTP (see note)
DC	OSTARTACTION	NONE, WTO, JTSTP (see note)
APPC	OSTARTACTION	NONE, WTO, JTSTP (see note)
OTMA	OSTARTACTION	NONE, WTO, JTSTP (see note)
MSC	OSTARTACTION	NONE, WTO, JTSTP (see note)

The ABEND, STOP, and WAIT actions, if specified, default to WTO.

See "Using OSTARTACTION=JTSTP" on page 115.

Table 12. Valid destination analysis actions for OSTARTACTION

TYPE (cause)	AREA statement	Valid actions (taken against the destination)
APPL	OSTARTACTION	NONE, WTO (see note)
DC	OSTARTACTION	NONE, WTO (see note)
APPC	OSTARTACTION	NONE, WTO (see note)
OTMA	OSTARTACTION	NONE, WTO (see note)
MSC	OSTARTACTION	NONE, WTO (see note)

Note: When a destination is either started or stopped, the only appropriate action is to issue a WTO to notify the administrator that a destination has exceeded the threshold specified by the AREA statement.

The ABEND, STOP, and WAIT actions, if specified, default to WTO.

Examples

See the following examples of the AREA control statement:

```
AREA=(ID=AREA0001,
      PERCENT=(TOTAL=50,USED=10),
      TYPE=(ALL),
NOTIFY=(TYPE=TSO,JNAME=IQCSNTFY,USERID=USRT001,TEXT=(AREA0001 NOTIFIC)),
      CSTOPACTION=WTO,
      CSTARTACTION=WTO,
      OSTOPACTION=WTO,
      OSTARTACTION=WTO)
```

```
AREA=(ID=AREA0002,
      PERCENT=(TOTAL=60,USED=10),
      TYPE=(ALL),
      CSTOPACTION=WTO,
      CSTARTACTION=WTO,
      OSTOPACTION=WTO,
      OSTARTACTION=JTSTP)
```

FAILSAFE control statement

Syntax

▶▶── FAILSAFE=(ID= *name* ── ,PERCENT=(TOTAL= *nn* ── ,USED= *nn*) ──▶

                          ┌── NONE ──┐
   ├─ ,ACTION= ─┼── WTO ──┼──▶
                   ├── WAIT ──┤
                   ├── STOP ──┤
                   └── ABEND ──┘

▶───▶
    └─ ,NOTIFY=(TYPE=TSO ── ,JNAME= *member* ────────────────────────────)─┘
                       └─ ,USERID= *tsoid* ── ,TEXT=(*area0001_notification*) ─┘

├──) ──◀
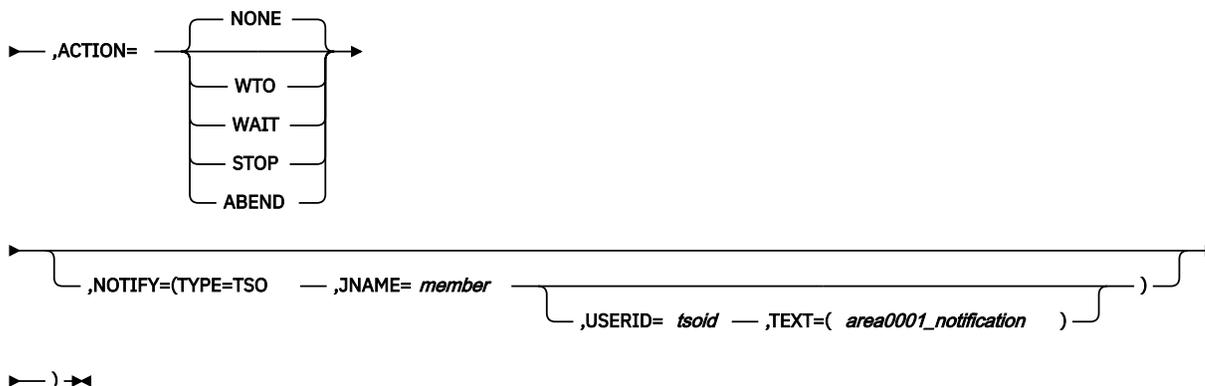
Usage notes

Up to 10 FAILSAFE statements can be specified. Any additional FAILSAFE statements after the first 10 are ignored.

For information about configuring message queue overflow protection using FAILSAFE, see Chapter 9, "Message queue overflow protection," on page 85.

For information about recovering from actions taken by FAILSAFE statements, see "Troubleshooting reference for AREA and FAILSAFE actions" on page 439.

IMS entity type used for FAILSAFE analysis

FAILSAFE does not allow targeting of specific IMS entities; therefore, the TYPE= parameter is not allowed. An implicit TYPE=ALL is enforced, which includes the following entities:

TYPE=APPL
A dependent region application program

TYPE=DC
A VTAM terminal

TYPE=APPC
An LU 6.2/APPC application

TYPE=OTMA
An OTMA application

TYPE=MSC
An IMS-to-IMS communications page

Keywords

ID=*name*
The ID parameter specifies a 1- to 8-character ID used to identify the FAILSAFE. This ID must be unique. Duplicate IDs will cause unexpected results.

A valid name consists of uppercase alphabetic characters (A through Z), special characters ($, #, or @), and numeric values (0 through 9). The first character of the name cannot be a numeric value.

PERCENT=(TOTAL=*nn*,USED=*nn*)
The required PERCENT parameter specifies a percentage value of 00 - 99.

TOTAL specifies how much of the total queue will be looked at for analysis. A TOTAL value of 00 causes the FAILSAFE processing to check the USED value. After the USED value is reached, the specified action will be taken.

USED specifies how much of the queue a process can use before implementing the specified action. A USED value of 00 causes the specified action to be implemented immediately.

ACTION=*action*
> Values for the ACTION parameter are: NONE, WTO, WAIT, STOP, ABEND.
>
> The following keywords are available for variable *action*:
>
> **NONE**
>> Takes no action.
>
> **WTO**
>> Issues IQC68*nn* messages as write-to-operator (WTO). WTO messages are also written to the job log of the control region or the dependent region.
>>
>> **Exception:** When a FAILSAFE condition is met, only one WTO is issued. The purpose of this condition is to ensure that the MCS is not overwhelmed with notification WTOs.
>
> **WAIT**
>> Waits for a user action.
>>
>> You must use the TSO/ISPF Display Waited Tasks panel to end the wait.
>>
>> In addition, IMS Queue Control Facility issues IQC68*nn* messages as write-to-operator (WTO). WTO messages are also written to the job log of the control region or the dependent region.
>
> **STOP**
>> Conducts a Stop action, which can be one of the following:
>>
>> - For TYPE=APPL (applications), issues an A7 status code to the application
>> - For TYPE=APPC (LU 6.2 /APPC), issues a DFS0777 message to the MTO and deallocates the input terminal
>> - For TYPE=DC (Data communications input terminal, BTAM / VTAM), issues a DFS074 message to the input terminal
>> - For TYPE=OTMA, issues a NAK message to the client
>> - For TYPE=MSC, issues a DFS1945 message to the MTO, issues a DSF2140 message to the MTO in the partner IMS system, and stops the sending path MSNAME
>>
>> In addition, IMS Queue Control Facility issues IQC68*nn* messages as write-to-operator (WTO). WTO messages are also written to the job log of the control region or the dependent region.
>
> **ABEND**
>> Terminates the application in the dependent region abnormally with abend code U405.
>>
>> In addition, IMS Queue Control Facility issues IQC68*nn* messages as write-to-operator (WTO). WTO messages are also written to the job log of the dependent region.
>>
>> ABEND is applicable to TYPE=APPL only; this keyword is treated as STOP for other destination types.

NOTIFY=(TYPE=TSO,JNAME=*member*,USERID=*tsoid*,TEXT=(*area0001_notification*))
> The optional NOTIFY parameter sends notifications to the TSO user IDs when the AREA or FAILSAFE parameters are triggered.
>
> Multiple NOTIFY parameters can be added to each AREA control statement. The NOTIFY parameter sets a job to be started when the ACTION for the AREA is taken. The job sends notifications to TSO user.
>
> **TYPE=TSO**
>> This required subparameter, is the type of the started JCL procedure. It is not checked and it can be used to perform other services.

JNAME=_jobname_
 This required subparameter, is the name of a procedure in the PROCLIB DD concatenation for the
 control region. This procedure is started before the requested action for the AREA. The procedure
 is not started if the requested action is NONE.

USERID=_tsoid_
 This optional subparameter, set in the started procedure, is the designated TSO user ID that
 receives the notifications. This subparameter is informational only.

TEXT=_area0001_notification_
 This optional subparameter, set in the started procedure, sends text to the TSO user ID. This
 subparameter is informational only.

Examples

See the following examples of the FAILSAFE control statement:

```
FAILSAFE=(ID=FAIL0001,
          PERCENT=(TOTAL=40,USED=04),
          ACTION=WTO)
```

```
FAILSAFE=(ID=FAIL0099,PERCENT=(TOTAL=82,USED=00),
NOTIFY=(TYPE=TSO,JNAME=IQCNTM2T,USERID=TEXTME,TEXT=(FAIL0099 NOTIFY)),
      ACTION=STOP)
```

IMSPLEX control statement

Syntax
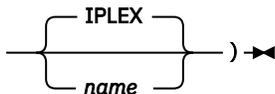
Usage notes

The required IMSPLEX statement specifies a 1- to 5-character CSLPLEX name to which IMS Queue
Control Facility communicates. This IMSPLEX name is used for communications between the IMS Queue
Control Facility server address spaces and the IMS Queue Control Facility IMS extensions. If the current
IMS systems do not use SCI, do not change them for IMS Queue Control Facility.

Both the IMS Queue Control Facility server address spaces and the IMS subsystems that are manipulating
IMS message queues must use the same SCI IMSPLEX(_plxnm_) name. (See the Restriction for providing a
SCI IMSPLEX(_plxnm_) name.)

The value _name_ must be the same name as specified in the SCI initialization PROCLIB member,
IMSPLEX(NAME=_plxnm_).

If you omit IMSPLEX or if no IMS Queue Control Facility extensions PROCLIB member exists, then the
default name is set to IPLEX.

Restriction: To allow the (_name_) value for the IMS Queue Control Facility IMSPLEX statement to be
defined and/or modified from the IMS Queue Control Facility TSO session, IMS Queue Control Facility has
placed the following restriction on the SCI IMSPLEX(_plxnm_) parameter which is also used by other IMS
components, including Resource Manager, Operations Manager, and IMS Connect.

This restriction states that the SCI IMSPLEX(_plxnm_) parameter must follow the MVS naming convention.
This convention consists of uppercase alphabetic characters (A through Z), special characters ($, #, and
@), and numeric values (0 through 9). The first character of the name cannot be a numeric value. This
MVS naming convention is not enforced by IMS Queue Control Facility. If the recommended MVS naming
convention is not followed, you will not be able to specify a non-MVS naming convention name from the
IMS Queue Control Facility TSO session.

This restriction might require modification to the IMSPLEX= (*name*) parameter for the SCI and those IMS components or other services that use the SCI interface. This restriction would only require modification to other IMS components and services that use the SCI if the IMSPLEX name does not meet the MVS naming convention.

Note: IMS Queue Control Facility has a member name of IQCF*imsid*, where *imsid* is the IMS ID of the IMS control region.

For example, if an IMS whose IMS ID is IMSA is started with the IMS Queue Control Facility extensions, the member name for that IMS is IQCFIMSA.

If the IMS SCI stops, message IQC6019I EXTENSION DISCONNECTED FROM SCI, IMSID=*imsid* is issued. After the SCI has been restarted, the IMS Queue Control Facility extension automatically reregisters to the SCI, and message IQC6018I EXTENSION RECONNECTED to SCI IMSID=*imsid* is issued.

ONABEND control statement

The ONABEND statement specifies the action to take if an unexpected IMS Queue Control Facility abend occurs in the IMS control region.

Syntax

```
►►─ ONABEND=(  ──┬── DUMP ───┬──  )─►◄
                 └── NODUMP ──┘
```
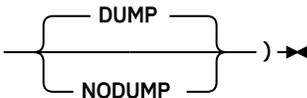
Keywords

The following keywords are available:

DUMP
IMS Queue Control Facility issues an SDUMP of the IMS control region and any associated IMS Queue Control Facility address space if IMS Queue Control Facility resides on the same LPAR as the IMS on which the abend occurred. When the SDUMP is complete, IMS Queue Control Facility terminates processing in the IMS control region. IMS remains active and operational; however, no further IMS Queue Control Facility manipulation of IMS message queues can be done by that IMS until the IMS Queue Control Facility extensions are restarted.

NODUMP
IMS Queue Control Facility does not issue an SDUMP. IMS Queue Control Facility terminates processing in the IMS control region, and IMS subsystem remains active and operational. No further IMS Queue Control Facility manipulation of IMS message queues can be done by that IMS until the IMS Queue Control Facility extensions are restarted.

This NODUMP option provides for the minimum disruption of IMS with no interference from SDUMP processing. However, NODUMP might result in no diagnostics being available to solve the abend.

ONCOLDSTART control statement

Syntax

▶▶── ONCOLDSTART=(──→

```
                              NOREQ
            ┌ JOBNAME= ──┬──────────┬── ,PROCNAME= procname ── ,STARNAME= regionname ──┐
            │            └ jobname ──┘                                                  │
```

────) ◀◆
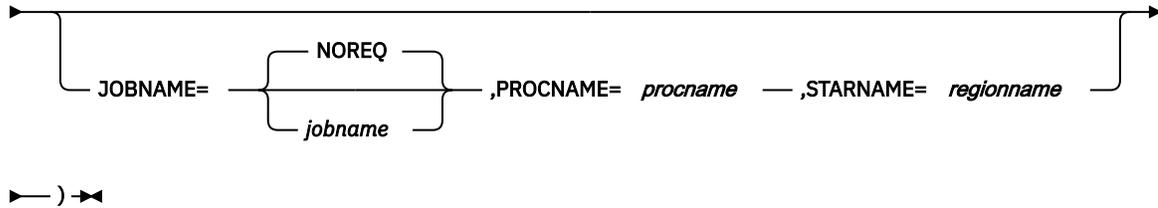
Keywords

JOBNAME=NOREQ
> The NOREQ value specifies that no action is taken on cold start.

JOBNAME=*jobname*
> The *jobname* value specifies a 1- to 8-character name of the IMS Queue Control Facility PROCLIB member that is started by IMS IMS Queue Control Facility extensions to load your data.
>
> This PROCLIB member contains the JCL for an IMS Queue Control Facility batch job.
>
> A valid name consists of uppercase alphabetic characters (A through Z) special characters ($,#, or @), and numeric values (0 through 9). The first character of the name cannot be a numeric value.

PROCNAME=*procname*
> The *procname* value specifies a 1- to 8-character name of the IMS Queue Control Facility PROCLIB member that is sent by IMS Queue Control Facility extensions to the IMS Queue Control Facility server to load your data. The IMS Queue Control Facility server opens this member, finds the control statements, and processes the actions that are specified.
>
> A valid name consists of uppercase alphabetic characters (A through Z) special characters ($,#, or @), and numeric values (0 through 9). The first character of the name cannot be a numeric value.

STARNAME=*regionname*
> The *regionname* value specifies a 1- to 8-character name of the region that is started by an internal **/STA REG** command that loads your data. This region member contains the JCL for an IMS Queue Control Facility batch job.
>
> A valid name consists of uppercase alphabetic characters (A through Z) special characters ($,#, or @), and numeric values (0 through 9). The first character of the name cannot be a numeric value.

Note: This statement only starts the user-specified job or procedure after the user runs one of the following IMS cold starts or emergency restarts:

1. **NRE CHECKPOINT 0**
2. **ERE COLDSYS** | **ERE COLDSYS OVERRIDE**
3. **ERE COLDCOMM** | **ERE COLDCOMM OVERRIDE**

ONCOLDSTART starts the specified job or procedure after an emergency restart with COLDSYS or COLDCOMM specified. These two emergency restart variations do not rebuild the queues (unlike **/ERESTART** or **/ERESTART COLDBASE**).

ONINITERR control statement

Syntax

```
▶▶─ ONINITERR=(      ┌─ TERM ─┐       )─◄►
                     ├────────┤
                     └─ ABEND ─┘
```
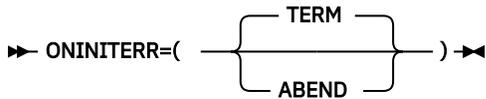
Keywords

ONINITERR(<u>TERM</u> | ABEND)
> The ONINITERR parameter specifies the action to take if a non-abend type of error occurs during IMS Queue Control Facility initialization in the IMS control region (for example, if IMS Queue Control Facility cannot obtain storage for blocks or buffers).

> **TERM**
>> IMS Queue Control Facility issues an error message and ends processing in the control region. IMS Queue Control Facility processing is not available for this control region until IMS Queue Control Facility is restarted. The IMS control region continues initialization and is available.

> **ABEND**
>> IMS Queue Control Facility issues a user abend that causes the IMS control region to terminate. You must restart IMS.

> **Note:** This statement applies only to errors that are detected after the IMS Queue Control Facility IMS extensions PROCLIB member is parsed because parsing must be complete for IMS Queue Control Facility to understand what was coded on ONINITERR. Errors that occur before the successful parsing of the PROCLIB member are treated as if ONINITERR(TERM) was coded.

> Non-abend errors that occur after IMS Queue Control Facility is initialized in the control region are recovered from and do not cause either IMS Queue Control Facility extensions or IMS to terminate.

ONTHRESHOLDAB control statement

Syntax

```
▶▶─ ONTHRESHOLDAB=(   ┌─ JOBNAME=NOUNLOAD ─┐                             )─◄►
                      │                    │
                      └─ JOBNAME= jobname ─┘  └─ ,PROCNAME= procname ─┘
```

Keywords

(JOBNAME=NOUNLOAD | *jobname***,PROCNAME=***procname***)**
> The JOBNAME parameter specifies the name of the IMS Queue Control Facility PROCLIB member defined in the overflow table. This IMS Queue Control Facility PROCLIB member is either the IMS Queue Control Facility default PROCLIB member or the updated PROCLIB member provided from the TSO session.

> If both a job name and a procedure name are specified for this threshold, the procedure name will be used.

> **JOBNAME=NOUNLOAD**
>> The NOUNLOAD value specifies that no action is taken when the threshold is crossed.

> **JOBNAME=***jobname*
>> The *jobname* value specifies a 1- to 8-character name of the IMS Queue Control Facility PROCLIB member that is started by IMS Queue Control Facility extensions when an offload of the queue is required. If NOUNLOAD is not specified, then the action that is taken by IMS Queue Control Facility extensions is to submit the job name that is specified in the overflow table definition at the

time the AB threshold is crossed. A sample member (IQCPRCAB) is provided in the *hlq*.SIQCSAMP data set.

This PROCLIB member contains the JCL for an IMS Queue Control Facility batch job.

A valid name consists of uppercase alphabetic characters (A through Z) special characters ($,#, or @), and numeric values (0 through 9). The first character of the name cannot be a numeric value.

PROCNAME=*procname*
The *procname* value specifies a 1- to 8-character name of the IMS Queue Control Facility PROCLIB member that is sent by IMS Queue Control Facility extensions to the IMS Queue Control Facility server when an offload of the queue is required at the crossover of threshold AB. The IMS Queue Control Facility server opens this member, finds the control statements, and processes the offload of the queue.

A valid name consists of uppercase alphabetic characters (A through Z) special characters ($,#, or @), and numeric values (0 through 9). The first character of the name cannot be a numeric value.

ONTHRESHOLDBC control statement

Syntax

```
                                JOBNAME=NOUNLOAD
►►─ ONTHRESHOLDBC=( ─────┬────────────────────────┬──────────────────────────┬─ )─►◄
                         │                        │                          │
                         └─ JOBNAME= jobname ─────┘   └─ ,PROCNAME= procname ─┘
```

Keywords

(JOBNAME=NOUNLOAD |*jobname***,PROCNAME=***procname***)**
The JOBNAME parameter specifies the name of the IMS Queue Control Facility PROCLIB member defined in the overflow table. This IMS Queue Control Facility PROCLIB member is either the IMS Queue Control Facility default PROCLIB member or the updated PROCLIB member provided from the TSO session.

If both a job name and a procedure name are specified for this threshold, the procedure name will be used.

JOBNAME=NOUNLOAD
The NOUNLOAD value specifies that no action is taken when the threshold is crossed.

JOBNAME=*jobname*
The *jobname* value specifies a 1- to 8-character name of the IMS Queue Control Facility PROCLIB member that is started by IMS Queue Control Facility extensions when an offload of the queue is required. If NOUNLOAD is not specified, then the action that is taken by IMS Queue Control Facility extensions is to submit the job name that is specified in the overflow table definition at the time the BC threshold is crossed. A sample member (IQCPRCBC) is provided in the *hlq*.SIQCSAMP data set.

A valid name consists of uppercase alphabetic characters (A through Z) special characters ($,#, or @), and numeric values (0 through 9). The first character of the name cannot be a numeric value.

PROCNAME=*procname*
The *procname* value specifies a 1- to 8-character name of the IMS Queue Control Facility PROCLIB member that is sent by IMS Queue Control Facility extensions to the IMS Queue Control Facility server when an offload of the queue is required at the crossover of threshold BC. The IMS Queue Control Facility server opens this member, finds the control statements, and processes the offload of the queue.

A valid name consists of uppercase alphabetic characters (A through Z) special characters ($,#, or @), and numeric values (0 through 9). The first character of the name cannot be a numeric value.

ONTHRESHOLDCD control statement

Syntax

```
►►─ ONTHRESHOLDCD=(  ┌─── JOBNAME=NOUNLOAD ───┐                          )─►◄
                     └─── JOBNAME= jobname ───┘  └─ ,PROCNAME= procname ─┘
```

Keywords

(JOBNAME=NOUNLOAD | *jobname***,PROCNAME=***procname***)**
>   The JOBNAME parameter specifies the name of the IMS Queue Control Facility PROCLIB member defined in the overflow table. This IMS Queue Control Facility PROCLIB member is either the IMS Queue Control Facility default PROCLIB member or the updated PROCLIB member provided from the TSO session.
>
>   If both a job name and a procedure name are specified for this threshold, the procedure name will be used.

>   **JOBNAME=NOUNLOAD**
>   >   The NOUNLOAD value specifies that no action is taken when the threshold is crossed.

>   **JOBNAME=***jobname*
>   >   The *jobname* value specifies a 1- to 8-character name of the IMS Queue Control Facility PROCLIB member that is started by IMS Queue Control Facility extensions when an offload of the queue is required. If NOUNLOAD is not specified, then the action that is taken by IMS Queue Control Facility extensions is to submit the job name that is specified in the overflow table definition at the time the CD threshold is crossed. A sample member (IQCPRCCD) is provided in the *hlq*.SIQCSAMP data set.
>   >
>   >   A valid name consists of uppercase alphabetic characters (A through Z) special characters ($,#, or @), and numeric values (0 through 9). The first character of the name cannot be a numeric value.

>   **PROCNAME=***procname*
>   >   The *procname* value specifies a 1- to 8-character name of the IMS Queue Control Facility PROCLIB member that is sent by IMS Queue Control Facility extensions to the IMS Queue Control Facility server when an offload of the queue is required at the crossover of threshold CD. The IMS Queue Control Facility server opens this member, finds the control statements, and processes the offload of the queue.
>   >
>   >   A valid name consists of uppercase alphabetic characters (A through Z) special characters ($,#, or @), and numeric values (0 through 9). The first character of the name cannot be a numeric value.

ONTHRESHOLDUT control statement

Syntax

```
►►─ ONTHRESHOLDUT=(  ┌─── JOBNAME=NOUNLOAD ───┐                          )─►◄
                     └─── JOBNAME= jobname ───┘  └─ ,PROCNAME= procname ─┘
```

Keywords

JOBNAME=NOUNLOAD | *jobname***,PROCNAME=***procname*
>   The JOBNAME parameter specifies the name of the IMS Queue Control Facility PROCLIB member defined in the overflow table. This IMS Queue Control Facility PROCLIB member is either the IMS Queue Control Facility default PROCLIB member or the updated PROCLIB member provided from the TSO session.

If both a job name and a procedure name are specified for this threshold, the procedure name will be used.

JOBNAME=NOUNLOAD
The NOUNLOAD value specifies that no action is taken when the threshold is crossed.

JOBNAME=*jobname*
The *jobname* value specifies a 1- to 8-character name of the IMS Queue Control Facility PROCLIB member that is started by IMS Queue Control Facility extensions when an offload of the queue is required. If NOUNLOAD is not specified, then the action taken by IMS Queue Control Facility extensions is to submit the job name that is specified in the overflow table definition at the time that the threshold is crossed. A sample member (IQCPRCUT) is provided in the *hlq*.SIQCSAMP data set.

A valid name consists of uppercase alphabetic characters (A through Z) special characters ($,#, or @), and numeric values (0 through 9). The first character of the name cannot be a numeric value.

PROCNAME=*procname*
The *procname* value specifies a 1- to 8-character name of the IMS Queue Control Facility PROCLIB member that is sent by IMS Queue Control Facility extensions to the IMS Queue Control Facility server when an offload of the queue is required at the crossover of threshold that you defined. The IMS Queue Control Facility server opens this member, finds the control statements, and processes the offload of the queue.

A valid name consists of uppercase alphabetic characters (A through Z) special characters ($,#, or @), and numeric values (0 through 9). The first character of the name cannot be a numeric value.

ONWARMSTART control statement

Syntax


```
                    JOBNAME=NOREQ
►►─ ONWARMSTART=(──┬────────────────────┬──┬──────────────────────┬──┬──────────────────────┬──)─►◄
                   └─ JOBNAME= jobname ─┘  └─ ,PROCNAME= procname ─┘  └─ ,STARNAME= regionname ─┘
```

Keywords

JOBNAME=NOREQ
The NOREQ value specifies that no action is taken on warm start.

JOBNAME=*jobname*
The *jobname* value specifies a 1- to 8-character name of the IMS Queue Control Facility PROCLIB member that is started by IMS Queue Control Facility extensions when the action that you specified is taken.

This PROCLIB member contains the JCL for an IMS Queue Control Facility batch job.

A valid name consists of uppercase alphabetic characters (A through Z) special characters ($,#, or @), and numeric values (0 through 9). The first character of the name cannot be a numeric value.

PROCNAME=*procname*
The *procname* value specifies a 1- to 8-character name of the IMS Queue Control Facility PROCLIB member that is sent by IMS Queue Control Facility extensions to the IMS Queue Control Facility server when the action that you specified is taken. The IMS Queue Control Facility server opens this member, finds the control statements, and processes the action that are specified.

STARNAME=*regionname*
The *regionname* value specifies a 1- to 8-character name of the region that is started by an internal **/STA REG** command that loads your data. This region member contains the JCL for an IMS Queue Control Facility batch job.

A valid name consists of uppercase alphabetic characters (A through Z) special characters ($,#, or @), and numeric values (0 through 9). The first character of the name cannot be a numeric value.

SCIPROC control statement

The SCIPROC statement specifies a 1- to 8-character value for the name of the IMS Structured Call Interface (SCI) PROCLIB member that is started by IMS Queue Control Facility extensions to start the IMS SCI address space.

Syntax

▶▶── SCIPROC= *proc_name* ──▶◀

Keywords

SCIPROC=*proc_name*
> Specifies a 1- to 8-character value for the name of the IMS Structured Call Interface (SCI) PROCLIB member that is started by IMS Queue Control Facility extensions to start the IMS SCI address space.
>
> This PROCLIB member contains the JCL for the IMS SCI job.
>
> A valid name can consist of uppercase alphabetic characters (A through Z), special characters ($, #, or @), and numeric values (0 through 9). The first character of the name must be an alpha character.

WTODESTINATION control statement

The WTODESTINATION statement specifies the destination for all queue overflow protection messages.

Syntax

```
                 ┌─ WTODESTINATION=(WTO) ─┐
▶▶───────────────┼───────────────────────┼───────▶◀
                 └─ WTODESTINATION=(MTO) ─┘
```

Keywords

WTO
> IMS Queue Control Facility issues only WTO for all queue overflow protection messages.
>
> WTO is the default value if the WTODESTINATION statement is not specified.

MTO
> IMS Queue Control Facility issues WTO for all queue overflow protection messages and additionally routes these messages to the IMS control region.
>
> **Note:** If WTODESTINATION(MTO) is specified in the extensions PROCLIB member, the total queue usage increases at a much faster rate because of the duplication of overflow protection IQC messages.

Extensions ONTHRESHOLD*nn* PROCLIB member reference

This topic describes the syntax and parameters for the IMS Queue Control Facility extensions ONTHRESHOLD*nn* PROCLIB member.

Note: For an explanation of the PROCLIB member format, usage, and configuration, see "7. Enabling and configuring the IMS Queue Control Facility extensions" on page 31.

This PROCLIB member must be available to the IMS Queue Control Facility server. The server uses this member to initiate the specified function.

The following diagram shows the syntax for the IMS Queue Control Facility extensions ONTHRESHOLD*nn* PROCLIB member. This member must also be present in the server startup JCL.

```
►►─ IMS(imsid ) ── QCFIN(DSN( user_proclib ),MEM( member_name )) ── QCFPRINT( user_data_set ) ─►

            ┌────────── LOAD( user_data_set ) ──────────┐►◄
    ►───────┼─────── SCRAPLOG( user_data_set ) ────────┤
            ├──────── RECOVER( user_data_set ) ─────────┤
            ├──────── BROWSE( user_data_set ) ──────────┤
            ├──────── UNLOAD( user_data_set ) ──────────┤
            └──────── QUERY( user_data_set ) ───────────┘
```
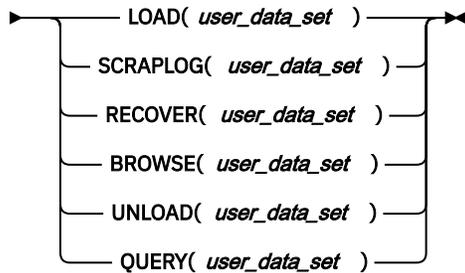
The following statements can be coded in the ONTHRESHOLD*nn* PROCLIB member.

IMS(*imsid*)
    This parameter is ignored. *imsid* identifies the IMS ID.

QCFIN(DSN(*user_proclib*),MEM(*member_name*))
    The QCFIN parameter is required for all IMS Queue Control Facility extensions ONTHRESHOLD*nn* PROCLIB members.

   user_proclib
        This value identifies the user PROCLIB that contains the QCFIN member name (*member_name*).

   member_name
        This value identifies the member name that contains the input commands for the step that is being processed.

QCFPRINT(*user_data_set*)
    The QCFPRINT parameter is required for all IMS Queue Control Facility extensions ONTHRESHOLD*nn* PROCLIB members. The *user_data_set* value identifies the data set that is used for print output.

LOAD(*user_data_set*)
    The LOAD parameter is required for all IMS Queue Control Facility extensions ONTHRESHOLD*nn* PROCLIB member LOAD steps. The *user_data_set* value identifies the data set that contains the source of the LOAD input data.

SCRAPLOG(*user_data_set*)
    The SCRAPLOG parameter is required for all IMS Queue Control Facility extensions ONTHRESHOLD*nn* PROCLIB member LOAD steps. The *user_data_set* value identifies the output data set that contains the rejected messages from the LOAD steps.

RECOVER(*user_data_set*)
    The RECOVER parameter is required for all IMS Queue Control Facility extensions ONTHRESHOLD*nn* PROCLIB member RECOVER steps. The *user_data_set* value identifies the data set that contains the RECOVERY input data.

BROWSE(*user_data_set*)
    The BROWSE parameter is required for all IMS Queue Control Facility extensions ONTHRESHOLD*nn* PROCLIB member BROWSE steps. The *user_data_set* value identifies the data set that is used for the BROWSE function output.

UNLOAD(*user_data_set*)
    The UNLOAD parameter is required for all IMS Queue Control Facility extensions ONTHRESHOLD*nn* PROCLIB member UNLOAD steps. The *user_data_set* value identifies the output data set that contains the UNLOAD function data.

QUERY(*user_data_set*)
    The QUERY parameter is required for all IMS Queue Control Facility extensions ONTHRESHOLD*nn* PROCLIB member QUERY steps. The *user_data_set* value identifies the data set that is used for the QUERY function output.

Chapter 6. Installation verification procedure

The installation verification procedure ensures that IMS Queue Control Facility is installed and configured correctly.

Procedure

Perform the following steps to verify the IMS Queue Control Facility installation:

1. Customize and submit the sample IMS Queue Control Facility batch job: *hlq*.SIQCSAMP(IQCBAQRY)

   - Customize the sample JCL according to the comments in the sample
   - Submit the JCL
   - Check the output from the job and validate that the client process returned the following information:

     – A return code of 0
     – Information in the QCFPRINT DD

2. Customize the IQC REXX member: *hlq*.SIQCEXEC(IQC31)

   - Ensure that you are pointing to the correct data sets
   - Ensure that the HLQ variable contains the target authorized data set high-level qualifier

3. Customize and submit the sample server JCL: *hlq*.SIQCSAMP(IQCSSERV)

   - Check the active running job for a message indicating that the IMS Queue Control Facility server is ready
   - Enter the following TSO command:

     ```
     EX HLQ.SIQCEXEC(IQC31)
     ```

   - Press Enter on the initial IMS Queue Control Facility panel
   - Select the target server from the server selection panel. If no server exists in the list to select, ensure that the server has been installed correctly and is running
   - Select **1** (Status) from the Main Menu panel
   - Select **1** (Environment) from the Status Menu panel

Results

The Environment panel is displayed.

Check that IMS Queue Control Facility extensions are enabled. Look for message IQC6001I in the system log:

```
QCF EXTENSION ENABLED
```

Chapter 7. Migration, fallback, and coexistence

You can use IMS Queue Control Facility functions as migration or fallback aids to requeue messages across supported IMS releases.

Messages that are created on one release of IMS can be inserted into another release of IMS, if the source and destination resources (LTERMs, transactions, MSC names, for example) are defined on both systems.

All supported releases of IMS are supported by IMS Queue Control Facility.

Migrating (nonshared queues)

To requeue messages for migration, perform the following tasks:

- Stop the old system with a DUMPQ.
- Start the new system.
- Requeue messages by running the RECOVERDM function on the SLDS created by the DUMPQ.

If the new system is in a shared-queues environment, IMS Queue Control Facility places the messages in the proper (shared or local) queue.

Migrating (shared queues)

You can requeue messages for migration with IMS Queue Control Facility.

To requeue messages for migration, perform the following tasks:

- Unload all messages from the shared queues by running the UNLOAD function.
- Stop the old system.
- Start the new system.
- Requeue messages by running the LOAD function.

Performing fallback (nonshared queues)

You can perform fallback in the nonshared-queues environment.

To requeue messages for fallback, perform the following tasks:

- Start the old IMS system (after failure of the new IMS system).
- Run the RECOVERAB function to requeue messages in the nonshared-queues environment.

Performing fallback (shared queues)

You can perform fallback in the shared-queues environment.

To requeue messages for fallback, perform the following tasks:

- If IMS is stopped, start the new release of IMS.
- Unload all messages from the shared queues by running the UNLOAD function.
- Start the old release of the IMS system.
- Run the LOAD function on the data set that was created by the UNLOAD function. In this case, the LOAD function is run as a stand-alone procedure.

Old threshold job names

If you are not building new threshold jobs or procedures to unload the message queues, and you plan to use the threshold jobs that are distributed with IMS Queue Control Facility to do the unload, then

you must define these names in the IMS Queue Control Facility configuration file definition on the ONTHRESHOLD*xx* statement.

QCFIN function statement SBMPAB / BC / CD / UT

SBMP*xx* on the QCFIN FUNCTION statement is a parameter for which you must supply a valid setting.

You must set the value of SBMP*xx* to either AB, BC, CD, or UT to define the threshold for which this job is to be scheduled.

If you omit this new parameter, the threshold job is scheduled only once. This parameter identifies the threshold for which the job is executing. Define this new parameter for the QCFIN parameters that are being supplied to threshold jobs AB, BC, CD, and UT.

Generation data sets

When you are deciding what to do for the threshold jobs such as unloads of the message queues, you might want to consider using Generation Data Set Groups (GDG)s .

Use of GDGs will help to prevent the unloaded threshold data from being overwritten if the same threshold is crossed again before the unloaded data set from the first crossing has finished processing.

Coexistence

If you run the existing partner product user exit routines, you might need to redesign and code them in order to run them with this version of IMS Queue Control Facility.

Ensure that the IMS Queue Control Facility IMS extensions are installed and running with all IMS subsystems that have the ability to manipulate the IMS message queues.

Chapter 8. Installing and configuring the Syntax Checker

IMS Queue Control Facility Syntax Checker performs syntax checking of either an IMS Queue Control Facility server configuration or an IMS Queue Control Facility extension configuration file.

Topics:

- "Configuring and submitting the Syntax Checker" on page 79
- "Additional validation by the Syntax Checker" on page 81

Configuring and submitting the Syntax Checker

This topic shows a sample of the JCL that is required to start the Syntax Checker.

The IMS Queue Control Facility configuration file syntax checker communicates with IMS Base Primitive Environment (BPE) services to perform a portion of this function.

The IMS Queue Control Facility Syntax Checker uses the BPE parsing service to parse the configuration data and then does additional validation. The Syntax Checker parsing stops parsing on the first syntax error it encounters. A BPE error message is issued showing the line number and character number where the parsing error occurred.

The Syntax Checker also performs additional validation.

Before you start the IMS Queue Control Facility server address space or before starting the IMS environment, the IMS Queue Control Facility server and IMS Queue Control Facility extension configuration files which are stored as PROCLIB members can be syntax validated to remove the possibility of having to recycle either the IMS Queue Control Facility server or the IMS environment.

Sample JCL to start the Syntax Checker

The *hlqual.siqcsamp*(IQCSYNTX) member contains the following sample procedure. Replace the items that appear in lowercase *italics* with values that match your environment's specifications.

```
//*******************************************************************/
//* JOB NAME = IQCSYNTX                                             */
//*                                                                 */
//* DESCRIPTIVE NAME = SAMPLE PROCEDURE TO START                    */
//*                    QCF SYNTAX CHECKER                           */
//*                                                                 */
//* STATUS = VERSION 03  RELEASE 02  MODIFICATION LEVEL 00          */
//*                                                                 */
//* FUNCTION = RUN QCF SYNTAX CHECKER                               */
//*******************************************************************/
//*                                                                 */
//*     LICENSED MATERIALS - PROPERTY OF IBM                        */
//*                                                                 */
//*     IMS QCF FOR Z/OS VERSION 3 RELEASE 2                        */
//*                                                                 */
//*     5697-N50                                                    */
//*                                                                 */
//*     COPYRIGHT IBM CORP. 2004, 2014                              */
//*                                                                 */
//*     ALL RIGHTS RESERVED.                                        */
//*     US GOVERNMENT USERS RESTRICTED RIGHTS -                     */
//*     USE, DUPLICATION OR DISCLOSURE RESTRICTED                   */
//*     BY GSA ADP SCHEDULE CONTRACT WITH IBM CORP.                 */
//*                                                                 */
//*******************************************************************/
//* NOTES =                                                         */
//* 1) REVIEW ALL STATEMENTS BEFORE SUBMITTING THIS JOB.            */
//*                                                                 */
//* 2) CHANGE IQCTYPE=type TO EITHER:                      */
//*     E - FOR QCF EXTENSION PROCLIB MEMBER VALIDATION, OR         */
//*     S - FOR QCF SERVER PROCLIB MEMBER VALIDATION.               */
```

```
//*                                                               */
//* 3) CHANGE IQCCFG=cfg_name TO THE QCF EXTENXION OR QCF  */
//*    SERVER PROCLIB MEMBER THAT IS TO BE VALIDATED.           */
//*                                                               */
//* 4) CHANGE hlqual FIELDS TO VALID HIGH LEVEL QUALIFIERS  */
//*    THAT COMPLY WITH YOUR SITES NAMING STANDARDS.            */
//*    CHANGE imshlqual FIELDS TO VALID HIGH LEVEL QUALIFIERS */
//*    FOR IMS LOAD LIBRARY.                                    */
//*    CHANGE qcfhlqual FIELDS TO VALID HIGH LEVEL QUALIFIERS  */
//*    FOR QCF LOAD LIBRARY.                                    */
//*                                                               */
//*  APAR     DATE     ID    DESCRIPTION                        */
//*                                                               */
//**********************************************************************/
//IQCSYNTX  PROC
//**********************************************************************
//*  BRING UP QCF SYNTAX CHECKER                                       *
//**********************************************************************
//STEP1 EXEC PGM=IQCINI0$,
//      PARM='BPEINIT=IQCSINI0,IQCTYPE=type,IQCCFG=cfg_name'
//STEPLIB DD DISP=SHR,DSN=qcfhlqual.SIQCLINK
//        DD DISP=SHR,DSN=imshlqal.RESLIB
//PROCLIB DD DISP=SHR,DSN=qcfhlqual.PROCLIB
//SYSPRINT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
```

Parameter reference for Syntax Checker

Code the following parameters on the Syntax Checker procedure EXEC PARM= statement:

IQCTYPE=*type*
Specify a 1-character type to denote whether the configuration file that is to be checked by the Syntax Checker is either for the IMS Queue Control Facility server or for the IMS Queue Control Facility extension. If more than one character is specified, only the first character following IQCTYPE= is validated and any trailing characters are ignored.

type can be E or S, where:

- E indicates that an IMS Queue Control Facility extension configuration file is to be checked by the Syntax Checker.
- S indicates that an IMS Queue Control Facility server configuration file is to be checked by the Syntax Checker.

If you do not specify a 1-character name, the Syntax Checker will generate error message IQC6611E, stating that the QCFTYPE= parameter was not specified.

If you specify an invalid type, the Syntax Checker will generate error message IQC6612E, stating that the QCFTYPE= parameter is invalid.

If you specify IQCTYPE=E and IQCCFG=*a_server_proclib_mbr* or IQCTYPE=S and IQCCFG=*an_extension_proclib_mbr*, parsing errors will occur.

IQCCFG=*cfg_name*
Specify an 8-character name for your IMS Queue Control Facility server or IMS Queue Control Facility extension configuration PROCLIB member.

If you did not specify a PROCLIB member, the Syntax Checker will generate error message IQC6610E, stating that the QCF configuration PROCLIB was not specified.

If you specify a PROCLIB member that does not exist, the Syntax Checker will generate error message IQC6614E, stating that the specified IMS Queue Control Facility configuration PROCLIB could not be found.

If both the IQCCFG= and IQCTYPE= parameters are omitted, the Syntax Checker will generate error message IQC6613E. stating that both parameters were not specified.

The *hlqual.siqcsamp* (IQCSERVI) member contains a sample IMS Queue Control Facility server configuration and an IMS Queue Control Facility extension configuration. This parameter is required.

If you do not specify an 8-character name for your IMS Queue Control Facility server or IMS Queue Control Facility extension configuration PROCLIB member, the Syntax Checker will generate an error message.

DD statement reference for Syntax Checker

The DD statements and their specifications are described as follows:

PROCLIB DD
Specify this required parameter to indicate which of one or more partitioned data sets contain your IMS Queue Control Facility server and IMS Queue Control Facility extension configuration PROCLIB members.

Specify your IMS Queue Control Facility server configuration PROCLIB member or IMS Queue Control Facility extension configuration PROCLIB member as IQCCFG= parameter of the EXEC parameters.

STEPLIB DD
Specify this required set of authorized libraries that are contained in the Syntax Checker code to run.

Additional validation by the Syntax Checker

The Syntax Checker will perform additional validation to determine if the IMS Queue Control Facility server configuration file or if the IMS Queue Control Facility extension configuration file is to be validated based on the IQCTYPE=*type* and to determine which IMS Queue Control Facility configuration file to validate based on the IQCCFG=*cfg_name* parameters specified on the EXEC statement.

The Syntax Checker will validate the EXEC statement parameters of IQCTYPE=*type* and IQCCFG=*cfg_name*.

If the Syntax Checker determines that either parameter is either in error or missing, an error message will be issued, as well as error message IQC6621E, stating that the Syntax Checker has terminated with an EXEC statement PARM error.

Detailed descriptions about the error messages that are issued as a result of invalid or missing parameters follow.

- If the IQCQCF parameter is missing, error message IQC6610E will be issued.
- If the IQCTYPE parameter is missing, error message IQC6611E will be issued.
- If the IQCTYPE parameter specifies an invalid type (not E or S), error message IQC6612E will be issued.
- If both parameters are missing, error message IQC6613E will be issued.
- If the IQCQCF parm is specified but the PROCLIB name is not found, error message IQC6614E will be issued.

The Syntax Checker will first parse the configuration file and then perform additional validation of the config file parameters and provide additional information about the validation. Additional information about the validation follows.

- The PERCENT (TOTAL= USED=) keywords on the AREA and FAILSAFE statement will allow you to specify 0.

  The Syntax Checker will issue error message IQC6620I to inform you that you might wish to verify that selection even though 0 is valid.

  If no other errors are found in the validation, the job will terminate with successful completion message IQC6617I.

- The PERCENT (TOTAL= USED=) keywords on the AREA and FAILSAFE statement will do additional validation of excess of 99.

  The Syntax Checker will issue error message IQC6619E to inform you that your specification was in excess of 99.

  The job will terminate with error completion message IQC6205E.

- The AREA and FAILSAFE statements will allow you to specify any 1- to 8-character name for each statement, even if names are duplicated.

  The Syntax Checker will notify you of all duplicate names (ID=) within the AREA statements or duplicate names (ID=) within the FAILSAFE statements.

  The job will terminate with error completion message IQC6205E.

  The configuration file with duplicate names is valid and will not cause an execution time error.

  Possible scenarios for duplicate names and the Syntax Checker's response to them follow.

  – If you specify the same name (ID=) for multiple AREA statements, the Syntax Checker will notify you of all the duplications.

  – If you specify the same name (ID=) for multiple FAILSAFE statements, the Syntax Checker will notify you of all the duplications.

  – If you specify the same name for both AREA and FAILSAFE, Syntax Checker will notify you of duplicate AREA/FAILSAFE.

- The IMSPLEX Statement will allow you to specify a blank name, and the area statements will still function; however, IMS Queue Control Facility Extensions will not be able to communicate with the server.

  The Syntax Checker will issue error message IQC6615E, notifying you that the SCE Registration will fail.

  The job will terminate with error completion message IQC6205E.

Part 3. Using IMS Queue Control Facility

IMS Queue Control Facility functions are run using the ISPF panels or by issuing JCL job streams. Several server commands are specified with an MVS **MODIFY** command.

Most functions have control statement keywords with names similar to that of the function, and JCL steps. Functions, control statements, JCL, and subfunctions are described in the following topics.

Control statements, JCL, and subfunctions that are common to multiple functions are described first.

Some exit routines are replaceable because IMS Queue Control Facility user-exit routines are link-edited into the appropriate load modules. This is different from IMS, which loads its user exit routines.

Topics:

Chapter 9. Message queue overflow protection

IMS Queue Control Facility queue overflow protection is available for messages after the primary queue reaches a threshold in a nonshared-queues environment.

Topics:

Message queue overflow protection overview

IMS Queue Control Facility queue overflow protection is available for messages sent after the primary queue reaches a threshold in a nonshared-queues environment.

About queue overflow

IMS message queues have a limit on the number of messages that they can contain. The IMS Queue Manager can detect when this limit is reached and will respond to this limit by shutting down IMS (UABEND 758). This condition is known as *queue overflow*.

The prevention of an IMS shutdown due to a queue overflow condition is a vital issue in a production environment. IMS Queue Control Facility has the capability to prevent a queue overflow condition.

IMS Queue Control Facility and queue overflow protection

IMS Queue Control Facility implements queue overflow protection by monitoring the message queue usage. IMS Queue Control Facility analyzes the message queue usage to determine when to send an alert or to take action to prevent the excessive queue usage from continuing.

You can configure queue overflow protection by specifying various parameters to specify partition values, set queue usage limits, and specify appropriate alerts and actions.

In this discussion, the term queue overflow protection is equivalent to the term queue space usage notification (QSUN).

Queue overflow protection methods

IMS Queue Control Facility provides three methods for implementing queue overflow protection. They are listed in the order of their development through progressive releases of IMS Queue Control Facility. Each subsequent method was designed to provide more capability and flexibility in configuring queue overflow protection.

- **Type 1: Queue space notification exit (DFSQSPC0)**

  Type 1 protection is the original mechanism for providing basic overflow protection and has the most limited capabilities.

  The Queue Manager DFSQSPC0 user exit provides analysis to detect excessive queue activity for a specific IMS entity.

Actions that can be implemented when these partition boundaries are exceeded include notifications messages.

- **Type 2: Queue overflow protection using threshold settings to define queue space partitions**

  Type 2 protection provides more advanced overflow protection for IMS Queue Control Facility than type 1 protection.

  Type 2 protection provides analysis to detect excessive queue activity for total and specific IMS entities.

  Four analysis boundaries are defined using the IMS threshold template.

  Actions that can be implemented when these partition boundaries are exceeded include WAIT, WTO, STOP, and UNLOAD.

  Type 2 overflow protection using threshold partitions is the default protection method for a newly installed version of IMS Queue Control Facility.

- **Type 3: Queue overflow protection using area and failsafe settings to define queue space partitions**

  Type 3 protection provides the most advanced overflow protection for IMS Queue Control Facility.

  Type 3 protection provides analysis to detect excessive queue activity for total and specific IMS entities.

  Area protection analyzes specific queue usage. Analysis is performed on the accumulation of the queue activity due to a specific IMS process.

  Failsafe protection analyzes total queue utilization. Analysis is performed on the accumulation of the queue activity due to all IMS processes.

  A possible total of twenty analysis boundaries can be defined using AREA and FAILSAFE configuration statements.

  Area and failsafe protection can be used concurrently.

  Actions that can be implemented when these partition boundaries are exceeded include WAIT, WTO, STOP, and ABEND.

  Type 3 overflow protection is implemented by entering AREA and FAILSAFE configuration settings in the IMS Queue Control Facility extensions PROCLIB member.

  Type 2 overflow protection using threshold partitions is the default protection method for a newly installed version of IMS Queue Control Facility because there are no default type 3 AREA and FAILSAFE configuration settings specified in the extensions member.

Queue space partition methods

Implementing IMS Queue Control Facility queue overflow protection requires that the IMS queue space be logically divided into partitions.

These logical partitions allows IMS Queue Control Facility to monitor varying degrees of IMS queue usage. Partitions allow IMS Queue Control Facility to analyze the total queue usage, and identify the cause or source of the excessive queue usage. Partitions are specified as a percentage of the total queue space.

There are two methods available for defining the format of these partitions:

- IMS Queue Control Facility allows you to define partitions using threshold settings

  This partition method is used for type 2 queue overflow protection.

  You can configure threshold settings that divide the queue space into four logical partitions.

- IMS Queue Control Facility allows you to define partitions using area and failsafe settings

  This partition method is used for type 3 queue overflow protection.

  You can configure area and failsafe settings that divide the queue space into a maximum of ten logical partitions (or areas) for each area and failsafe group (for a total of 20 partitions).

Defining logical partitions using threshold settings

IMS Queue Control Facility allows you to define partitions using threshold settings. This partition method is used for type 2 queue overflow protection.

A threshold is a logical partition boundary that is expressed as a percentage of the total queue space. Percentage values can range from 1 - 99.

In an IMS system, the queue space is, by default, divided into four partitions using IMS runtime parameters QTL (lower percentage) and QTU (upper percentage). The partitions are formed from the following threshold specifications:

- A lower percentage of the total queue space (x)

  The default QTL value is 60%.
- A upper percentage of the total queue space (y)

  The default QTU value is 75%.
- The average of the lower and upper percents ((x + y) / 2)

IMS Queue Control Facility type 2 overflow protection allows you to override these default values.

An action can be triggered when a threshold boundary is crossed. Threshold actions consist of wait (WAIT) and stop (ABEND).

The notification threshold generates an alert (WTO) when queue utilization has reached a specified percentage. The default value for this threshold is 85%.

Scenario: Defining partitions using threshold settings

In this scenario, you are experiencing message queue overflows and message loss because some of your message queues are too small and they are filling up.

You can put controls in place to monitor message queue utilization and to notify you or take an action to prevent imminent overflow and loss of messages in your nonshared-queues environment.

To address your system and message queue problems, you decide that you need to change the QTL (lower threshold value) and QTU (upper threshold value) parameters in the IQCQSNUN table so that your four message queue partitions are larger and equal in size.

The three message queue data sets are SMSG, LMSG, and QBLK.

You divide the space that is available in each of these data sets into four partitions, which are based on the maximum number of records that are available in the data set and the percentages that you specify for QTL and QTU.

The following figure shows four IMS queue space partitions defined by threshold settings:

IMS queue space

| Partition A (Area A) | Partition B (Area B) | Partition C (Area C) | Partition D (Area D) |

0   t1   t2   t3   t5   t4
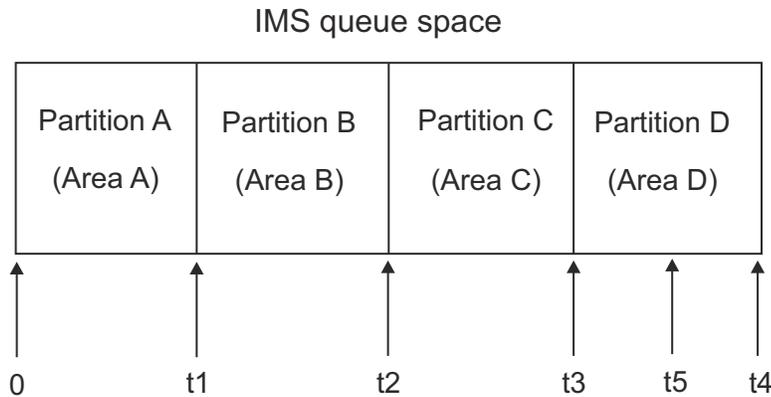
Figure 4. IMS queue space partitioned by threshold settings

The threshold calculations that you can use to define the four partitions are:

- Threshold 4 (t4) = maximum number of records available in the data set
- Threshold 1 (t1) = QTL * t4 (default QTL = 60%)
- Threshold 3 (t3) = QTU * t4 (default QTU = 75%)
- Threshold 2 (t2) = midpoint between t1 and t3 (60 + 75 = 135/2 = 67.5%)
- Threshold 5 (t5) = the boundary defined by IQC6101I notification message (85%)

These calculations would result in the following partition sizes:

- Partition A (Area A): 0 records through Threshold 1
- Partition B (Area B): Threshold 1 + 1 record through Threshold 2
- Partition C (Area C): Threshold 2 + 1 record through Threshold 3
- Partition D (Area D): Threshold 3 + 1 record through Threshold 4

Defining logical partitions using area and failsafe settings

IMS Queue Control Facility allows you to define partitions using area and failsafe settings. This partition method is used for type 3 queue overflow protection.

A partition is a boundary that is expressed as a percentage of the total queue space. Percentage values can range from 1 - 99.

You can define logical partition boundaries using area and failsafe settings. The number of specified partitions can be from 1 - 10.

The specification of the partitions is performed separately for each area and failsafe, resulting in a possible total of 20 partitions.

Area logic:

- Queue overflow protection defined by area settings monitors the utilization of the queue space by individual IMS processes.
- You can define up to ten partitions created by ten area settings that partition the total queue space.
- Each area setting can be configured to generate specific actions when triggered because its boundary was crossed.

Failsafe logic:

- Queue overflow protection defined by failsafe settings monitors the total utilization of the queue space by the cumulative effect of multiple processes.

- You can define up to ten partitions created by ten failsafe settings that partition the total queue space.
- Each failsafe setting can be configured to generate specific actions when triggered because its boundary was crossed.

Example: Defining partitions using area and failsafe settings

The existence of AREA and FAILSAFE statements in the IMS Queue Control Facility extensions PROCLIB member enables type 3 overflow protection. Type 2 protection (the product default) is implemented only if there are no type 3 AREA and FAILSAFE configuration settings specified in the extensions member.

This example shows several simple area and failsafe partition settings and their associated control statements that would be entered in the IMS Queue Control Facility extensions PROCLIB member.

Area partitions track the accumulation of queue space usage due to a specific IMS process. An individual process is defined as message queue inserts to and from one source or destination.

Failsafe partitions track the accumulation of queue space usage due to multiple IMS processes. Multiple processes are defined as total message queue inserts to and from multiple sources and destinations.

The following figure shows five IMS queue space partitions defined by area and failsafe settings:
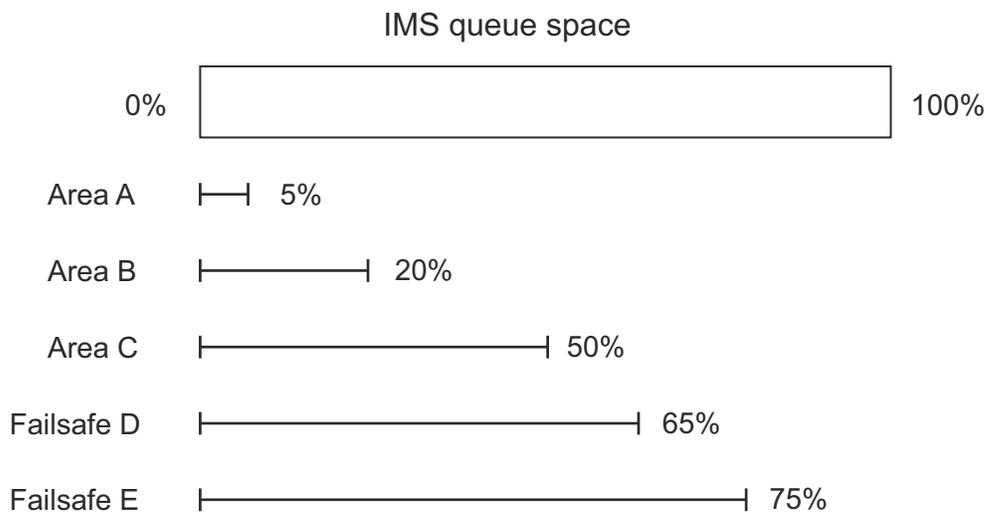
Figure 5. IMS queue space partitioned by area and failsafe settings

Note: This example and its scenarios are highly simplified. The settings illustrated here might be unrealistic in a production environment.

Example area actions:

Notification messages indicate the amount of queue space used by the triggering process.

- Area A, set for 5%, responds with a notification when exceeded by an individual process.

  Example control statement:

  ```
  AREA=(ID=AREA0001,
        PERCENT=(TOTAL=05,USED=99),
        TYPE=(ALL),
        CSTOPACTION=WTO,
        CSTARTACTION=WTO,
        OSTOPACTION=WTO,
        OSTARTACTION=WTO)
  ```

- Area B, set for 20%, responds with a notification when exceeded by an individual process.

Example control statement fragment:

```
PERCENT=(TOTAL=20,USED=99),
```

- Area C, set for 50%, responds with a notification when exceeded by an individual process.

    Example control statement fragment:

```
PERCENT=(TOTAL=50,USED=99),
```

Example failsafe actions:

- Failsafe D, set for 65%, responds with a notification when exceeded by the cumulative affect of multiple processes.

    Example control statement:

```
FAILSAFE=(ID=FAIL0004,
          PERCENT=(TOTAL=65,USED=99),
          ACTION=WTO)
```

- Failsafe E, set for 75%, responds with a notification when exceeded by the cumulative affect of multiple processes.

    Example control statement:

```
FAILSAFE=(ID=FAIL0005,
          PERCENT=(TOTAL=75,USED=99),
          ACTION=WTO)
```

Scenarios:

- Area settings can capture individual processes that exceed an area boundary.
- However, area settings cannot capture the cumulative effect of several processes that individually perform within boundary settings, but together exceed the queue space.
- A process running at 6% triggers a warning action from Area A.
- A process running at 21% triggers a severe action from Area B.
- A process running at 51% triggers a critical action from Area C.
- Seventeen processes, each running at 4% (total of 68%), will not trigger any Area actions, but will trigger a severe action from Failsafe D.
- Sixteen processes, each running at 4% (total of 64%), will not trigger any Area or Failsafe actions.

    However, if a process running at 35% is introduced (total of 99%), the new process (35%) triggers a severe action is from Area B, and the cumulative effect of all processes (99%) triggers a critical action from Failsafe E.

Configuring type 1 queue overflow protection

Type 1 queue overflow protection makes use of an IMS user exit routine, called the queue space notification exit (module DFSQSPC0).

Configuration summary for type 1 overflow protection

- You must add parameter PROTECTIONTYPE(UEXIT) to the control region start-up member IQC*imsid*I.
- The parameter causes member DFSQSCP0 to take over queue overflow protection.

Notes on using type 1 overflow protection

The queue space notification exit (DFSQSPC0) includes the following attributes:

- DFSQSPC0 is an IMS user exit routine.

- This exit is driven by Queue Manager when it detects that the total queue count has exceeded a partition boundary.
- The user exit actions consist of requesting Queue Manager to issue various notification messages or to stop a dependent region.

Reading threshold usage information

The DFSQSPC0 exit routine can return meaningful threshold usage information to IMS in register 15 or in the QSPCFLG2 flag byte.

Register 15

In register 15, DFSQSPC0 can return zero if no thresholds have been passed. If a threshold has been passed, one of the following IMS message keys can be returned:

2013
Creates message DFS2013 that reports the number of queue block data sets that have exceeded the upper threshold

2014
Creates message DFS2014 that reports the number of short message queue data sets that have exceeded the upper threshold

2015
Creates message DFS2015 that reports the number of large message queue data sets that have exceeded the upper threshold

2016
Creates message DFS2016 that reports the number of queue block data sets that have passed the lower threshold

2017
Creates message DFS2017 that reports the number of short message queue data sets that have passed the lower threshold

2018
Creates message DFS2018 that reports the number of large message queue data sets that have passed the lower threshold

Requirement: If you have replaced DFSQSPC0 with a user exit routine, the user exit routine's message number must be returned in register 15 as the negative of the positive message number (LNR). For example, if the user exit routine's message number is 9999, the message number that is returned in register 15 must be -9999.

Flag QSPCF2NO

DFSQSPC0 can set flag QSPCF2NO (in byte QSPCFLG2, mapped by DFSPARM) to indicate that the unit-of-work exceeded the allowed buffer count.

The requested action for each ITASK type is as follows:

- regions stop = A7 status code
- VTAM = DFS074 message
- APPC = DFS0777 message
- OTMA = NAK message
- BTAM = DFS074 message
- MSC = DFS1945 message

If DFSQSPC0 sets the QSPCF2NO flag, the IMS Queue Control Facility Overflow Protection exit logic is not called for this unit-of-work.

Configuring type 2 queue overflow protection (thresholds)

Type 2 overflow protection provides more advanced overflow protection for IMS Queue Control Facility than type 1 protection.

Topics:

Type 2 queue overflow protection configuration overview

Initial threshold values for IMS Queue Control Facility type 2 queue overflow protection are obtained as "hard-coded" values from IMS.

You can modify these "hard-coded" values through the IMS Queue Control Facility TSO client. The Queue Space Notification (QSN) table contained in member IQCQSNUN captures and stores any threshold values modified through the TSO client.

Threshold values for type 2 overflow protection that are modified through the TSO client are valid only for the current IMS session. An IMS restart causes IMS Queue Control Facility to revert to the default "hard-coded" system values.

You can preserve these settings across multiple IMS sessions by starting a procedure after an IMS restart that restores the values to IMS Queue Control Facility.

Configuration summary for type 2 overflow protection

Use the IMS Queue Control Facility TSO client to configure the following for type 2 overflow protection attributes:

- **Threshold percentage values that define the queue space partitions**

  IMS startup parameters QTL and QTU are used initially to define four queue space partitions for committed and uncommitted messages.

  You can change threshold values through the TSO client Queue Threshold Parameters panel. However, these modified values are valid only for the current IMS session. An IMS restart causes IMS Queue Control Facility to revert to the default hardcoded system values.

  Additionally, you can preserve your custom threshold values over multiple IMS sessions by starting a procedure after any IMS restart that restores these custom values from member IQCQSNUN.

- **Percentage of the queue space that returns a notification message IQC6101**

  The hardcoded IMS value is used initially.

  You can change percentage values through the TSO client Queue Threshold Parameters panel. However, these modified values are valid only for the current IMS session. An IMS restart causes IMS Queue Control Facility to revert to the default hardcoded system values.

  Additionally, you can preserve your custom values over multiple IMS sessions by starting a procedure after any IMS restart that restores these custom values from member IQCQSNUN.

- **Names of jobs and procedures that are started when a required action must be taken for committed messages**

  The hardcoded IMS values are used initially.

  You can change these values through the TSO client Queue Threshold Parameters panel. However, these modified values are valid only for the current IMS session. An IMS restart causes IMS Queue Control Facility to revert to the default hardcoded system values.

Additionally, you can preserve your custom values over multiple IMS sessions by starting a procedure after any IMS restart that restores these custom values from member IQCQSNUN.

- **Alerts and actions to be taken for uncommitted messages** (in-process but incomplete messages)

  The hardcoded IMS values are used initially.

  You can change these values through the TSO client QSN Table Maintenance panel or a batch job. However, the modified values are valid only for the current IMS session. An IMS restart causes IMS Queue Control Facility to revert to the default hardcoded system values.

  Additionally, you can preserve your custom values over multiple IMS sessions by starting a procedure after any IMS restart that restores these custom values from member IQCQSNUN.

- **Automatic queue unload for committed messages**

  Automatic unload of committed messages is a feature of type 2 and type 3 queue overflow protection that allows you to regain queue space by removing messages from the queue and relocating them. Use the automatic unload feature to save all messages or specific groups of messages.

  Values for automatic queue unload are configured using the ONTHRESHOLD*xx* statements in the IMS Queue Control Facility extensions PROCLIB member.

System configuration process flow

1. Initial threshold values used by IMS Queue Control Facility are hard-coded values obtained from IMS.
2. The IMS Queue Control Facility TSO client writes any threshold configuration changes back to the data set member IQCQSNUN.

   Member IQCQSNUN contains the Queue Space Notification (QSN) table that stores default and modified values for use by IMS Queue Control Facility.

   The TSO client then passes the appropriate changes to the IMS system, which recalculates the four queue space partitions.
3. The type 2 queue overflow protection logic returns the actions that you specified when the newly calculated queue partitions approach critical levels.
4. The type 2 mechanism avoids having to stop IMS when you change these threshold configuration values through IMS Queue Control Facility.
5. Modified configuration values are valid only for the current IMS session.

   An IMS restart causes IMS Queue Control Facility to revert to the initial hard-coded system values.
6. To preserve any modified values for type 2 overflow protection across multiple IMS sessions, you can start a procedure after any IMS restart that restores those values from member IQCQSNUN.

Sample QSN table containing type 2 overflow protection settings

Member IQCQSNUN contains the Queue Space Notification (QSN) table that stores default and custom values for use by IMS Queue Control Facility.

To preserve any modified values for type 2 overflow protection across multiple IMS sessions, you can submit a batch job after any IMS restart that restores those values from member IQCQSNUN.

```
function QC_LTBL
JOBSCARD IQCPRCABIQCPRCBCIQCPRCCDIQCPRCUT
AREAA    100N100N100N100N100N100N100N100N100N100N100N100N
AREAB    050W050W050W050W050W050W050W050W050W050W050W050W
AREAC    008W005W008W005W008W005W008W005W008W005W008W005W
AREAD    008S005S008S005S008S005S008S005S008S005S008S005S
PERCENTS 060075080Y
end
```

Threshold settings (type 2 overflow protection)

Use the IMS Queue Control Facility TSO client Queue Threshold Parameters panel or a batch job to override the default IMS runtime parameters QTL and QTU that define lower and upper threshold percentages for the IMS message queue.

Type 2 protection uses IMS threshold settings that divide the queue space into four logical partitions. These four partitions are created from IMS runtime parameters that specify a lower percentage threshold (QTL) and an upper percentage threshold (QTU). A third threshold is created that represents the sum of the two percentages divided by 2.

IMS Queue Control Facility allows you to modify the default IMS QTL and QTU values by using the IMS Queue Control Facility TSO client or a batch job.

The QTL and QTU parameters are expressed as a percentage of the maximum space that is available for use in the message queue data sets according to the following guidelines:

- QTL is the lower threshold value and defaults to 60%.
- QTU is the upper threshold value and defaults to 75%.

Specific actions, triggered by partition boundaries being exceeded, can also be configured.

The IMS Queue Control Facility TSO client panels display the initial threshold values set in the IMS Queue Control Facility data set member IQCQSNUN. A sample IQCQSNUN member can be found in the IMS Queue Control Facility sample library file SIQCSAMP.

Specifying queue use that issues message IQC6101

The notification threshold generates an alert (WTO) that warns you when total queue utilization has reached a specified percentage.

The default value for this threshold is 85%.

Alerts and actions for uncommitted messages (type 2 overflow protection)

Use the IMS Queue Control Facility TSO client QSN Table Maintenance panel or a batch job to change the default percentages of queue use that trigger actions on uncommitted messages.

Uncommitted messages are in-process messages that are not complete.

Each partition (A, B, C, and D) has twelve entries. These entries represent the process type (six of these entries) and the state of the destination (two of these entries). Therefore, each partition can have twelve entries (2 * 6 = 12).

- The process can be any of the following types:

    Type 0 - old interface (unknown process)
    Type 2 - APPL (applications) assigned DRRN
    Type 3 - APPC (LU 6.2 /APPC) assigned DRRN
    Type 4 - DC (Data communications input terminal, BTAM / VTAM)
    Type 5 - OTMA
    Type 6 - MSC

- The state of the destination can be either started or stopped.

For each entry, you can specify the percentage of use that causes an action result. Queue Manager provides a queue count for each of its processes. When the queue count exceeds the percentage of use that is defined for that partition, the specified action is taken.

The action can be any of the following types:

- **N** - No action.
- **W** - WAIT the user.

    You must use the TSO/ISPF QSN waited panel to end the wait.

- **S** - Stop action, which can be any of the following actions:
  - For APPL (applications), issue an A7 status code
  - For APPC (LU 6.2 /APPC), issue message DFS777
  - For DC (Data communications input terminal, BTAM / VTAM), issue message DFS074
  - For OTMA, issue a NAK message
  - For MSC, issue message DFS1945

Managing tasks in the WAIT state

When the task is placed in WAIT state, the task remains in the WAIT state until you take an appropriate action. A message is issued when the WAIT queue goes from empty to not empty. By using option **4 - Wait** on the QCF Main Menu panel, you can view those tasks that are in the WAIT state and take one of the following actions:

- For applications that are in a WAIT state, you can release the task to perform one of the following actions:
  - Continue processing
  - Abort processing
  - Issue an A7 status code
  - Terminate the application and put the input message on the suspend queue
- For other caller types that are in a WAIT state, you can release the task to perform one of the following actions:
  - Continue processing
  - Take one of these following actions for the following caller types:

Caller type	Action
APPC (LU 6.2 /APPC)	Issue message DFS777
DC (Data communications input terminal, BTAM / VTAM)	Issue message DFS074
OTMA	Issue a NAK message
MSC	Issue message DFS1945

Note: If you specify that processing should continue for a task (RELEASE on the ISPF panel), that task will not be placed into a WAIT state again, even if that task matches the specified criteria unless the usage is in area D.

Preserving modified threshold settings across IMS sessions

Initial threshold values for IMS Queue Control Facility type 2 queue overflow protection are obtained as "hard-coded" values from IMS.

You can modify these "hard-coded" values through the IMS Queue Control Facility TSO client. The Queue Space Notification (QSN) table contained in member IQCQSNUN captures and stores any threshold values modified through the TSO client.

Threshold values for type 2 overflow protection that are modified through the TSO client are valid only for the current IMS session. An IMS restart causes IMS Queue Control Facility to revert to the default "hard-coded" system values.

You can preserve these settings across multiple IMS sessions by starting a procedure after an IMS restart that restores the values to IMS Queue Control Facility.

JCL to restore modified type 2 overflow protection settings

The following code shows example JCL that reads the Queue Space Notification (QSN) table from member IQCQSNUN and restores the default and modified type 2 overflow protection values from the table to IMS Queue Control Facility.

In this example, the JCL job name is IQCLDTBL. You can use any name for the JCL. The IMSMSG procedure in the IMS Control Region DFSPB*xxx* member must be configured to run this IQCLDTBL procedure at IMS restart.

```
//*----------------------------------------------------------------------
//*      LICENSED MATERIALS - PROPERTY OF IBM
//*
//*      IMS QCF FOR Z/OS VERSION 3 RELEASE 2
//*
//*      5697-N50
//*
//*      COPYRIGHT IBM CORP. 2000, 2014
//*
//*      ALL RIGHTS RESERVED.
//*      US GOVERNMENT USERS RESTRICTED RIGHTS -
//*      USE, DUPLICATION OR DISCLOSURE RESTRICTED
//*      BY GSA ADP SCHEDULE CONTRACT WITH IBM CORP.
//*
//*----------------------------------------------------------------------
//*
//* NAME: IQCLDTBL
//* DESC: SAMPLE PROCEDURE - USED TO RESTORE THE CUSTOM TYPE 2
//*       OVERFLOW PROTECTION SETTINGS THAT ARE STORED IN IQCQSNUN
//*       MEMBER. REFER TO USER'S GUIDE FOR ADDITIONAL INFORMATION.
//*
//* FUNCTION: RESTORE THE CUSTOM TYPE 2 OVERFLOW PROTECTION SETTINGS
//*           FROM THE IQCQSNUN MEMBER AFTER AN IMS RESTART
//*
//*----------------------------------------------------------------------
//* NOTES =
//*
//* 1) REVIEW ALL STATEMENTS BEFORE STORING THE JCL
//* 2) &IMSID IS PARAMETER, PASSED ON START OF JCL
//*    &PLEXN IS PARAMETER, PASSED ON START OF JCL
//* 4) CHANGE IMSHLQ.RESLIB TO THE VALID IMS RESLIB NAME
//* 5) CHANGE HLQUAL.SIQCLINK TO THE VALID QCF LOADLIB NAME
//* 6) CHANGE THE HLQUAL FIELDS TO VALID HIGH LEVEL QUALIFIERS
//*    THAT COMPLY WITH YOUR SITES NAMING STANDARDS.
//* 7) CHANGE IMSID     TO THE IMSID OF CONTROL REGION
//*    CHANGE PLEXN     TO THE REAL PLEX NAME
//* 8) CHANGE THE SYSDA  FIELD TO A VALID LOCATION UNIT
//*    IDENTIFICATION THAT COMPLIES WITH YOUR SITES NAMING
//*    STANDARDS.
//*
//IQCLDTBL PROC RGN=0M,
//            SOUT=H,
//            PLEXN=plexn,
//            IMSID=imsid
//*
//*----------------------------------------------------------------------
//STEP1    EXEC PGM=IQCINI0$,REGION=&RGN,TIME=1440,
// PARM='BPEINIT=IQCBINI0,BPECFG=IQCBACNF,IMSPLEX=&PLEXN,QCFIMS=&IMSID'
//*
//STEPLIB  DD  DISP=SHR,DSN=hlqual.SIQCLINK
//         DD  DISP=SHR,DSN=imshlq.RESLIB
//*
//PROCLIB  DD  DISP=SHR,DSN=hlqual.PROCLIB
//*
//SNAPDUMP DD  SYSOUT=&SOUT
//SYSUDUMP DD  SYSOUT=&SOUT
//SYSPRINT DD  SYSOUT=&SOUT
//QCFPRINT DD  SYSOUT=&SOUT
//*
//QCFIN    DD  DISP=SHR,DSN=IQC.mvsid.imsid.UNCOMMIT(IQCQSNUN)
```

Configuring the IMSMSG procedure to run the IQCLDTBL procedure

To run the IQCLDTBL procedure after an IMS restart, you must specify the IMSRDR procedure in the IMS Control Region DFSPB*xxx* member using the PRDR parameter:

```
PRDR=IMSRDR
```

The IMSRDR procedure is used to read an IMSMSG job that initiates the IMS control program region and message regions.

By adding the following JCL into your IMSMSG procedure, the IQCLDTBL procedure is run at IMS start-up. (The IQCLDTBL procedure restores your modified type 2 overflow protection settings from member IQCQSNUN.)

```
//*********************************************************************
//* QCF 3.2
//* This job executes the procedure IQCLDTBL which is used to
//* restore custom type 2 overflow protection procedure settings.
//* Change imsid to the IMS ID of the Control Region.
//* Change hlqual to the valid high level
qualifier.
//*********************************************************************
//*
//IQCLDTBL JOB IMS,
// 'imsid',
// CLASS=A,
// MSGCLASS=H,MSGLEVEL=(1,1),
// REGION=0M
//*
/*JOBPARM SYSAFF=RSI3
// JCLLIB ORDER=(hlqual.PROCLIB)
//*
//STEP1    EXEC IQCLDTBL
//*
```

Configuring type 2 overflow protection using the TSO client

The IMS Queue Control Facility uses the initial configuration values set in the IMS Queue Control Facility data set member IQCQSNUN to implement type 2 overflow protection.

About this task

The following configuration notes apply to configuring type 2 overflow protection:

- The settings in the IMS Queue Control Facility extensions PROCLIB member determine whether type 2 or type 3 overflow protection is implemented.

  The existence of AREA and FAILSAFE statements in the IMS Queue Control Facility extensions PROCLIB member enables type 3 overflow protection.

  Type 2 protection (the product default) is implemented only if there are no type 3 AREA and FAILSAFE configuration settings specified in the extensions member.

- The absence of type 3 AREA and FAILSAFE settings in the IMS Queue Control Facility extensions PROCLIB member activates the appropriate type 2 menu option (**5**) on the QCF Main Menu.

  Use QCF Main Menu option **5 - Tables** to configure type 2 overflow protection.

  QCF Main Menu option **6 - Notify**, used for type 3 overflow protection, is not active.

- Configuration changes made with the IMS Queue Control Facility TSO client are valid only for the current IMS session. They are lost when the IMS is terminated.

  To preserve custom values for type 2 overflow protection across multiple IMS sessions, you can start a procedure after any IMS restart that restores those values from member IQCQSNUN.

- Before you begin the following procedure, ensure that you have selected the appropriate server and IMS ID that reads the appropriate IMS Queue Control Facility extensions PROCLIB member.

Procedure

Perform the following steps to configure type 2 queue overflow protection:

1. From the QCF Main menu, select option **5 - Tables**.

   The Queue Overflow Parameters Sub-Menu panel is displayed.

2. Select option **1 - Select queue space notification table**.

   The Queue Space Notification Tables panel is displayed.

3. Select option **1 - Edit queue space notification tables**.

   The QSN Table Names panel is displayed.

4. Type the E (Edit) row action in the row representing the current overflow parameters table.

   The Queue Threshold Parameters panel is displayed.

```
 Help

---------------------------------Queue Threshold parameters----------------------------
                              Queue Threshold parameters
Command ===>

Press ENTER to continue or END to exit.           APAR . . : PH19814 2020/01/
  Server . . : IQCSERV9                            JDTE . . : 2020.016
  IMS ID . . : IMSA                                TIME . . : 14:49:18
  Member . . : IQCQSNUN                            DATE . . : 2020/01/16

 Committed and uncommitted messages: Thresholds.
 Queue  upper  threshold  . . . 075  Queue  lower  threshold . . . . 060
 Queue IQC6101 threshold  . . . 085

 Committed messages: Job/Procedure names to be called at threshold crossover.
 Ignore application calls inserting messages to express PCBs . . . . Y

 Jobnames  - A-B:  . . . . NOUNLOAD  B-C: . . . NOUNLOAD  C-D: . . . NOUNLOAD
 Procnames - A-B:  . . . .           B-C: . . .           C-D: . . .

 Jobname   - IQC6101  . . NOUNLOAD
 Procname  - IQC6101  . .
```

Figure 6. Queue Threshold Parameters panel

5. Use this panel to modify the following parameters for committed and uncommitted messages:

   - Queue upper threshold (QTU)
   - Queue lower threshold (QTL)
   - IQC6101 threshold
   - JOB and procedure names for actions (including automatic unload of committed messages) that take place for committed messages when threshold boundaries are exceeded

6. Press Enter to continue to the configure settings for uncommitted messages.

   The QSN Table Maintenance panel is displayed.

```
 Help
 ------------------------------------QSN Table Maintenance------------------------------------
 Command ===>

 ENTER to continue, END to go to previous panel.    APAR . . : PH19814 2020/01/
   Server . . : IQCSERV9                            JDTE . . : 2020.016
   IMS ID . . : IMSA                                TIME . . : 14:51:49
   Member . . : IQCQSNUN                            DATE . . : 2020/01/16

  Uncommitted messages: Modify QSN entries using the AREAs "ALL" Callers TYPE.
 For each AREA status Started/Stopped enter a set of Percent/Action(N, S, W).

 A: Start . . 100 N  Stop . . 100 N     B: Start . . 050 W  Stop . . 050 W
 C: Start . . 008 W  Stop . . 005 W     D: Start . . 008 S  Stop . . 005 S

  Uncommitted messages: Modify individual entries in the QSN table.
 For each Caller status Started/Stopped enter a set of Percent/Action(N, S, W).

        ___TYP0____  ___APPL____  ___APPC____  ____DC_____  ___OTMA____  ___MSC_____
 AREA Start  Stop Start  Stop Start  Stop Start  Stop Start  Stop Start  Stop
 A    100 N 100 N 100 N 100 N 100 N 100 N 100 N 100 N 100 N 100 N 100 N 100 N
 B    050 W 050 W 050 W 050 W 050 W 050 W 050 W 050 W 050 W 050 W 050 W 050 W
 C    008 W 005 W 008 W 005 W 008 W 005 W 008 W 005 W 008 W 005 W 008 W 005 W
 D    008 S 005 S 008 S 005 S 008 S 005 S 008 S 005 S 008 S 005 S 008 S 005 S
```

Figure 7. QSN Table Maintenance panel

7. Use this panel to modify the following parameters for uncommitted messages:

   • Start and stop percentages individual process types

   • Start and stop actions for individual process types

8. Press Enter to save or cancel your settings.

   The Update Confirmation panel is displayed.

9. Type Y or N and press Enter.

   The QSN Table Names panel is displayed.

Configuring type 3 queue overflow protection (AREA/FAILSAFE)

Type 3 overflow protection provides the most advanced overflow protection for IMS Queue Control Facility.

Topics:

AREA settings (type 3 overflow protection)

Initial area and failsafe values for type 3 queue overflow protection are set by the user as control statements in the IMS Queue Control Facility extensions PROCLIB member.

Area and failsafe values for type 3 overflow protection that are modified through the TSO client are valid only for the current IMS session. An IMS restart causes IMS Queue Control Facility to revert to the initial settings contained in the IMS Queue Control Facility extensions PROCLIB member.

You can preserve these settings across multiple IMS sessions by starting a procedure after an IMS restart that restores the values to IMS Queue Control Facility.

Configuration summary for type 3 overflow protection (AREA)

Type 3 AREA overflow protection has the following attributes:

- Both AREA and FAILSAFE type 3 overflow protection methods are defined in the IMS Queue Control Facility extensions PROCLIB member.
- The existence of AREA and FAILSAFE statements in the IMS Queue Control Facility extensions PROCLIB member enables type 3 overflow protection.
- AREA protection can prevent the overflow of IMS queues by identifying excessive usage of specific queues due to an individual IMS entity (process).
- Specific usage is defined as queue usage by a specific IMS entity, such as an application (dependent region) or terminal.
- Analysis is performed on the accumulation of the queue activity due to a specific IMS process.
- In AREA overflow protection, the queue can be divided into up to ten AREA partitions. Each AREA partition is expressed as a percentage of the total queue space.
- You can also implement automatic unload of committed messages by configuring the ONTHRESHOLD*xx* statement in the extension member. You can use the supplied IMS Queue Control Facility sample unload jobs.
- AREA and FAILSAFE overflow protection logics can be run individually or concurrently.
- AREA and FAILSAFE overflow protection mechanisms are optional.
- If AREA and FAILSAFE overflow protection are not specified, then Type 2 overflow protection is enabled.
- If AREA and FAILSAFE overflow protection are specified, then Type 2 overflow protection is disabled.

See also "7. Enabling and configuring the IMS Queue Control Facility extensions" on page 31.

AREA analysis

AREA overflow protection uses three types of analysis to identify excessive specific queue usage:

- Source (messages sent) analysis tries to determine the IMS process that causes excessive queue usage
- Destination (messages arriving into the queue) analysis tries to determine the IMS process that triggers the queue overflow
- Destination analysis additionally can check for other source IMS processes that impact the queue overflow condition

The same AREA card is applicable to both source and destination analysis.

Source analysis (cause of messages origin)

The five types of Sources are: APPL (dependent region application program), DC (terminal), APPC (terminal), OTMA (terminal), and MSC (terminal).

1. Determine if specific queue usage meets the AREA specification.
2. Calculate specific queue usage criteria by multiplying the Area TOTAL by the Area USED.
3. Analyze the queue usage of the specific IMS entity to determine if it exceeds the calculated specific queue usage.
4. Take specified action if true.

Destination analysis (cause of queue overflow trigger)

The two types of Destinations are: transactions (dependent region programs) and terminals.

1. Determine if specific queue usage meets the AREA specification.
2. Calculate specific queue usage criteria by multiplying the Area TOTAL by the Area USED.
3. Analyze the queue usage of the specific IMS destination entity to determine if it exceeds the calculated specific queue usage.
4. Take the WTO action if true.

Destination analysis (other source of message origin affecting queue overflow)

Destination analysis for "other" sources is done to detect an IMS entity that by itself is not directly responsible for triggering the queue overflow, but might be indirectly responsible for the overall excessive queue usage. A common example of this would be a looping terminal.

1. Determine if a specific source entity was found to be responsible for the excessive queue usage of the destination entity.

2. If one was not found then do additional analysis to determine a likely candidate.

Area TOTAL and USED percentages, used to determine levels of specific queue usage, apply to both source analysis and destination analysis:

• Use the PERCENT=(TOTAL=) setting to specify the size of an area.

• Use the PERCENT=(USED=) setting to specify the percentage of that area used.

IMS entity types used for AREA analysis

IMS Queue Control Facility can evaluate associate queue usage with particular entities.

The TYPE=(*IMS-entity*) parameter identifies a specific IMS process entity. You can select any combination of the following types:

TYPE=APPL
A dependent region application program

TYPE=DC
A VTAM terminal

TYPE=APPC
An LU 6.2/APPC application

TYPE=OTMA
An OTMA application

TYPE=MSC
An IMS-to-IMS communications

TYPE=ALL
Select all entities

TYPE=NONE
Select no entities

AREA card action states

The actions of AREA cards are divided into four states. These states are used by Source Analysis and Destination Analysis for the analysis of IMS processes and resources. The applicability of the states is dependent on the type of process or resource being analyzed. In some cases, a state is not applicable for that particular process or resource.

The action states are described in the following list:

CSTOPACTION
In source and destination analysis, CSTOPACTION specifies an action when the cause of excessive queue usage that triggers the AREA statement is in a stopped state (not applicable in some cases).

Source analysis example: Not applicable.

For a PST or CLB, the CSTOPACTION parameter has no effect. If a PST or CLB is in a stopped state, then there should be no usage of records or buffers for IMS to report for the PST or CLB.

Destination analysis example:

Analysis has determined that a destination (SMB or CNT) is in a stopped state, and the queue usage has exceeded area specifications (for example, a stopped transaction).

The only valid actions for the stopped destination are NONE and WTO. The other action options have no affect on the already stopped destination and default to WTO.

CSTARTACTION
In source and destination analysis, CSTARTACTION specifies an action when the cause of excessive queue usage that triggers the AREA statement is in a started state.

Source analysis example:

Analysis has determined that a source (PST or CLB) is in a started state and its queue usage has exceeded area specifications.

When a PST or CLB is in a started state, there could be excessive queue usage if the application or LTERM is looping. Valid actions for the source are: NONE, WTO, WAIT, STOP, and ABEND.

Destination analysis example:

Analysis has determined that a destination (SMB or CNT) is in a started state and its queue usage has exceeded area specifications.

When an SMB or CNT is in a started state and the messages are queuing up, the valid actions for the destination are NONE and WTO. The other actions are not valid and default to WTO.

OSTOPACTION
In source and destination analysis, OSTOPACTION specifies an action when another ("other") more significant contributor to excessive queue usage is in a stopped state.

Source analysis example: Not applicable.

For a PST or CLB, the CSTOPACTION parameter has no effect. If a PST or CLB is in a stopped state, then there should be no usage of records or buffers for IMS to report for the PST or CLB.

Destination analysis example:

Additional analysis is performed to locate another destination if one was not found in CSTOPACTION analysis. The other destination is identified as being in a stopped state.

When an SMB or CNT is in a stopped state and the messages begin to queue up, the valid actions for the destination are NONE and WTO. The other actions are not valid and default to WTO.

OSTARTACTION
In source and destination analysis, OSTARTACTION specifies an action when another ("other") more significant contributor to excessive queue usage is in a started state.

Source analysis example:

Additional analysis is performed to locate another source if one was not found in CSTARTACTION analysis.

When a group of PSTs or CLBs are in a started state, there could be very high use of records or buffers for IMS to report if the applications/LTERMs are outputting to a specific destination. The valid actions for the source are NONE or WTO.

Destination analysis example:

Additional analysis is performed to locate another destination if one was not found in CSTARTACTION analysis. The other destination is identified as being in a started state.

When an SMB or CNT is in a started state and the messages begin to queue up, the valid actions for the destination are JTSTP, NONE and WTO. The other actions are not valid and default to WTO.

The following action values apply as indicated:

    NONE - applicable to all TYPE values
    WTO - applicable to all TYPE values
    WAIT - applicable to TYPE=APPL only
    STOP - applicable to all TYPE values
    ABEND - applicable to TYPE=APPL only

JTSTP - applicable to all TYPE values

Determining AREA statement settings

You can specify up to ten areas, each with a specific TOTAL and USED percentage to achieve the required degree of queue protection. For example:

```
                          Total Queue Space
  0%-------------------------------------------------------------------100%

   Area 1: 0%------------25%
```

The control statement for Area 1 would appear as: PERCENT=(TOTAL=25, ...)

In addition, you can specify how much this Area 1 is to be considered in evaluating how much queue usage is occurring. For example:

```
             Area 1 Space
  0%---------------------------25%

   Percentage of Area 1: 0%------------50%
```

The control statement for this fraction of Area 1 would appear as: PERCENT=(TOTAL=25,USED=50)

The specification of 50% usage of Area 1 would yield an effective total queue space usage of 12.5% (.50 x .25=.125). An action or notification is triggered when a process takes up 50% of Area 1 (defined as 25% of the total queue). This results in protection of the queue at 12.5% effective usage of the total queue space.

This mathematical methodology allows you to fine-tune protection requirements within a single defined area for individual IMS entity types (APPL, DC, APPC, OTMA, MSC).

As a cautionary note, be aware that it is possible to have the same effective total queue space percentage resulting from different combinations of TOTAL and USED percentages. For example. the following two statements both result in 15%.

```
PERCENT=(TOTAL=30,USED=50)
PERCENT=(TOTAL=60,USED=25)
```

Example AREA statements for a production environment

The following example shows sample configuration from the IMS Queue Control Facility extensions PROCLIB member:

```
AREA=(ID=AREA0001,                    Notify when a specific
      PERCENT=(TOTAL=20,USED=25),     process (ALL) uses 05% of queue
      TYPE=(ALL),
      CSTOPACTION=WTO,
      CSTARTACTION=WTO,
      OSTOPACTION=WTO,
      OSTARTACTION=WTO)

AREA=(ID=AREA0002,                    Wait a specific process
      PERCENT=(TOTAL=25,USED=40),     (OTMA/MSC) that uses
      TYPE=(OTMA,MSC),                10% of queue
      CSTOPACTION=WTO,
      CSTARTACTION=WAIT,
      OSTOPACTION=WTO,
      OSTARTACTION=WTO)

AREA=(ID=AREA0003,                    Stop a specific process
      PERCENT=(TOTAL=30,USED=50),     (DC/Dependent Region)
      TYPE=(APPL,DC),                 that uses 15% of queue
      CSTOPACTION=WTO,
      CSTARTACTION=STOP,
      OSTOPACTION=WTO,
```

```
        OSTARTACTION=WTO)

AREA=(ID=AREA0004,                    Abend a specific process
      PERCENT=(TOTAL=40,USED=50),     (ALL)
      TYPE=(ALL),                     that uses 20% of queue
      CSTOPACTION=WTO,
      CSTARTACTION=ABEND,
      OSTOPACTION=WTO,
      OSTARTACTION=WTO)


...up to 10 AREA statements
```

FAILSAFE settings (type 3 overflow protection)

Initial area and failsafe values for type 3 queue overflow protection are set by the user as control statements in the IMS Queue Control Facility extensions PROCLIB member.

Area and failsafe values for type 3 overflow protection that are modified through the TSO client are valid only for the current IMS session. An IMS restart causes IMS Queue Control Facility to revert to the initial settings contained in the IMS Queue Control Facility extensions PROCLIB member.

You can preserve these settings across multiple IMS sessions by starting a procedure after an IMS restart that restores the values to IMS Queue Control Facility.

Configuration summary for type 3 overflow protection (FAILSAFE)

Type 3 FAILSAFE overflow protection has the following attributes:

- Both AREA and FAILSAFE type 3 overflow protection methods are defined in the IMS Queue Control Facility extensions PROCLIB member.
- The existence of AREA and FAILSAFE statements in the IMS Queue Control Facility extensions PROCLIB member enables type 3 overflow protection.
- FAILSAFE protection can prevent the overflow of IMS queues by identifying excessive total queue usage of IMS Queue Manager queues.
- Total queue usage is defined as the sum of queue usage by all specific IMS processes (entities).
- FAILSAFE protection analyzes total queue utilization. Analysis is performed on the accumulation of the queue activity due to all IMS processes.
- In FAILSAFE overflow protection, the queue can be divided into up to ten FAILSAFE partitions. Each FAILSAFE partition is expressed as a percentage of the total queue space. Each partition represents total queue usage as an accumulation of all IMS entities queue activity.
- You can also implement automatic unload of committed messages by configuring the ONTHRESHOLD*xx* statement in the extension member. You can use the supplied IMS Queue Control Facility sample unload jobs.
- FAILSAFE protection differs from Area protection in that FAILSAFE targets total queue usage, not specific queue usage.
- FAILSAFE may allow any specific IMS entity to continue processing when FAILSAFE has been activated.
- AREA and FAILSAFE overflow protection logics can be run individually or concurrently.
- AREA and FAILSAFE overflow protection mechanisms are optional.
- If AREA and FAILSAFE overflow protection are not specified, then Type 2 overflow protection is enabled.
- If AREA and FAILSAFE overflow protection are specified, then Type 2 overflow protection is disabled.

See also .

FAILSAFE analysis

FAILSAFE uses two types of analysis to identify excessive total queue usage:

Source analysis

1. Determine if the total queue usage meets the FAILSAFE TOTAL specification.

2. Identify the current source at the time that FAILSAFE is activated.

3. Implement specified actions against this and all IMS ensuing entities.

4. Identify whether a specific source can continue processing from the USED specification.

Destination analysis

1. Determine if the total queue usage meets the FAILSAFE TOTAL specification.

2. Identify the current destination at the time that FAILSAFE is activated.

3. Implement the notification action against this and all subsequent IMS entities.

FAILSAFE TOTAL and USED percentages, used to determine levels of specific queue usage, apply to both source analysis and destination analysis:

- Use the PERCENT=(TOTAL=) setting to calculate the size of the FAILSAFE.
- Use the PERCENT=(USED=) setting to set a limit that will allow a specific process to continue running even though a FAILSAFE specification has been met.

  For example, a FAILSAFE TOTAL is set to WAIT everything at 75% of queue usage (TOTAL=75). However, this action would also prevent the IMS operator from issuing commands because the commands would be stopped by the FAILSAFE. By adding the USED=01 setting, you can allow those commands that use less than 1% of the queues to run.

```
FAILSAFE=(ID=FAIL0001,
          PERCENT=(TOTAL=75,USED=01),
          ACTION=WAIT)
```

FAILSAFE protection will allow a specific IMS entity to continue processing as long as its queue usage is less than the specified USED percentage.

IMS entity type used for FAILSAFE analysis

FAILSAFE does not allow targeting of specific IMS entities; therefore, the TYPE= parameter is not allowed. An implicit TYPE=ALL is enforced, which includes the following entities:

TYPE=APPL
A dependent region application program

TYPE=DC
A VTAM terminal

TYPE=APPC
An LU 6.2/APPC application

TYPE=OTMA
An OTMA application

TYPE=MSC
An IMS-to-IMS communications page

FAILSAFE actions

FAILSAFE actions are global. They are applicable against all IMS processes.

FAILSAFE actions are implemented by a single action parameter: ACTION=(*action-value*).

- The following action values apply as indicated:

  NONE - applicable to all TYPEs
  WTO - applicable to all TYPEs
  WAIT - applicable to all TYPEs
  STOP - applicable to all TYPEs

ABEND - applicable to TYPE=APPL only (all others=STOP)

You can specify up to ten FAILSAFE actions, each with a specific TOTAL and USED percentage to achieve the required degree of queue protection. For example:

```
                    Total Queue Space
  0%-------------------------------------------------------------------100%


  Failsafe 1:   0%-----------------------------------------------70%
```

The control statement for this FAILSAFE would appear as: PERCENT=(TOTAL=70,...)

In addition, you can specify that a process be allowed to continue if its queue usage is less than a specified percentage. For example:

```
                    Total Queue Space
  0%-------------------------------------------------------------------100%


  Failsafe 1:   0%-----------------------------------------------70%

  Individual process: 0%-2%
```

The control statement for this FAILSAFE would appear as: PERCENT=(TOTAL=70,USED=02)

The specification of 2% queue usage on FAILSAFE 1 allows an individual process (using less than 2% queue space) to continue running even though the FAILSAFE 1 limit for total multiple processes (70%) has been reached.

A specification of 0% queue usage (USED=00) means that no process, regardless of size, is allowed to continue.

Example FAILSAFE statements for a production environment

The following example shows sample configuration from the IMS Queue Control Facility extensions PROCLIB member:

```
FAILSAFE=(ID=FAIL0001,              Total queue usage is 55%;
      PERCENT=(TOTAL=55,USED=05),       do notify of any process
      ACTION=WTO)                       that uses more than 5% of queue

FAILSAFE=(ID=FAIL0002,              Total queue usage is 60%;
      PERCENT=(TOTAL=60,USED=04),       do notify of any process
      ACTION=WTO)                       that uses more than 4% of queue

FAILSAFE=(ID=FAIL0003,              Total queue usage is 65%;
      PERCENT=(TOTAL=65,USED=03),       do notify of any process
      ACTION=WTO)                       that uses more than 3% of queue

FAILSAFE=(ID=FAIL0004,              Total queue usage is 70%;
      PERCENT=(TOTAL=70,USED=02),       do notify of any process
      ACTION=WTO)                       that uses more than 2% of queue

FAILSAFE=(ID=FAIL0005,              Total queue usage is 75%;
      PERCENT=(TOTAL=75,USED=01),       do wait any process
      ACTION=WAIT)                      that uses more than 1% of queue

FAILSAFE=(ID=FAIL0006,              Total queue usage is 80%, do
      PERCENT=(TOTAL=80,USED=00),       stop everything
      ACTION=STOP)

...up to 10 FAILSAFE statements
```

FAILSAFE card example 1

The following example shows sample configuration from the IMS Queue Control Facility extensions PROCLIB member.

```
FAILSAFE=(ID=FAIL0001,
      PERCENT=(TOTAL=70,USED=02),
      ACTION=WTO),
```

FAIL0001: This FAILSAFE will be activated when total queue usage is greater than 70% (TOTAL=70).

A notification (ACTION=WTO) reporting queue usage greater than 70% will be issued. The notification is sent only one time, so that the MVS console is not flooded with WTOs.

The IMS entity identified in the WTO might not actually be the one responsible for excessive queue usage. That entity was simply the first IMS entity whose queue request activated FAILSAFE.

Note that this entity is using more than 2% of the total queue (USED=02), which is why it activated the WTO.

The specified action (ACTION=*action-type*) is not applied against IMS entities that use less than 2% queue usage.

FAILSAFE card example 2

The following example shows sample configuration from the IMS Queue Control Facility extensions PROCLIB member.

```
FAILSAFE=(ID=FAIL0002,
      PERCENT=(TOTAL=75,USED=00),
      ACTION=WAIT),

...up to 10 FAILSAFE statements
```

FAIL0002: This FAILSAFE will be activated when total queue usage is greater than 75% (TOTAL=75).

A wait will be initiated for each IMS entity (ACTION=WAIT).

All IMS entities will be waited (no exceptions) when the 75% FAILSAFE limit is reached because USED=00 is specified.

Preserving modified AREA/FAILSAFE settings across IMS sessions

Initial area and failsafe values for type 3 queue overflow protection are set by the user as control statements in the IMS Queue Control Facility extensions PROCLIB member.

You can modify these "hard-coded" values through the IMS Queue Control Facility TSO client. Member IQCQSUNP captures and stores any area and failsafe values modified through the TSO client.

Area and failsafe values for type 3 overflow protection that are modified through the TSO client are valid only for the current IMS session. An IMS restart causes IMS Queue Control Facility to revert to the initial settings contained in the IMS Queue Control Facility extensions PROCLIB member.

You can preserve these settings across multiple IMS sessions by starting a procedure after an IMS restart that restores the values to IMS Queue Control Facility.

JCL to restore modified type 3 overflow protection settings

The following code shows example JCL that reads member IQCQSUNP and restores the default and modified type 3 overflow protection values to IMS Queue Control Facility.

In this example, the JCL job name is IQCLDTB3. You can use any name for the JCL. The IMSMSG procedure in the IMS Control Region DFSPB*xxx* member must be configured to run this IQCLDTB3 procedure at IMS restart.

```
//*-------------------------------------------------------------------
//*      LICENSED MATERIALS - PROPERTY OF IBM
//*
//*      IMS QCF FOR Z/OS VERSION 3 RELEASE 2
//*
//*      5697-N50
//*
//*      COPYRIGHT IBM CORP. 2000, 2014
//*
//*      ALL RIGHTS RESERVED.
//*      US GOVERNMENT USERS RESTRICTED RIGHTS -
//*      USE, DUPLICATION OR DISCLOSURE RESTRICTED
//*      BY GSA ADP SCHEDULE CONTRACT WITH IBM CORP.
//*
//*-------------------------------------------------------------------
//*
//* NAME: IQCLDTB3
//* DESC: SAMPLE PROCEDURE - USED TO RESTORE THE CUSTOM TYPE 3
//*       OVERFLOW PROTECTION SETTINGS THAT ARE STORED IN IQCQSUNP
//*       MEMBER. REFER TO USER'S GUIDE FOR ADDITIONAL INFORMATION.
//*
//* FUNCTION: RESTORE THE CUSTOM TYPE 3 OVERFLOW PROTECTION SETTINGS
//*           FROM THE IQCQSUNP MEMBER AFTER AN IMS RESTART
//*
//*-------------------------------------------------------------------
//* NOTES =
//*
//* 1) REVIEW ALL STATEMENTS BEFORE STORING THE JCL
//* 2) &IMSID IS PARAMETER, PASSED ON START OF JCL
//*    &PLEXN IS PARAMETER, PASSED ON START OF JCL
//* 4) CHANGE IMSHLQ.RESLIB TO THE VALID IMS RESLIB NAME
//* 5) CHANGE HLQUAL.SIQCLINK TO THE VALID QCF LOADLIB NAME
//* 6) CHANGE THE HLQUAL FIELDS TO VALID HIGH LEVEL QUALIFIERS
//*    THAT COMPLY WITH YOUR SITES NAMING STANDARDS.
//* 7) CHANGE IMSID     TO THE IMSID OF CONTROL REGION
//*    CHANGE PLEXN     TO THE REAL PLEX NAME
//* 8) CHANGE THE SYSDA  FIELD TO A VALID LOCATION UNIT
//*    IDENTIFICATION THAT COMPLIES WITH YOUR SITES NAMING
//*    STANDARDS.
//*
//IQCLDTB3 PROC RGN=0M,
//             SOUT=H,
//             PLEXN=plexn,
//             IMSID=imsid
//*
//*-------------------------------------------------------------------
//STEP1    EXEC PGM=IQCINI0$,REGION=&RGN,TIME=1440,
// PARM='BPEINIT=IQCBINI0,BPECFG=IQCBACNF,IMSPLEX=&PLEXN,QCFIMS=&IMSID'
//*
//STEPLIB  DD  DISP=SHR,DSN=hlqual.SIQCLINK
//         DD  DISP=SHR,DSN=imshlq.RESLIB
//*
//PROCLIB  DD  DISP=SHR,DSN=hlqual.PROCLIB
//*
//SNAPDUMP DD  SYSOUT=&SOUT
//SYSUDUMP DD  SYSOUT=&SOUT
//SYSPRINT DD  SYSOUT=&SOUT
//QCFPRINT DD  SYSOUT=&SOUT
//*
//QCFIN    DD  DISP=SHR,DSN=IQC.mvsid.imsid.NOTPARMS(IQCQSUNP)
```

Configuring the IMSMSG procedure to run the IQCLDTB3 procedure

To run the IQCLDTB3 procedure after an IMS restart, you must specify the IMSRDR procedure in the IMS Control Region DFSPB*xxx* member using the PRDR parameter:

```
PRDR=IMSRDR
```

The IMSRDR procedure is used to read an IMSMSG job that initiates the IMS control program region and message regions.

By adding the following JCL into your IMSMSG procedure, the IQCLDTB3 procedure is run at IMS start-up. (The IQCLDTB3 procedure restores your modified type 3 overflow protection settings from member IQCQSUNP.)

```
//**********************************************************************
//* QCF 3.2
//* This job executes the procedure IQCLDTB3 which is used to
//* restore custom type 3 overflow protection procedure settings.
//* Change imsid to the IMS ID of the Control Region.
//* Change hlqual to the valid high level
qualifier.
//**********************************************************************
//*
//IQCLDTB3 JOB IMS,
// 'imsid',
// CLASS=A,
// MSGCLASS=H,MSGLEVEL=(1,1),
// REGION=0M
//*
/*JOBPARM SYSAFF=RSI3
// JCLLIB ORDER=(hlqual.PROCLIB)
//*
//STEP1    EXEC IQCLDTB3
//*
```

Configuring type 3 overflow protection using the TSO client

The IMS Queue Control Facility uses AREA and FAILSAFE configuration values set in the IMS Queue Control Facility extensions PROCLIB member to implement type 3 overflow protection.

About this task

The following configuration notes apply to type 3 overflow protection using the TSO client:

- Both Area and Failsafe type 3 overflow protection methods are defined in the IMS Queue Control Facility extensions PROCLIB member.
- The existence of AREA and FAILSAFE statements in the IMS Queue Control Facility extensions PROCLIB member enables type 3 overflow protection.

  Type 2 protection (the product default) is implemented only if there are no type 3 AREA and FAILSAFE configuration settings specified in the extensions member.

- The existence of AREA and FAILSAFE settings in the IMS Queue Control Facility extensions PROCLIB member activates the appropriate type 3 menu option (**6**) on the QCF Main Menu.

  Use QCF Main Menu option **6 - Notify** to configure type 3 overflow protection.

  QCF Main Menu option **5 - Tables**, which is used to configure type 2 overflow protection, is not active.

- Configuration changes made with the IMS Queue Control Facility TSO client are valid only for the current IMS session. They are lost when the IMS is terminated.

  The restart of the IMS will come up original default values and with whatever values are contained in the configuration members.

  The configuration member is where you put the values you want to be always in effect.

- Before you begin the following procedure, ensure that you have selected the appropriate server and IMS ID that reads the appropriate IMS Queue Control Facility extensions PROCLIB member.

Procedure

Perform the following steps to configure type 3 queue overflow protection:

1. From the QCF Main menu, select option **6 - Notify**.

   The Queue Overflow Parameters Sub-Menu panel is displayed.

2. Select option **1 - Select queue space notification table**.

The Queue Space utilization notification parameters panel is displayed.

3. Select option **1 - Edit queue space utilization notification parameters**.

   The Queue space utilization notification parameters table panel is displayed.

4. Type the E (Edit) row action in the row representing the current overflow parameters table.

   The Queue Threshold Parameters panel is displayed.

```
  Help

 ─────────────────────────────────────────────────────────────────────────
Command ===>

Press ENTER to continue or END to exit.          APAR . . : PH19814 2020/01/
  Server . . : IQCSERV9                           JDTE . . : 2020.016
  IMS ID . . : IMS1                               TIME . . : 15:59:46
  Member . . : IQCQSUNP                           DATE . . : 2020/01/16

 Committed messages: Thresholds.
 Queue  upper  threshold  . . . 075  Queue  lower  threshold . . . . 060
 Queue IQC6101 threshold  . . . 085

 Committed messages: Job/Procedure names to be called at threshold crossover.
 Ignore application calls inserting messages to express PCBs . . . . Y

 Jobnames  - A-B:  . . . . NOUNLOAD  B-C: . . . NOUNLOAD  C-D: . . . NOUNLOAD
 Procnames - A-B:  . . . .           B-C: . . .           C-D: . . .

 Jobname    - IQC6101  . . NOUNLOAD
 Procname   - IQC6101  . .
```

Figure 8. Queue Threshold Parameters panel

5. Use this panel to modify the following parameters for committed and uncommitted messages:

   • Queue upper threshold (QTU)

   • Queue lower threshold (QTL)

   • IQC6101 threshold

   • JOB and procedure names for actions (including automatic unload of committed messages) that take place for committed messages when threshold boundaries are exceeded

6. Press Enter to continue to the configure AREA settings.

   The Queue Overflow Notification Parameters panel is displayed.

```
  Help

 ────────────────────────Queue Overflow Notification Parameters────────────
 Command ===>                                           Scroll ===> PAGE

 ENTER to continue, END to go to previous panel.   APAR . . : PH19814 2020/01/
 Server . . : IQCSERV9                             JDTE . . : 2020.016
 IMS ID . . : IMS1                                 TIME . . : 17:09:50
                                                   DATE . . : 2020/01/16
 Select Source/Destination types with a "/"
 Culprit and other actions: A - ABEND, N - NONE, O - WTO, S - STOP, W - WAIT,
                          J-JTSTP

  Area      Percent  _Source/Destination Types_ _Culprit Act _ __Other Act_
  Name      Tot Usd  ALL APPC APPL  DC MSC OTMA Strtd  Stppd Strtd  Stppd

 AREA0001  10  10    /                           O      O      O      O
 AREA0002  20  20    /                           O      O      O      O
 AREA0003  30  30    /                           O      O      O      O
 AREA0004  40  40    /                           O      O      O      O
```

Figure 9. Queue Overflow Notification Parameters panel

7. Press Enter to continue to the configure FAILSAFE settings.

The Queue Overflow Notification Parameters panel is displayed.

```
 Help
 -------------------------------------------------------------------------
                    Queue Overflow Notification Parameters
Command ===>                                              Scroll ===> PAGE

ENTER to continue, END to go to previous panel.   APAR . . : PH19814 2020/01/
Server . . : IQCSERV9                             JDTE . . : 2020.016
IMS ID . . : IMS1                                 TIME . . : 17:13:51
                                                  DATE . . : 2020/01/16
 Select percent queue utilization and action for each FAILSAFE statement.
 Actions: A - ABEND, N - NONE, O - WTO, S - STOP, W - WAIT


                            PERCENT
FAILSAFE ID     PERCENT     ALLOWED        ACTION
FAIL0001        50          05             O
FAIL0002        60          04             O
```

Figure 10. Queue Overflow Notification Parameters panel

8. Use this panel to define FAILSAFE partitions and actions.

9. Press Enter to save or cancel your settings.

   The Update Confirmation panel is displayed.

10. Type Y or N and press Enter.

   The Queue space utilization notification parameters table panel is displayed.

Configuring automatic unload of committed messages

Automatic unload of committed messages is a feature of type 2 and type 3 queue overflow protection that allows you to regain queue space by removing messages from the queue and relocating them.

The default for all thresholds is to not perform an unload.

You can enable automatic unload by modifying the ONTHRESHOLD*xx* statements in the IMS Queue Control Facility extensions PROCLIB member.

Regardless of whether you are implementing type 2 or type 3 overflow protection, the mechanism for automatic unload of messages from the queue always make use of the four threshold queue space partition model.

Configuring unload of committed messages

You use the IMS Queue Control Facility extensions PROCLIB member to configure JOBs and PROCLIBs to perform queue unload actions.

You can specify the names for the JOBs or PROCLIBs that IMS Queue Control Facility starts when thresholds are reached. Specify job names for the jobs that are started when the thresholds A-B, B-C, C-D, and the threshold that you specified are reached.

These jobs take action on committed messages. Committed messages are processed messages that are ready to be removed from the queue.

The *hlq*.SIQCSAMP (IQCIMSII) member contains a sample extensions configuration. Samples of the jobs are provided in the following members:

- IQCPRCAB
- IQCPRCBC
- IQCPRCCD
- IQCPRCUT

QCFIN DD statements for these jobs point to the default members with control cards IQCQSNAB, IQCQSNBC, IQCQSNCD, and IQCQSNUT in the partitioned data set for committed message processing. You can modify the control cards for the jobs by using the TSO client panels.

If you want to use the supplied IMS Queue Control Facility sample unload jobs, you can specify those names on the ONTHRESHOLD*xx* statements of the IMS Queue Control Facility extensions PROCLIB member.

The following example shows the format of the threshold settings in the extensions configuration member:

```
ONTHRESHOLDAB
    (
    JOBNAME=NOUNLOAD  ..do nothing..
    JOBNAME= A procedure member name for a QCF Batch Address Space
    PROCNAME= A QCF Server Address Space command member
    )
ONTHRESHOLDBC same format as ONTHRESHOLDAB
ONTHRESHOLDCD same format as ONTHRESHOLDAB
ONTHRESHOLDUT same format as ONTHRESHOLDAB
```

PROCLIB conditions:

- 1- to 8-character PROCLIB member name
- IMS Queue Control Facility extensions will send this member name to the IMS Queue Control Facility server.
- If a server is not present, IMS Queue Control Facility uses the jobname.

JOB conditions:

- 1- to 8-character batch job name
- IMS Queue Control Facility extensions will update the overflow table with this name during initialization.

IMS Queue Control Facility will submit the member name to the z/OS internal reader.

If NOUNLOAD is specified (default), IMS Queue Control Facility extensions will not submit the JCL to start the IMS Queue Control Facility default batch job.

Using generation data set groups (GDG)

If the same threshold is crossed again before processing is completed for the unloaded data set from the first crossing, your original unloaded data can be overwritten. Using generation data set groups (GDG) for the threshold jobs will help prevent overwriting the unloaded data for a given threshold.

If an IMS application or an IMS intelligent terminal is looping in error and inserting messages to either transactions or terminal destinations, one or more of the thresholds could be crossed repeatedly.

The intent of the threshold jobs is to unload committed messages from IMS, while the ACTION setting for the areas is to take an action on the IMS application or intelligent terminal.

The following example shows a PROCLIB that specifies a GDG for the UNLOAD data set:

```
 BROWSE    USER.PRIVATE.PROCLIB(QCFCMDBC) - 01.99    Line 00000000 Col 001 080
 Command ===>                                          Scroll ===> PAGE
******************************** Top of Data *********************************
IMS(IMSA)
QCFIN(DSN(USER01.PRIVATE.PROCLIB),MEM(UNBC))
QCFPRINT(QCFTEST.BC.QCFPRINT)
QUERY(QCFTEST.BATCH.QUERY)
ULOADGDG(DSN(IMSTESTG.QCF),UNIT(SYSDA) VOLSER(IMSDCL)
SPACE(1) SPACEUNIT(CYL))
SCRAPLOG(QCFTEST.BC.SCRAPLOG)
```

Guidelines for using AREA and FAILSAFE statements

The IMS Queue Control Facility AREA and FAILSAFE statements are intended to provide you with a methodology and a level of support to prevent the condition where the IMS message buffer queue space can become exhausted and IMS terminates abnormally.

See also Chapter 23, "Recovering from AREA and FAILSAFE actions," on page 439.

Reasons why queue space message buffers become exhausted

The major reasons that the IMS queue space message buffers can become exhausted are for any combination of the following reasons:

- Insufficient Dependent Regions to process all inputted transactions from an input device or from another transaction (insert to ALTPCB)
- Customer Program stopped, therefore building up the queue count for that transaction
- Looping IMS transaction sending output back to the input device, or to one or more other destinations
- Looping IMS intelligent input device looping on sending in 1 to n message segments
- Transactions that produce a larger volume of output data than the destination (such as another transaction, output to the inputting device, or an alternate output device) can process
- Input devices that produce a larger volume of data than the destination (such as a non-response transaction) can process

There are also other factors, in combination with the above listed reasons, that can require you to make additional considerations.

- If the transaction types are all response mode transactions in your environment, then the definition of the AREA and FAILSAFE statements are much easier to define.
- If the transaction types also include non-response mode transactions, and the larger the volume of non-response transactions in your environment, then the definition of the AREA and FAILSAFE statements are more difficult to define. You might want to include more of these statements in your definition.

Setting ACTION and TOTAL values

AREA and FAILSAFE statements provide you with notification information about the input devices and transactions when the inserting of messages to the IMS message buffer queue space has met the definitions defined in these statements.

When selecting the action to be taken on the FAILSAFE statement that defines the largest TOTAL=*percentage,* you must be careful to choose the correct ACTION=*value.* It is also very important to set the correct FAILSAFE and AREA statement TOTAL=*percentage* value.

The setting of both the TOTAL= and ACTION= values need to take into account the types of input devices and transactions. If your system is largely made up of response mode transactions rather than non-response transactions, then the setting of the TOTAL= and ACTION= is less restrictive.

If however, you have a large volume of non-response transactions from the OTMA network, then you will need to consider setting a lower TOTAL= *value* with the ACTION= set to STOP or ABEND.

If you have a run away OTMA client that is looping and sending in non-response transactions, and you specify ACTION=WTO or ACTION=WAIT, the problem will continue.

If you have a stopped program, and you specify ACTION=WTO or ACTION=WAIT, the problem will continue as well.

Considerations when defining AREA and FAILSAFE statements

Consider the following items when defining AREA and FAILSAFE statements:

- Response mode transaction input can require from 1 to n IMS queue space message buffers

These buffers will remain allocated if:

– ACTION=WTO is specified, or
– ACTION=WAIT is specified

However, for (ACTION=WAIT) an IMS queue space message buffer will be allocated to send out an IMS message to the MVS console to identify that a transaction on input device has been added to the QCF WAIT Queue.

Once the IMS message has been sent, the IMS queue space message buffer for the IMS wait notification message will be deallocated.

In both cases (ACTION=WTO and ACTION=WAIT) for the response mode transaction the IMS message queue space message buffer will remain allocated for the input message. IMS queue space message buffer count will increase.

• Response mode transaction input can require from 1 to *n* IMS queue space message buffers

These buffers will be deallocated if:

– ACTION=STOP is specified, or
– ACTION=ABEND is specified

However, for both ACTION=STOP and ACTION=ABEND, an IMS queue space message buffer will be allocated to send out an IMS message to notify the input device that the input message was rejected (ACTION=STOP), or to send out an IMS message to notify the input device that application has terminated abnormally (ACTION=ABEND).

Once the IMS message has been sent to the input device, the IMS queue space message buffer for the IMS error message will be deallocated.

In both response mode transaction cases (ACTION=STOP and ACTION=ABEND), the input message in the IMS message queue space message buffer will be deallocated in addition to the IMS error messages. The IMS message queue space message buffer count will increase for the input message, and IMS error message will decrease by the same amount for these actions.

• Non-response mode transaction input can require from 1 to *n* IMS queue space message buffers

These buffers will remain allocated if:

– ACTION=WTO is specified, or
– ACTION=WAIT is specified

However, for ACTION=WAIT, an IMS queue space message buffer will be allocated to send out an IMS message to the MVS console to identify that a transaction on input device has been added to the QCF WAIT Queue.

Once the IMS message has been sent, the IMS queue space message buffer for the IMS wait notification message will be deallocated.

In both non-response mode transaction cases (ACTION=WTO and ACTION=WAIT), the IMS message queue space message buffer will remain allocated for the input message. IMS queue space message buffer count will increase.

• Non-response mode transaction input can require from 1 to *n* IMS queue space message buffers

These buffers will be deallocated if:

– ACTION=STOP is specified, or
– ACTION=ABEND is specified

However, for both ACTION=STOP and ACTION=ABEND, an IMS queue space message buffer will be allocated to send out an IMS message to notify the input device that the input message was rejected (ACTION=STOP), or to send out an IMS message to notify the input device that application has terminated abnormally (ACTION=ABEND).

The problem here is that if the input device is sending many non-response messages one after the other, the IMS message to notify the input device that the input was rejected (ACTION=STOP), or the IMS message to notify the input device that application has terminated abnormally (ACTION=ABEND), will not be read by the input device.

This is because the message will remain in a send mode until it has completed sending in all of its non-response transactions. The IMS queue space message buffer for the IMS error message will remain allocated.

In both non-response mode transaction cases (ACTION=STOP and ACTION=ABEND), the input message in the IMS message queue space message buffer will be deallocated. However the IMS error messages will not be deallocated until the input device goes to read mode.

If the input device does not issue a read function to get any output (in this case the error messages), then the IMS message queue space message buffer count will increase for the input message and the IMS error message, and decrease only for the input message and not the error message.

Using OSTARTACTION=JTSTP

The OSTARTACTION=JTSTP option functions in the same manner as the STOP option. However, JTSTP does not default to the WTO action. When a queue has reached or exceeded the specified AREA Statement threshold, the JTSTP option causes either an A7 status code to be returned to the IMS application, or one of the following IMS DFS messages to be returned to the input device:

- VTAM = DFS074 message
- APPC = DFS0777 message
- OTMA = NAK message
- BTAM = DFS074 message
- MSC = DFS1945 message

A PCB status code A7 message call indicates that the number of output segments inserted has exceeded the limit by one. Any further queue manager calls are prohibited to prevent message queue overflow.

As an example, the following AREA statement is specified:

```
AREA=(ID=AREA0001,
          PERCENT=(TOTAL=50,USED=40),
          TYPE=(ALL),
          CSTOPACTION=WTO,
          CSTARTACTION=STOP,
          OSTOPACTION=WTO,
          OSTARTACTION=JTSTP)
```

This example assumes that each message uses a single buffer and that 200 messages triggers this AREA statement.

If a single transaction or terminal (source) loops in error, or places 200 messages on a single queue (destination), a CSTARTACTION=STOP setting causes all remaining message inserts made by that same transaction or terminal (source) during the same UOW to receive an A7 status code or one of the IMS DFS messages. During that single UOW, all message inserts starting with 201 and above are rejected and the queue will not increase in size.

If the queue remains at 200 messages and a second (or more) transaction or terminal (source) places a single message or multiple messages on the queue (destination), an OSTARTACTION=JTSTP setting causes the messages to be rejected. An A7 status code or one of the IMS DFS messages is sent.

When using the JTSTP option, it is important to understand that all attempted message inserts to this destination, once the AREA statement threshold has been reached, results in these messages being rejected, regardless of whether the message comes from an application or a terminal. A possible consequence might be that an application program must back out database updates and return a message to the input device stating that the input has been rejected.

Chapter 10. Using IMS Queue Control Facility functions

IMS Queue Control Facility functions allow you to maintain the message queues and perform other IMS Queue Control Facility tasks. Functions are run by using ISPF panels or by issuing JCL job streams. Several server commands are specified with an MVS **MODIFY** command.

Topics:

IMS Queue Control Facility functions overview

IMS Queue Control Facility functions allow you to maintain the message queues and perform other IMS Queue Control Facility tasks.

Most functions have similarly named control statement keywords and JCL steps; however, some do not.

Some exit routines are replaceable because unlike IMS, which loads its user exit routines, IMS Queue Control Facility has some user exit routines link-edited into appropriate load modules.

BROWSE
> The BROWSE function browses the queues.

LOAD
> The LOAD function loads messages to the queues.

QUERY
> The QUERY function determines the age and number of messages on the queues.

RECOVER
> The RECOVER function recovers messages on the cold queue after a cold start in shared queues environments.

RECOVERAB
> The RECOVERAB function recovers messages after a cold start following an abnormal termination in nonshared-queues environments.

RECOVERDM
> The RECOVERDM function recovers messages after a cold start following a normal termination in nonshared-queues environments.

REPROCESS
> The REPROCESS function reprocess messages after an application error.

UNLOAD
> The UNLOAD function removes messages from the queues.

Additionally, IMS Queue Control Facility functions include the following logic and job steps that can help you to accomplish the following tasks:

- Control processing logic allows you to select and cancel IMS Queue Control Facility processing in shared- and nonshared-queues environments.
- Load function logic can be run as a stand-alone procedure for fallback from an IMS shared-queues environment.
- The IQCPARMS logic allows you to enter your own route codes for WTO messages.
- Job steps DFNDS, DLTDS, and SORTx run non-IMS Queue Control Facility routines.

These functions are described in the following topics, along with all control statements, JCL, and subfunctions.

The control statements, JCL, and subfunctions which are common to multiple functions are described first.

BROWSE function

The BROWSE function extracts information from messages in the queues and produces an output data set.

The BROWSE function examines the queues to determine the number and type of messages that exist on the queues.

The BROWSE function is the only function that identifies the messages that are on the cold queue in a shared-queues environment.

The BROWSE function performs no actions on the queues but it does produces reports.

You can use the BROWSE function to do the following tasks:

- Determine if the queues require maintenance. Then you can use the UNLOAD function to remove unwanted messages.
- Determine if messages exist on the queue that should be requeued by the RECOVER function.
- Determine if messages exist that contain a text string that you can specify and queue to the destination of your choice.
- Copy records for later insertion by the LOAD function.

LOAD function

The LOAD function works in both shared- and nonshared-queues environments to load messages onto the message queues and to produce a data report.

Typically you run the LOAD function for stress and regression testing.

The LOAD function can also be used for migration. The LOAD function requeues messages to the queue from the location where the messages were unloaded.

The input to the LOAD function can be the output data set from the following sources:

- BROWSE, UNLOAD, or RECOVER functions
- RECOVERAB, RECOVERDM, or REPROCESS procedures

QUERY function

The QUERY function in both shared- and nonshared-queues environment examines the age of messages on different destination queues and produces a data report.

Typically you run the QUERY function to determine which messages to remove from the queues. QUERY provides a list of all queues that have messages on them.

QUERY also reports the number of messages and the age of the oldest and the newest message. QUERY lets you determine how many messages are older than a selected age. This query can be done based on queue type (such as transaction, LTERM, MSC, serial, suspend, remote, APPC, DEADQ, and OTMA).

For all queues except for the cold queue, the message counts can be broken down further by each destination name on the queue type.

You can also enter a console command to display the destinations over the last number of minutes that are using the message queues the most.

The QUERY function is typically followed by UNLOAD, which removes unwanted messages from the queues.

The QUERY function performs no actions on the queues, but it does produce a report.

RECOVER function

Use the RECOVER function in a shared-queues environment for cold queue recovery of messages after an IMS cold start.

You can recover the unprocessed messages that were on the IMS message queue because IMS Queue Control Facility selects messages for requeuing based on criteria that you specify.

A single-step procedure called RECOVER is used to recover the messages.

RECOVERAB and RECOVERDM functions

The RECOVERAB and RECOVERDM functions recover messages after a cold start in a nonshared-queues environment.

Use the RECOVERAB and RECOVERDM functions for SLDS recovery of messages that were not processed because of an IMS cold start. IMS Queue Control Facility selects only messages for requeuing that were on the IMS message queue and were not processed by IMS.

In a nonshared-queues environment, recover messages using the procedure appropriate to the IMS termination type that occurred:

- Use the RECOVERAB function after a cold start following abnormal termination.
- Use the RECOVERDM function after a cold start following normal termination.

The RECOVERAB and RECOVERDM functions are both multistep requeuing procedures:

- Read messages from the SLDS during the SELECT step
- Analyze and cancel messages that were successfully processed in the CANCEL step of RECOVERAB
- Sort messages into their correct order for processing in the SORTx steps
- Reload the messages to the IMS online program that places them back on the message queue in the LOAD step

REPROCESS function

Use the REPROCESS function in both shared- and nonshared-queues environments to reprocess messages.

Use this function after you discover that an application program has incorrectly processed messages because of a logic error. After the program error is corrected, messages that were processed must be reprocessed.

The REPROCESS function selects previously processed messages for reloading into the message queue. An application program reads messages from the SLDS in the SELECT step, sorts messages into their correct order for reprocessing in the SORTB step, and then reloads the messages to the IMS online program that places them back on the message queue in the LOAD step.

UNLOAD function

The UNLOAD function works in both shared- and nonshared-queues environments to delete messages from the IMS message queues or to remove messages for requeuing.

Typically you run the UNLOAD function to perform one of the following tasks:

- Support queue maintenance for deleting unwanted messages
- Unload messages to a data set, for possible later requeuing when you are doing stress, regression, or application testing
- Unload messages for fallback or migration

The UNLOAD function lets you remove messages from the queues, either for permanent deletion or for subsequent requeuing by using the LOAD function.

Automatically locating checkpoints

You can automatically locate checkpoints to recover and reprocess messages in a nonshared-queues environment. The checkpoint locator applies to the RECOVERAB, RECOVERDM, and REPROCESS functions.

The RECOVERAB-DBRC function provides a list of logs, starting with the log that contains the first record of the checkpoint preceding the IMS crash (DUMPQ, PURGE, or SNAP). All relevant archived logs that follow are included in the list. IMS Queue Control Facility uses the list of logs to rebuild the queue.

The RECOVERDM-DBRC function provides a list of logs, starting with the log that contains the first record of the checkpoint preceding the DUMPQ or PURGE statement for normal IMS termination. All archived logs that contain these checkpoint records are included. IMS Queue Control Facility uses the list of logs to rebuild the queue.

The REPROCESS-DBRC function provides a list of logs, starting with the log that contains the checkpoint preceding the DUMPQ, PURGE, or SNAPQ statement set in the control cards. All relevant archived logs that follow are included. IMS Queue Control Facility uses the list of logs to rebuild the queue.

You can use either of the following two techniques to automatically locate checkpoints:

- Prepare JCL for the SELECT logic (and other steps in the function) by using the RECON data sets and skeletal JCL, and then submit this JCL.

  This technique uses a IMS Queue Control Facility batch logic.

- Make changes to the SELECT processing logic. Parameters are available in the CHKPT control card.

  In addition, add or modify DD statements in the SELECT step and submit the job.

  With this technique, the RECON data sets are read and the logs are dynamically allocated. The entire process completes in a single run.

Common JCL for IMS Queue Control Facility job steps

This topic describes the common JCL EXEC statement PARM field that you can specify to run all of the IMS Queue Control Facility job steps.

The other job steps include the steps in the following list that are described in their appropriate topics:

- BROWSE (shared and nonshared queues)
- CANCEL (nonshared queues)
- DFNDS (nonshared queues)
- DLTDS (nonshared queues)
- LOAD (shared and nonshared queues)
- QUERY (shared and nonshared queues)
- RECOVER (shared queues)
- SELECT (shared and nonshared queues)
- SORTA (nonshared queues)
- SORTB (shared and nonshared queues)
- UNLOAD (shared and nonshared queues)
- IQCABAT0 (shared and nonshared queues)

⚠️ **Attention:** Before trying to run these job steps, you need to understand the purpose of each IMS Queue Control Facility function and how each function relates to other functions.

⚠️ **Attention:** This discussion of JCL statements for IMS Queue Control Facility job steps is intended only as a guide. Specific JCL requirements vary for different environments, the different procedures that are used, and the varying circumstances that require requeuing of IMS messages.

EXEC statement PARM field

The BPECFG, IMSPLEX, and QCFIMS PARM keywords are coded on the EXEC statement of your JCL job streams.

This topic describes the BPECFG, IMSPLEX, and QCFIMS PARM keywords.

⚠️ **Attention:** The IMS SCI address space must be active on the z/OS images on which IMS Queue Control Facility batch jobs will run.

Example

Code the BPECFG, IMSPLEX, and QCFIMS keywords in the PARM= field of the EXEC JCL statement of your JCL stream.

```
//QCFBATCH EXEC  PGM=IQCINI0$,
//       PARM='BPEINIT=IQCBINI0,BPECFG=IQCIVPCF,IMSPLEX=IPLEX,QCFIMS=IMS1'
```

Keyword reference

Code the BPECFG, IMSPLEX, and QCFIMS PARM keywords in the following format:

BPECFG=*bpe_config_proclib_mbr*
The optional keyword, BPECFG specifies an 8-character name for your BPE configuration PROCLIB member.

IMSPLEX(*name*)
The required keyword, IMSPLEX specifies a 1- to 5-character CSLPLEX name to which IMS Queue Control Facility communicates. This IMSPLEX name is used for communications between the IMS Queue Control Facility batch jobs and the IMS Queue Control Facility extensions.

Both the IMS Queue Control Facility batch jobs and the IMS subsystems that manipulate IMS message queues must use the same SCI IMSPLEX name.

The *name* parameter must be the same name as is specified in the SCI initialization PROCLIB member, IMSPLEX(NAME=*name*).

Note: QCF will have a member name of *batchid* that is the PROC name of the QCF batch job.

QCFIMS(*imsid*)
The required keyword, QCFIMS specifies a 1- to 4-character IMS name. IMS Queue Control Facility concatenates this ID to the prefix, IQC# to develop the full SCI member name for the target IMS Queue Control Facility extensions.

For example, if QCFIMS(IMSA) were coded, the target IMS Queue Control Facility extensions full SCI member name would be, IQC#IMSA.

This SCI member is used for communications between the IMS Queue Control Facility batch job and the IMS Queue Control Facility extensions.

Common control statements

Several of the IMS Queue Control Facility control statements are common to all of the IMS Queue Control Facility functions.

These topics describe the control statements that are common to the IMS Queue Control Facility functions.

Before using these functions, read the topics that describe the functions to understand them and how they relate to the other functions.

The SELECT, LOAD, and CANCEL functions are steps in the IMS Queue Control Facility recover and reprocess procedures. QCF control statements can be in uppercase, lowercase, or mixed case.

Attention: The FUNCTION, INCLUDE and SELECT control statements have different subsets of parameters for different IMS Queue Control Facility functions. The valid formats of these statements are shown for every function for which they apply. All other control statements have the same sets of parameters that are shown in the following subtopics for all of the IMS Queue Control Facility functions for which they apply.

Topics:

- "Control processing control statements" on page 122
- "EXCLUDE statement" on page 125
- "FUNCTION statement" on page 130
- "INCLUDE statement" on page 135
- "NOWTOMSG statement" on page 143
- "SELECT statement" on page 143
- "TITLE statement" on page 146
- "WTOMSG statement" on page 147

Control processing control statements

Each of the IMS Queue Control Facility functions processing can be controlled by control statements.

These topics provide the details about each of the control statements that IMS Queue Control Facility control processing uses to perform the functions.

Functions of control processing

IMS Queue Control Facility control processing logic processes each control statement in the QCFIN input stream.

The IMS Queue Control Facility control processing logic is the main controller for shared and nonshared queues. IMS Queue Control Facility control processing logic processes each control statement in the QCFIN input stream by performing the following tasks:

- Interprets each statement
- Validates the data
- Builds control blocks
- Issues error messages
- Runs the appropriate logic

When the QCFIN stream has been exhausted and no errors have occurred, IMS Queue Control Facility control processing calls the appropriate processing logic as specified on the FUNCTION control statement.

The control statements that IMS Queue Control Facility control processing recognizes follow. IMS Queue Control Facility control processing processes only the FUNCTION and NOWTOMSG control statements; all of the other statements are only validated and interpreted.

```
* comment
CHNGDEST
CURMSGTIME
EXCLUDE
FUNCTION
INCLUDE
MAXMSGCT
MAXWAIT
NOWTOMSG
PACING
```

SELECT
    TITLE
    TOIPARMS
    WAIT
    WTOMSG

IMS Queue Control Facility control processing does not determine whether a control statement is appropriate for the function that is selected. For example, if CHNGDEST is specified and the function is not LOAD, no warning message is issued. Nor does IMS Queue Control Facility control processing check whether the appropriate DD statements are in the JOB stream; that checking is left to the individual processing logic.

The FUNCTION statement is required. FUNCTION specifies the program function that is to be executed. This is how IMS Queue Control Facility control processing knows which processing logic should be called.

The other control statements (SELECT, INCLUDE, and EXCLUDE) might not be applicable or coded depending on which function and operands are selected. Refer to the explanation of each function for details.

Because IMS Queue Control Facility control processing interprets the entire QCFIN stream before IMS Queue Control Facility control processing calls the selected processing logic, control statements can be in any order; and the FUNCTION statement need not be the first statement in the QCFIN stream.

Also, control statements for IMS Queue Control Facility control processing are not positional and many of them can be abbreviated. However, only the QUEUETYPE keyword allows multiple parameters on the same statement; all other keywords still require multiple records for multiple parameters (see the appropriate topics for the individual statements).

Control statement abbreviations

You can use abbreviations for some of the control statement keywords.

The IMS Queue Control Facility control processing keyword abbreviations table shows a list of all of the keyword abbreviations that are allowed by the IMS Queue Control Facility control processing interpreter:

Table 13. IMS Queue Control Facility control processing keyword abbreviations

Keyword	Abbreviation
ACTION	AC
APPC	AP
AREA	AR
BROWSE	BR
COLD	CO
CHNGDEST	CD
DEBUG	DB
DELETE	DE
DESTINATION	DE, DST
ENDTIME	ET
EXCLUDE	EX
FUNCTION	FU
GLOBAL	GL
IMSIDS	II

Table 13. IMS Queue Control Facility control processing keyword abbreviations (continued)

Keyword	Abbreviation
IMSDSTID	IDI
IMSSRCID	ISI
INCLUDE	IN
LOAD	LO
LOCAL	LO
LTERM	LT
LUNAME	LUN
LUNAMEDST	LUND
LUNAMESRC	LUNS
LUSIDENAME	LUSN
LUTPNAME	LUTP
MAXMSGCT	MM
NOCONVMSG	NC
NOSYSMSG	NS
NOWTOMSG	NW
OTMA	OT
OVERFLOW	OV
PACING	PA
PERCENTDST	PD
QUERY	QU
QUEUETYPE	QT
READ	RE
RECCOUNTDST	RD
RECOVER	RE
REMOTE	RE
REPORTONLY	RO
ROUTECODE	RC
SELECT	SE
SERIAL	SE
SOURCE	SO, SRC
STARTTIME	ST
SUSPEND	SU
TMEMBER	TM
TMEMBERDST	TMD

Keyword	Abbreviation
TMEMBERSRC	TMS
TPIPE	TP
TRANCODE	TR
TRANSACTION	TR
TYPE	TY
UNLOAD	UN
UNLOCK	UN
UOWORGID	UOI
UOWORGTK	UOT
UOWPROID	UPI
UOWPROTK	UPT
VTAM	VT

Comment (*) statement

You can use the comment statement in your QCFIN control statement stream to add comments about your settings.

You can insert comment (*) statements at any point in the QCFIN control statement stream, including between continuation statements.

Syntax diagram for Comment statement

```
▶▶─ * ──────────────────────────────────▶◀
      └─ user comments from columns 2 through 80 ─┘
```
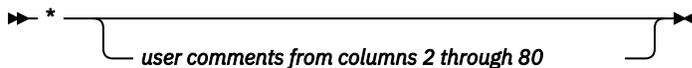
EXCLUDE statement

You can use EXCLUDE statements to exclude certain messages or parts of messages from your selections.

EXCLUDE statements are optional, and you can use any number of statements (there is no maximum limit).

The purpose of the EXCLUDE statement is to exclude a specific part of the set of messages that would otherwise be selected from the selection process. For a message to be excluded, the message must satisfy all of the specified criteria for at least one EXCLUDE statement.

Guidelines for coding the EXCLUDE statement

- The EXCLUDE statement, and its continuation lines, can contain one or all of the operands in any order.
- Duplicate operands for the same EXCLUDE statement are not allowed.
- Only one operand can be specified per line.
- Additional operands must be specified on continuation lines; this is done by following the operand with a comma and beginning the next operand in column 10 of the next line.

Example of EXCLUDE statement

```
EXCLUDE DESTINATION=C*,
        DESTYPE=LTERM,
        IMSDSTID=IMS1
EXCLUDE DESTYPE=(APPC,TRANS)
```

Syntax diagram for EXCLUDE statement
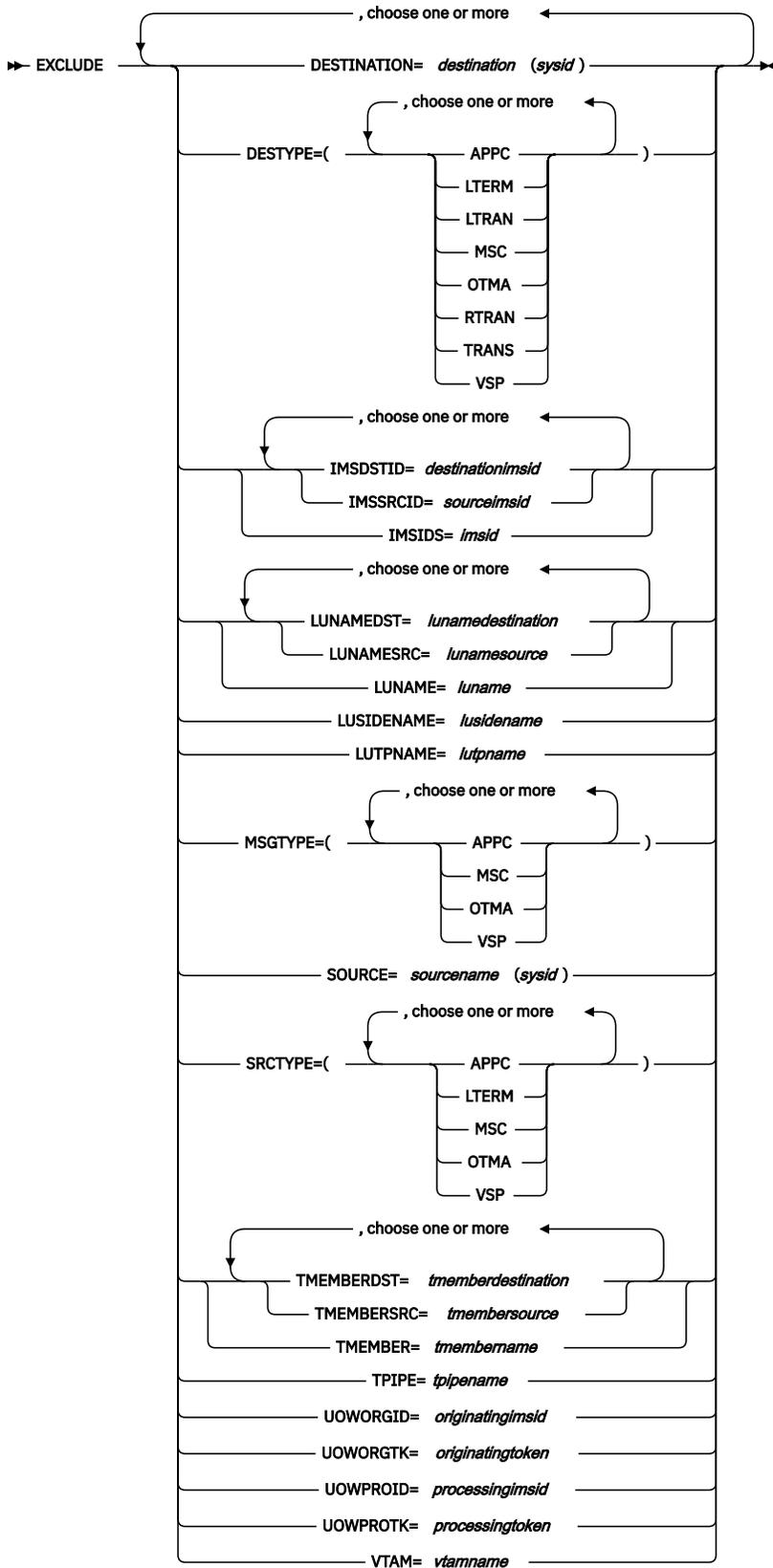
Parameter reference for EXCLUDE statement

The variable fields are defined as follows:

APPC

Exclude from the selection process certain types of APPC messages; the operand in which APPC is specified determines which types of messages are excluded, as follows:

DESTYPE

Exclude APPC messages based on their destination.

SRCTYPE

Exclude APPC messages based on their source.

MSGTYPE

Exclude APPC messages based on their source and destination. In this case, if either source or destination applies the message is selected.

DESTINATION=*destination(sysid)*

A 1- to 8-character transaction code or logical terminal name. Groups of names that begin with the same characters can be specified generically by using an asterisk (*) after the characters that are the same.

(sysid)

One to five decimal digits in parentheses, in the form (*n*) to (*nnnnn*) where*n* is a decimal number from 1 to 2036; the number is a system ID for Multiple Systems Coupling transactions.

(*sysid*) is optional and can be used alone without *destination* to specify a destination system without specifying a particular transaction code or logical terminal name.

IMSDSTID=*destinationimsid*

A 1- to 8-character name of a destination IMS system. All messages destined to this IMS system will be excluded. Groups of names that begin with the same characters can be specified generically by using an asterisk (*) after the characters that are the same.

IMSIDS=*imsid*

A 1- to 8-character name of an IMS system. All messages coming from or destined to this IMS system will be excluded. Groups of names that begin with the same characters can be specified generically by using an asterisk (*) after the characters that are the same.

IMSSRCID=*sourceimsid*

A 1- to 8-character name of a source IMS system. All messages that originated at this IMS system will be excluded. Groups of names that begin with the same characters can be specified generically by using an asterisk (*) after the characters that are the same.

LTERM

Exclude from the selection process certain types of LTERM messages; the operand in which LTERM is specified determines which types of messages are excluded, as follows:

DESTYPE

Exclude LTERM messages based on their destination.

SRCTYPE

Exclude LTERM messages based on their source.

LTRAN

Exclude from the selection process local transaction messages based on their destination (DESTYPE).

LUNAME=*luname*

A 1- to 8-character field that is the LU name for LU 6.2 transactions. Groups of names that begin with the same characters can be specified generically by using an asterisk (*) after the characters that are the same. An asterisk (*) as the first character will exclude all LU 6.2 transactions.

LUNAMEDST=*lunamedestination*

A 1- to 8-character field for selecting LU 6.2 destination transactions by luname where the destination is to an LU 6.2 device. Groups of names that begin with the same characters can be specified generically by using an asterisk (*) after the characters that are the same. An asterisk (*) as the first character will exclude all LU 6.2 destination transactions.

LUNAMESRC=*lunamesource*
A 1- to 8-character field for selecting LU 6.2 source transactions by luname where the destination is not to an LU 6.2 device. Groups of names that begin with the same characters can be specified generically by using an asterisk (*) after the characters that are the same. An asterisk (*) as the first character will exclude all LU 6.2 source transactions.

LUSIDENAME=*lusidename*
A 1- to 8-character field that is the LU sidename for LU 6.2 transactions in which the destination is an LU 6.2 device. Groups of names that begin with the same characters can be specified generically by using an asterisk (*) after the characters that are the same. An asterisk (*) as the first character will exclude all LU sidename LU 6.2 transactions.

LUTPNAME=*lutpname*
A 1- to 16-character field that is the tpname for LU 6.2 transactions. Groups of names that begin with the same characters can be specified generically by using an asterisk (*) after the characters that are the same. An asterisk (*) as the first character will exclude all LU 6.2 tpnames.

MSC
Exclude from the selection process certain types of MSC messages; the operand in which MSC is specified determines which types of messages are excluded, as follows:

DESTYPE
Exclude messages destined to MSC links (in other words, destined to MSNAMES and remote transactions).

SRCTYPE
Exclude messages whose source is an MSC link MSNAME.

MSGTYPE
Exclude MSC messages based on their source and destination. In this case, if either source or destination applies the message is selected.

OTMA
Exclude from the selection process certain types of OTMA messages; the operand in which OTMA is specified determines which types of messages are excluded, as follows:

DESTYPE
Exclude OTMA messages based on their destination.

SRCTYPE
Exclude OTMA messages based on their source.

MSGTYPE
Exclude OTMA messages based on their source and destination. In this case, if either source or destination applies the message is selected.

RTRAN
Exclude from the selection process remote transaction messages based on their destination (DESTYPE).

SOURCE=*sourcename(sysid)*
A 1- to 8-character logical terminal name. Groups of names that begin with the same characters can be specified generically by using an asterisk (*) after the characters that are the same.

(sysid)
One to five decimal digits in parentheses, in the form (*n*) to (*nnnnn*) where *n* is a decimal number from 1 to 2036; the number is a system ID for Multiple Systems Coupling transactions.

(*sysid*) is optional and can be used without *sourcename* to specify a source system without specifying a particular logical terminal name.

TMEMBER=*tmembername*
A 1- to 16-character field that is the TMEMBER name for OTMA transactions. Groups of names that begin with the same characters can be specified generically by using an asterisk (*) after the characters that are the same. An asterisk (*) as the first character will exclude all OTMA transactions.

TMEMBERDST=*tmemberdestination*
A 1- to 16-character field for selecting OTMA destination transactions by TMEMBER name where the destination is to OTMA. Groups of names that begin with the same characters can be specified generically by using an asterisk (*) after the characters that are the same. An asterisk (*) as the first character will exclude all OTMA destination transactions.

TMEMBERSRC=*tmembersource*
A 1- to 16-character field for selecting OTMA source transactions by TMEMBER name where the destination is not to OTMA. Groups of names that begin with the same characters can be specified generically by using an asterisk (*) after the characters that are the same. An asterisk (*) as the first character will exclude all OTMA source transactions.

TPIPE=*tpipename*
A 1- to 8-character field that is the TPIPE name for OTMA transactions. Groups of names that begin with the same characters can be specified generically by using an asterisk (*) after the characters that are the same. An asterisk (*) as the first character will exclude all OTMA tpipes.

TRANS
Exclude from the selection process transaction messages based on their destination (DESTYPE).

VSP
Exclude from the selection process ETO and LU 6.1 VTAM subpool messages; the operand in which VSP is specified determines which types of messages are excluded, as follows:

DESTYPE
Exclude VSP messages based on their destination.

SRCTYPE
Exclude VSP messages based on their source.

MSGTYPE
Exclude VSP messages based on their source and destination. In this case, if either source or destination applies the message is selected.

UOWORGID=*originatingimsid*
A 1- to 8-character field that is the exact name of an IMS system where a message originated. All messages with this system ID as the origination system will be excluded.

UOWORGTK=*originatingtoken*
A 16-character hexadecimal field specifying an originating token (time stamp) of a message. All messages with this originating token will be excluded.

UOWPROID=*processingimsid*
A 1- to 8-character field that is the name of an IMS system where a message was or will be processed. All messages with this system ID as the processing system will be excluded. Groups of names that begin with the same characters can be specified generically by using an asterisk (*) after the characters that are the same.

UOWPROTK=*processingtoken*
A 16-character hexadecimal field specifying a processing token (time stamp) of a message. All messages with this processing token will be excluded.

VTAM=*vtamname*
A 1- to 8-character field that is the VTAM name for messages that are not LU 6.2 or OTMA. Groups of names that begin with the same characters can be specified generically by using an asterisk (*) after the characters that are the same.

FUNCTION statement

The FUNCTION statement specifies the program function and it is required for IMS Queue Control Facility control processing.

The FUNCTION statement specifies global parameters.

Guidelines for coding the FUNCTION statement

- The FUNCTION statement should specify the program function.
- The FUNCTION, and its continuation statements, can contain no operands or any number of operands in any order.
- Duplicate operands for the same FUNCTION statement are not allowed.
- One or more operands can be specified per FUNCTION statement, separated by commas.
- Additional operands can be specified on continuation lines by following the previous operand with a comma and beginning the next operand in column 10 of the next line.

Example 1: FUNCTION BROWSE

```
FUNCTION BROWSE AGE=7,
         AREA=(GLOBAL,LOCAL)
```

Example 2: FUNCTION UNLOAD

```
FUNCTION UNLOAD,
         AREA=(ALL)
```

Example 3: FUNCTION QUERY

```
FUNCTION QUERY QUEUETYPE=(ALL),ACTION=(READ)
```

In this example, the selection is first made based on QUERY parameters, SELECT, INCLUDE, and EXCLUDE statements, if any. Then, destinations are sorted by numbers of messages in descending order.

SELECT, INCLUDE, or EXCLUDE, or any combination of the three can follow the QUERY function. And the FUNCTION statement is required if you use SELECT.

Syntax diagram for FUNCTION statement

```
►►─ FUNCTION ─┬── BROWSE ──────┬─────────────────────────────────────────────►
              ├── LOAD ────────┤
              ├── QUERY ───────┤      ┌──── , choose one or more ◄───┐
              ├── RECOVER ─────┤      │                              │
              ├── RECOVERAB ───┤   ┌─ AREA=( ──┬──── ALL ────────┬── ), ─┐
              ├── RECOVERDM ───┤   │           ├──── GLOBAL ──────┤
              ├── REPROCESS ───┤               ├──── LOCAL ───────┤
              └── UNLOAD ──────┘               ├──── OVERFLOW ────┤
                                               └──── QSN ─────────┘

   ┌─────────────────────────────────────────────────►
   └─ ACTION=( ──┬──── READ ──────┬── ), ─┐
                 ├─ REPORTONLY ───┤
                 ├─ DELETE ───────┤
                 └─ UNLOCK ───────┘

   ┌───────────────────────────────────────────────────────────►
   │                              AGEUNITS=DAYS, [1]
   ├─ AGE= nnnnnnnn ,[1] ─┬── AGEUNITS=DAYS, [1] ──┬─┐
   │                      └── AGEUNITS=SECONDS, [1] ─┘
   ├─ AGE= nnn ,[2] ─────────────────────────────────┐
   └─ MAXAGE= nnn ,[2] ──────────────────────────────┘

   ┌─────────────────────────────────────────────── RCT= y, ─┐
   │          ┌──── , choose one or more ◄───┐
   │          │          COLD                │
   └─ QUEUETYPE=( ──┬──── COLD ──────┬── ), ─┐
                    ├──── ALL ───────┤
                    ├──── APPC ──────┤
                    ├──── LONG ──────┤
                    ├──── LTERM ─────┤
                    ├──── OTMA ──────┤
                    ├──── REMOTE ────┤
                    ├──── SHORT ─────┤
                    ├──── SERIAL ────┤
                    ├──── SUSPEND ───┤
                    └── TRANSACTION ─┘

   ┌──────────────────────┬── TOPAM= nn, ──┬──►◄
   └─ SBMPxx, [3] ─┐      ├── TOPLA= nn, ──┤
                          ├── TOPLM= nn, ──┤
                          └── TOPSM= nn ───┘
```

Notes:

[1] Applicable only to BROWSE, QUERY, RECOVER, and UNLOAD functions.

[2] Applicable only to RECOVERAB, RECOVERDM, and REPROCESS functions.

[3] Required for all jobs that are started because a threshold was crossed.

Parameter reference for FUNCTION statement

If you do not specify a value for AREA=, AREA= defaults to one of the following values:

- LOCAL for nonshared queues
- GLOBAL, LOCAL, and OVERFLOW for shared queues

The variable fields are defined as follows:

AGE=
   Specify the minimum age of messages in a queue that you want IMS Queue Control Facility to process in the subsequent operation.

   nnnnnnnn
      Specify the minimum age in days or in seconds.

      A 1- to 8-digit parameter is applicable to BROWSE, QUERY, RECOVER, and UNLOAD functions.

      Valid values for *nnnnnnnn* are from 0 to 365 if AGEUNITS=DAYS (the unit is days) or from 0 to 31536000 if AGEUNITS=SECONDS (the unit is seconds).

      For example, to find messages that have been in a particular queue for more than 7 days, specify AGE=7 and AGEUNITS=DAYS.

      To find messages that have been in a particular queue for more than 120 seconds, specify AGE=120 and AGEUNITS=SECONDS.

   nnn
      Specify the minimum age in days. A 1- to 3-digit parameter is applicable to RECOVERAB, RECOVERDM, and REPROCESS functions.

      For example, to find messages that have been in a particular queue for more than 7 days, specify AGE=7.

   Definition: A *day* is a 24-hour period prior to the current time. For example, if the time is 11:53:24 on Wednesday, AGE=1 applies to any message that was inserted into the shared queues prior to 11:53:24 Tuesday.

AGEUNITS=DAYS | SECONDS
   Specify whether the AGE=*nnnnnnnn* parameter is specified in days or seconds. The default value is DAYS.

   AGEUNITS is applicable to BROWSE, QUERY, RECOVER, and UNLOAD functions.

   You can abbreviate DAYS and SECONDS as DAY and SEC, respectively.

MAXAGE=*nnn*
   Specify the maximum age, in days, of messages in a queue that you want IMS Queue Control Facility to process in the subsequent operation.

   For example, to find messages that have been in a particular queue for less than 7 days, specify MAXAGE=7.

   MAXAGE is applicable to RECOVERAB, RECOVERDM, and REPROCESS functions.

   Definition: A *day* is a 24-hour period prior to the current time. For example, if the time is 11:53:24 on Wednesday, AGE=1 applies to any message that was inserted into the shared queues prior to 11:53:24 Tuesday.

ALL
   All queues are to be retrieved.

APPC
   The APPC queue is to be retrieved.

COLD

The cold queue is to be retrieved (shared-queues environment).

DELETE

Delete all messages that meet the selection criteria from the cold queue.

GLOBAL

The global (primary) queue is to be retrieved (shared-queues environment).

LOCAL

The local queue is to be retrieved (shared- and nonshared-queues environment).

LONG

The long messages queue is to be retrieved.

LTERM

The LTERM queue is to be retrieved.

OTMA

The OTMA queue is to be retrieved.

OVERFLOW

The overflow queue is to be retrieved (shared-queues environment).

QSN

The QSN queue is to be retrieved (nonshared-queues environment).

RCT

Enter RCT=Y to get a count of the secondary messages.

READ

All portions of the messages meeting the selection criteria are to be retrieved.

REMOTE

The remote transaction queue is to be retrieved.

REPORTONLY

Only the header portion of messages meeting the selection criteria is to be retrieved from the shared queue. This improves performance of the operation, as minimal data is transferred between IMS and QCF.

If you specify ACTION=(REPORTONLY) on a FUNCTION or SELECT statement when using the QCF UNLOAD job, messages will be unloaded off the queues and there will be no backup of the unloaded messages.

SBMP*xx*

The valid values are SBMPAB, SBMPBC, SBMPDC, SBMPUT. You must specify the appropriate value for the threshold that was crossed. For example, if threshold A was crossed, you must specify SBMPAB. You must specify one of these values for all jobs that are started because a threshold was crossed.

SERIAL

The serial transaction queue is to be retrieved.

> **SUSPEND**
>
> The suspended transaction queue is to be retrieved.
>
> **TRANSACTION**
>
> The transaction ready queue is to be retrieved.

SHORT

The short messages queue is to be retrieved.

TOPLASTACTIVWTO (TOPLA)

Triggers WTO for the top *nn* last active destinations, where *nn* is any selected number up to 99. This parameter is valid with the QUERY function.

Restriction: TOPLASTACTIVWTO / TOPLA is mutually exclusive with TOPALLMSGWTO / TOPAM, TOPLONGMSGWTO / TOPLM, or TOPSHORTMSGWTO / TOPSM. Specifying more than one of these parameters simultaneously results in an error condition that terminates processing.

TOPALLMSGWTO (TOPAM)

Triggers WTO for the top *nn* destinations with the most messages, where *nn* is any selected number up to 99. This parameter is valid with the QUERY function.

Restriction: TOPALLMSGWTO / TOPAM is mutually exclusive with TOPLASTACTIVWTO / TOPLA, TOPLONGMSGWTO / TOPLM, or TOPSHORTMSGWTO / TOPSM. Specifying more than one of these parameters simultaneously results in an error condition that terminates processing.

TOPLONGMSGWTO (TOPLM)

Triggers WTO for the top *nn* destinations using the long message queue, where *nnnn* is any selected number up to 99 (nonshared queues, only). This parameter is valid with the QUERY function.

Restriction: TOPLONGMSGWTO / TOPLM is mutually exclusive with TOPLASTACTIVWTO / TOPLA, TOPALLMSGWTO / TOPAM, or TOPSHORTMSGWTO / TOPSM. Specifying more than one of these parameters simultaneously results in an error condition that terminates processing.

TOPSHORTMSGWTO (TOPSM)

Triggers WTO for the top *nn* destinations using the short message queue, where *nn* is any selected number up to 99 (nonshared queues, only). This parameter is valid with the QUERY function.

Restriction: TOPSHORTMSGWTO / TOPSM is mutually exclusive with TOPLONGMSGWTO / TOPLM, TOPLASTACTIVWTO / TOPLA, or TOPALLMSGWTO / TOPAM. Specifying more than one of these parameters simultaneously results in an error condition that terminates processing.

UNLOCK

All messages that meet the selection criteria are requeued from the cold queue.

INCLUDE statement

Use the INCLUDE statement to include certain types of messages and parts of messages in your selections.

INCLUDE statements are optional, and you can use any number of statements (there is no maximum limit).

If you use any INCLUDE statements, a message must satisfy all of the specified criteria for at least one INCLUDE statement in order to be selected.

Guidelines for coding the INCLUDE statement

- The INCLUDE statement, and its continuation lines, can contain one or all of the operands in any order.
- Duplicate operands for the same INCLUDE statement are not allowed.
- Only one operand can be specified per line.
- Additional operands must be specified on continuation lines by following the operand with a comma, and beginning the next operand in column 10 of the next line.

Example of INCLUDE statement

```
INCLUDE DESTINATION=C*,
        DESTYPE=LTERM,
        IMSDSTID=IMS1
```

Syntax diagram for INCLUDE statement

```
                          , choose one or more
                    ┌─────────◄─────────┐
►►─ INCLUDE ─┬──────┴─ DESTINATION= destination (sysid) ─┴──────►◄

                              , choose one or more
                          ┌─────────◄─────────┐
             ├─ DESTYPE=( ─┴─┬─ APPC ──┬─ ) ──┤
                             ├─ LTERM ─┤
                             ├─ LTRAN ─┤
                             ├─ MSC ───┤
                             ├─ OTMA ──┤
                             ├─ RTRAN ─┤
                             ├─ TRANS ─┤
                             └─ VSP ───┘

                          , choose one or more
                      ┌─────────◄─────────┐
             ├────────┴─┬─ IMSDSTID= destinationimsid ─┬──┤
                        ├─ IMSSRCID= sourceimsid ──────┤
                        └─ IMSIDS= imsid ──────────────┘

                          , choose one or more
                      ┌─────────◄─────────┐
             ├────────┴─┬─ LUNAMEDST= lunamedestination ─┬──┤
                        ├─ LUNAMESRC= lunamesource ───────┤
                        └─ LUNAME= luname ────────────────┘
             ├─ LUSIDENAME= lusidename ─┤
             ├─ LUTPNAME= lutpname ─┤

                              , choose one or more
                          ┌─────────◄─────────┐
             ├─ MSGTYPE=( ─┴─┬─ APPC ─┬─ ) ──┤
                             ├─ MSC ──┤
                             ├─ OTMA ─┤
                             └─ VSP ──┘
             ├─ PERCENTDST= percentdestination ─┤
             ├─ RECCOUNTDST= reccountdestination ─┤
             ├─ SOURCE= sourcename (sysid) ─┤

                              , choose one or more
                          ┌─────────◄─────────┐
             ├─ SRCTYPE=( ─┴─┬─ APPC ─┬─ ) ──┤
                             ├─ LTERM ─┤
                             ├─ MSC ──┤
                             ├─ OTMA ─┤
                             └─ VSP ──┘

                          , choose one or more
                      ┌─────────◄─────────┐
             ├────────┴─┬─ TMEMBERDST= tmemberdestination ─┬──┤
                        ├─ TMEMBERSRC= tmembersource ───────┤
                        └─ TMEMBER= tmembername ────────────┘
             ├─ TPIPE= tpipename ─┤
             ├─ UOWORGID= originatingimsid ─┤
             ├─ UOWORGTK= originatingtoken ─┤
             ├─ UOWPROID= processingimsid ─┤
             ├─ UOWPROTK= processingtoken ─┤
             └─ VTAM= vtamname ─┘
```

Parameter reference for INCLUDE statement

The variable fields are defined as follows:

APPC

Include in the selection process certain types of APPC messages; the operand in which APPC is specified determines which types of messages are included, as follows:

DESTYPE

Include APPC messages based on their destination.

SRCTYPE

Include APPC messages based on their source.

MSGTYPE

Include APPC messages based on their source and destination. In this case, if either source or destination applies the message is selected.

DESTINATION=*destination*(*sysid*)

A 1- to 8-character transaction code or logical terminal name. Groups of names that begin with the same characters can be specified generically by using an asterisk (*) after the characters that are the same.

(*sysid*)

One to five decimal digits in parentheses, in the form (*n*) to (*nnnnn*) where *n* is a decimal number from 1 to 2036; the number is a system ID for Multiple Systems Coupling transactions.

(*sysid*) is optional and can be used alone without *destination* to specify a destination system without specifying a particular transaction code or logical terminal name.

IMSDSTID=*destinationimsid*

A 1- to 8-character name of a destination IMS system. All messages destined to this IMS system will be included. Groups of names that begin with the same characters can be specified generically by using an asterisk (*) after the characters that are the same.

IMSIDS=*imsid*

A 1- to 8-character name of an IMS system. All messages coming from or destined to this IMS system will be included. Groups of names that begin with the same characters can be specified generically by using an asterisk (*) after the characters that are the same.

IMSSRCID=*sourceimsid*

A 1- to 8-character name of a source IMS system. All messages that originated at this IMS system will be included. Groups of names that begin with the same characters can be specified generically by using an asterisk (*) after the characters that are the same.

LTERM

Include in the selection process certain types of LTERM messages; the operand in which LTERM is specified determines which types of messages are included, as follows:

DESTYPE

Include LTERM messages based on their destination.

SRCTYPE

Include LTERM messages based on their source.

LTRAN

Include in the selection process local transaction messages based on their destination (DESTYPE).

LUNAME=*luname*

A 1- to 8-character field that is the LU name for LU 6.2 transactions. Groups of names that begin with the same characters can be specified generically by using an asterisk (*) after the characters that are the same. An asterisk (*) as the first character will include all LU 6.2 transactions.

LUNAMEDST=*lunamedestination*

A 1- to 8-character field for selecting LU 6.2 destination transactions by luname where the destination is to an LU 6.2 device. Groups of names that begin with the same characters can be specified generically by using an asterisk (*) after the characters that are the same. An asterisk (*) as the first character will include all LU 6.2 destination transactions.

LUNAMESRC=*lunamesource*

A 1- to 8-character field for selecting LU 6.2 source transactions by luname where the destination is not to an LU 6.2 device. Groups of names that begin with the same characters can be specified generically by using an asterisk (*) after the characters that are the same. An asterisk (*) as the first character will include all LU 6.2 source transactions.

LUSIDENAME=*lusidename*

A 1- to 8-character field that is the LU sidename for LU 6.2 transactions in which the destination is an LU 6.2 device. Groups of names that begin with the same characters can be specified generically by using an asterisk (*) after the characters that are the same. An asterisk (*) as the first character will include all LU sidename LU 6.2 transactions.

LUTPNAME=*lutpname*

A 1- to 16-character field that is the tpname for LU 6.2 transactions. Groups of names that begin with the same characters can be specified generically by using an asterisk (*) after the characters that are the same. An asterisk (*) as the first character will include all LU 6.2 tpnames.

MSC

Include in the selection process certain types of MSC messages; the operand in which MSC is specified determines which types of messages are included, as follows:

DESTYPE

Include messages destined to MSC links (in other words, destined to MSNAMES and remote transactions).

SRCTYPE

Include messages whose source is an MSC link MSNAME.

MSGTYPE

Include MSC messages based on their source and destination. In this case, if either source or destination applies the message is selected.

OTMA

Include in the selection process certain types of OTMA messages; the operand in which OTMA is specified determines which types of messages are included, as follows:

DESTYPE

Include OTMA messages based on their destination.

SRCTYPE

Include OTMA messages based on their source.

MSGTYPE

Include OTMA messages based on their source and destination. In this case, if either source or destination applies the message is selected.

PERCENTDST=*percentdestination*

Include in the selection process all messages queued to the same destination (such as an LTERM) that collectively equal or exceed the specified percentage of space on the short or long message queue data sets (SMSGQ or LMSGQ).

Restrictions:

• PERCENTDST is only supported in the IMS nonshared-queues environment.

• Valid destination-specific operands are:

  – DESTYPE
  – DESTINATION
  – LUNAMEDST
  – LUSIDENAME
  – TMEMBERDST

An example of using PERCENTDST follows. In the figure, the following usage information is true:

- TSUED01 is using 2 percent of the large message queue with 216 primary messages (or records) queued.
- DESRZA70.A7CICHBT DFSASYNC is using 1 percent of the large message queue with 192 primary messages (or records) queued.
- LTERM T0910122 is using 1 percent of the large message queue with 144 primary messages (or records) queued.
- CLIENT1 T3270LC is using 1 percent of the large message queue with 96 primary messages (or records) and 48 secondary messages (or records) queued.

```
  View  Table_Actions  Help
 -----------------------------------------------------------------------------
                        Messages Destinations (Detail)        Row 1 to 4 of 4
 Command ===>                                            Scroll ===> PAGE

 Select a row action or press END to exit
                                                   APAR . . : PH19814 2020/01/
 Server . . : QCFSERVE                             JDTE . . : 2020.016
 IMS ID . . : IMS1                                 TIME . . : 13:47:55
 QCF Func . : DETAIL                               DATE . . : 2020/01/16
 MSGQs  . . : ALL
 Row actions: C - Copy  D - Delete  X - Copy/Delete  L - List  U - Unlock


                            Str/ Q   - Pct - ------ Msgcnt -------
 A Destination name          Q   Tp  Lmg Smg Primry Secdry Total   Status
   TSUED01                   LOC LT    2   0    216      0    216  _____
   DESRZA70.A7CICHBT DFSASYNC LOC AP    1   0    192      0    192  _____
   T0910122                  LOC LT    1   0    144      0    144  _____
   CLIENT1 T3270LC           LOC OT    1   0     96     48    144  _____
 ***************************** Bottom of data ******************************
```

Figure 11. Messages Destinations (Detail) panel example

Example (Specifying PERCENTDST=01)

In this example, PERCENTDST is specified as 1 percent.

If you run the BROWSE function (QUEUETYPE=ALL specified) using the following set of control cards:

```
INCLUDE DESTYPE=TRANS,
        PERCENTDST=01
INCLUDE DESTYPE=LTERM,
        PERCENTDST=01
```

A QCFPRINT and BROWSE data set will be produced with:

- 216 msgs for TSUED01
- 192 msgs for DESRZA70.A7CICHBT DFSASYNC
- 144 msgs for T0910122
- 144 msgs for CLIENT1 T3270LC

This is because TSUED01, DESRZA70.A7CICHBT DFSASYNC, T0910122, CLIENT1 T3270LC are using more than 1 percent of the IMS message queue.

READ
All portions of the messages meeting the selection criteria are to be read and written to the BROWSE data set.

RECCOUNTDST=*reccountdestination*
Include in the selection process all messages queued to the same destination (such as an LTERM) that collectively equal or exceed a specified number of records on the short or long message queue data sets (SMSGQ or LMSGQ).

Restrictions:

- RECCOUNTDST is only supported in the IMS nonshared-queues environment.

- When you specify RECCOUNTDST, only one destination-specific operand can be specified. Valid destination-specific operands are:
  - DESTYPE
  - DESTINATION
  - LUNAMEDST
  - LUSIDENAME
  - TMEMBERDST
- If you specify RECCOUNTDST, primary and secondary (Primry & Secdry) are displayed in the printed report, only primary is displayed on the panel report.

An example of using RECCOUNTDST follows. This example is based on the following screen sample, which shows the IMS control region (IMS1) activity displayed after a QCF LIST or QUERY with a filter. In the figure, observe the following items:

- TSUED01 is using 2 percent of the large message queue with 216 primary messages (or records) queued.
- DESRZA70.A7CICHBT DFSASYNC is using 1 percent of the large message queue with 192 primary messages (or records) queued.
- CLIENT1 T3270LC is using 1 percent of the large message queue with 96 primary messages (or records) and 48 secondary messages (or records) queued.
- T0910122 is using 1 percent of the large message queue with 144 primary messages (or records) queued.
- VTAGB588 is using 0 percent of the large message queue with 48 primary messages (or records) and 48 secondary messages (or records) queued.
- IMSNET.L62MVS1 DFSASYNC is using 0 percent of the large message queue with 96 primary messages (or records) queued.
- T0912056 is using 0 percent of the large message queue with 96 primary messages (or records) queued.
- L63SP2T1 is using 0 percent of the large message queue with 96 primary messages (or records) queued.
- T1LRPT01 is using 0 percent of the large message queue with 96 primary messages (or records) queued.
- T0910582 is using 0 percent of the large message queue with 72 primary messages (or records) queued.
- T0913000 is using 0 percent of the large message queue with 72 primary messages (or records) queued.
- T2958327 is using 0 percent of the large message queue with 24 primary messages (or records) and 48 secondary messages (or records) queued.

```
   View  Table_Actions  Help
 -------------------------------------------------------------------------------
                       Messages Destinations (Detail)      Row 1 to 16 of 41
 Command ===>                                              Scroll ===> PAGE

 Select a row action or press END to exit
                                                   APAR . . : PH19814 2020/01/
 Server . . : QCFSERVE                             JDTE . . : 2020.016
 IMS ID . . : IMS1                                 TIME . . : 13:43:50
 QCF Func . : DETAIL                               DATE . . : 2020/01/16
 MSGQs  . . : ALL
 Row actions: C - Copy  D - Delete  X - Copy/Delete  L - List  U - Unlock


                               Str/ Q   - Pct - ------ Msgcnt -------
 A Destination name            Q    Tp  Lmg Smg Primry Secdry Total   Status
   TSUED01                     LOC  LT    2   0    216      0    216 _____
   DESRZA70.A7CICHBT DFSASYNC  LOC  AP    1   0    192      0    192 _____
   CLIENT1 T3270LC             LOC  OT    1   0     96     48    144 _____
   T0910122                    LOC  LT    1   0    144      0    144 _____
   VTAGB588                    LOC  LT    0   0     48     48     96 _____
   IMSNET.L62MVS1   DFSASYNC   LOC  AP    0   0     96      0     96 _____
   T0912056                    LOC  LT    0   0     96      0     96 _____
   L63SP2T1                    LOC  LT    0   0     96      0     96 _____
   T1LRPT01                    LOC  LT    0   0     96      0     96 _____
   T0910582                    LOC  LT    0   0     72      0     72 _____
   T0913000                    LOC  LT    0   0     72      0     72 _____
   T2958327                    LOC  LT    0   0     24     48     72 _____
 ****************************** Bottom of data *********************************
```

Figure 12. Messages Destinations (Detail) panel example

Example (Specifying RECCOUNTDST=65)

In this example, RECCOUNTDST is specified as 65 records.

If you run the BROWSE function (QUEUETYPE=ALL specified) using the following set of control cards:

```
INCLUDE DESTYPE=TRANS,
        RECCOUNTDST=65
INCLUDE DESTYPE=LTERM,
        RECCOUNTDST=65
```

A QCFPRINT and BROWSE data set will be produced with the following:

- 216 messages for destination TSUED01
- 192 messages for destination DESRZA70.A7CICHBT DFSASYNC
- 144 messages for destination CLIENT1 T3270LC
- 144 messages for destination T0910122
- 96 messages for destination VTAGB588
- 96 messages for destination IMSNET.L62MVS1 DFSASYNC
- 96 messages for destination T0912056
- 96 messages for destination L63SP2T1
- 96 messages for destination T1LRPT01
- 72 messages for destination T0910582
- 72 messages for destination T0913000
- 72 messages for destination T2958327

This is because the list above has 65 or more total messages on the message queue for each destination.

REPORTONLY
Only the header portion of messages meeting the selection criteria is to be retrieved from the shared queue. These messages are not written to the BROWSE data set. This improves performance of the BROWSE operation, as minimal data is transferred between IMS and IMS Queue Control Facility.

RTRAN
Include in the selection process remote transaction messages based on their destination (DESTYPE).

SOURCE=*sourcename(sysid)*
A 1- to 8-character logical terminal name. Groups of names that begin with the same characters can be specified generically by using an asterisk (*) after the characters that are the same.

(sysid)
One to five decimal digits in parentheses, in the form (*n*) to (*nnnnn*) where *n* is a decimal number from 1 to 2036; the number is a system ID for Multiple Systems Coupling transactions.

(*sysid*) is optional and can be used without *sourcename* to specify a source system without specifying a particular logical terminal name.

TMEMBER=*tmembername*
A 1- to 16-character field that is the TMEMBER name for OTMA transactions. Groups of names that begin with the same characters can be specified generically by using an asterisk (*) after the characters that are the same. An asterisk (*) as the first character will include all OTMA transactions.

TMEMBERDST=*tmemberdestination*
A 1- to 16-character field for selecting OTMA destination transactions by TMEMBER name where the destination is to OTMA. Groups of names that begin with the same characters can be specified generically by using an asterisk (*) after the characters that are the same. An asterisk (*) as the first character will include all OTMA destination transactions.

TMEMBERSRC=*tmembersource*
A 1- to 16-character field for selecting OTMA source transactions by TMEMBER name where the destination is not to OTMA. Groups of names that begin with the same characters can be specified generically by using an asterisk (*) after the characters that are the same. An asterisk (*) as the first character will include all OTMA source transactions.

TPIPE=*tpipename*
A 1- to 8-character field that is the TPIPE name for OTMA transactions. Groups of names that begin with the same characters can be specified generically by using an asterisk (*) after the characters that are the same. An asterisk (*) as the first character will include all OTMA tpipes.

TRANS
Include in the selection process transaction messages based on their destination (DESTYPE).

UOWORGID=*originatingimsid*
A 1- to 8-character field that is the exact name of an IMS system where a message originated. All messages with this system ID as the origination system will be included.

UOWORGTK=*originatingtoken*
A 16-character hexadecimal field specifying an originating token (time stamp) of a message. All messages with this originating token will be included.

UOWPROID=*processingimsid*
A 1- to 8-character field that is the name of an IMS system where a message was or will be processed. All messages with this system ID as the processing system will be included. Groups of names that begin with the same characters can be specified generically by using an asterisk (*) after the characters that are the same.

UOWPROTK=*processingtoken*
A 16-character hexadecimal field specifying a processing token (time stamp) of a message. All messages with this processing token will be included.

VSP
Include in the selection process certain types of VTAM subpool messages; the operand in which VSP is specified determines which types of messages are included, as follows:

DESTYPE
Include VSP messages based on their destination.

SRCTYPE
Include VSP messages based on their source.

MSGTYPE
Include VSP messages based on their source and destination. In this case, if either source or destination applies the message is selected.

VTAM=*vtamname*
A 1- to 8-character field that is the VTAM name for messages that are not LU 6.2 or OTMA. Groups of names that begin with the same characters can be specified generically by using an asterisk (*) after the characters that are the same.

NOWTOMSG statement

The NOWTOMSG statement suppresses the creation of write-to-operator (WTO) informational console messages.

NOWTOMSG is an optional control statement, and must be the first control statement provided. A flag is set to suppress creation of write-to-operator informational control messages.

Syntax diagram for NOWTOMSG statement

▶▶— NOWTOMSG —▶◀

SELECT statement

Use the SELECT statement to restrict the input log records to specified time stamp ranges or to specified record types.

The SELECT statement is optional. If used, the SELECT statement restricts the input log records to specified time stamp ranges or to specified record types.

Guidelines for coding the SELECT statement

- You can use only one SELECT statement.
- The SELECT statement, and its continuation statements, can contain one or all of the operands in any order.
- Duplicate operands are not allowed.
- Only one operand can be specified per line.
- Additional operands must be specified on continuation lines; this is done by following the operand with a comma and beginning the next operand in column 10 of the next line.

Example of SELECT statement

```
SELECT ACTION=(READ),
       TYPE=IO
```

Syntax diagram for SELECT statement

```
►►── SELECT ─────────────────────────────────────────────────────►

                         ┌──── UNLOCK ────┐
            ┌─ ACTION=( ──┼──── DELETE ────┼─── ), ─┐
                          ├──── READ ──────┤
                          └── REPORTONLY ──┘

                    ┌──── , choose one or more ◄───┐
            ►────────┤                              ├──►◄
                     ├──── ENDNUM= endnum ──────────┤
                     ├──── ENDTIME= endtime ────────┤
                     ├──────── NOCONVMSG ───────────┤
                     ├──────── NOSYSMSG ────────────┤
                     ├──── STARTNUM= startnum ──────┤
                     ├──── STARTTIME= starttime ────┤
                     └──── TYPE= ──┬──── I ──────────┘
                                   ├──── IO ─────┤
                                   ├──── O ──────┤
                                   ├──── DI ─────┤
                                   ├──── DO ─────┤
                                   └──── DIO ────┘
```
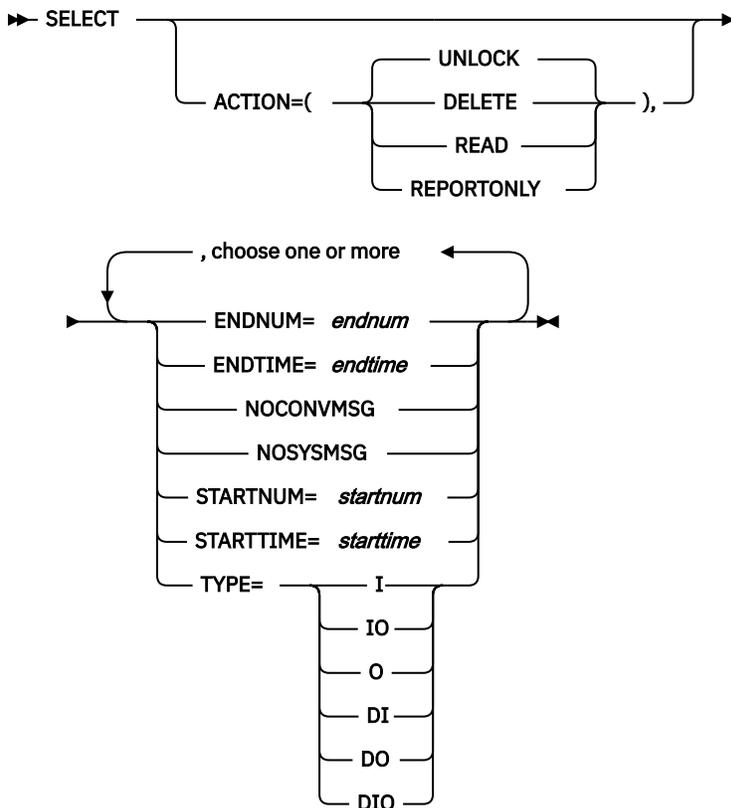
Parameter reference for SELECT statement

The variable fields are defined as follows:

DELETE
All messages meeting the selection criteria are to be deleted from the cold queue.

ENDNUM=*endnum*
An 8-digit hexadecimal representation of a 4-byte log sequence number.

Any record with a sequence number less than or equal to *endnum* satisfies this criteria.

ENDTIME=*endtime*
A value specifying a log record date and time stamp in the following format:

> *D=yyyy.ddd,T=hh:mm:ss.thmiju,Z=shh:mm*

where *yyyy.ddd* is the date, *hh:mm:ss.thmiju* is the time, and *shh:mm* is the zone; together, these three comprise the Universal Time Stamp.

- *yyyy* = year
- *ddd* = day
- *hh* = hour
- *mm* = minute
- *ss* = second
- *t* = tenths of a second
- *h* = hundredths of a second
- *m* = milliseconds
- *i* = tenths of a millisecond

- *j* = hundredths of a millisecond
- *u* = microseconds
- *s* = sign + or - from Greenwich
- *hh* = hour offset from Greenwich
- *mm* = minutes offset from Greenwich

Because most installations are in an hourly time zone, the zone can be represented simply as the sign (if negative- if west of GMT (Greenwich mean time)) and the offset. For example, San Jose, CA is Z=-7 or Z=-8; Uithoorn in the Netherlands is Z=1 or Z=2; GMT is Z=0. If minutes are required, for example for Gander, Newfoundland, the leading zero can be omitted (for example, Z=-3:30).

All date and time stamps are in UTC (coordinated universal time) format. The zone is *added* to the UTC to produce local time.

Any record with a date and time stamp less than or equal to *endtime* satisfies this standard.

NOCONVMSG
No conversational messages are to be selected except for MSC response conversational messages, which are always selected. Messages rejected because the NOCONVMSG operand will have a scrap point of 6, and are reported as part of the select statistics messages.

NOSYSMSG
Specifies that no system messages are to be selected. You can bypass IMS system messages by adding NOSYSMSG to the SELECT control statement.

READ
All portions of the messages meeting the selection criteria are to be read and written to the RECOVER data set.

REPORTONLY
Only the header portion of messages meeting the selection criteria is to be retrieved from the shared queue. These messages are not written to the RECOVER data set. This improves performance of the RECOVER operation, as minimal data is transferred between IMS and IMS Queue Control Facility.

STARTNUM=*startnum*
An 8-digit hexadecimal representation of a 4-byte log sequence number.

Any record with a sequence number greater than or equal to *startnum* satisfies this criteria.

STARTTIME=*starttime*
A value specifying a log record date and time stamp in the following format:

>   *D=yyyy.ddd,T=hh:mm:ss.thmiju,Z=shh:mm*

where *yyyy.ddd* is the date, *hh:mm:ss.thmiju* is the time, and *shh:mm* is the zone; together, these three comprise the Universal Time Stamp.

- *yyyy* = year
- *ddd* = day
- *hh* = hour
- *mm* = minute
- *ss* = second
- *t* = tenths of a second
- *h* = hundredths of a second
- *m* = milliseconds
- *i* = tenths of a millisecond
- *j* = hundredths of a millisecond
- *u* = microseconds
- *s* = sign + or - from Greenwich

- *hh* = hour offset from Greenwich
- *mm* = minutes offset from Greenwich

Because most installations are in an hourly time zone, the zone can be represented simply as the sign (if negative- if west of GMT (Greenwich mean time)) and the offset. For example, San Jose, CA is Z=-7 or Z=-8; Uithoorn in the Netherlands is Z=1 or Z=2; GMT is Z=0. If minutes are required, for example for Gander, Newfoundland, the leading zero can be omitted (for example, Z=-3:30).

All date and time stamps are in UTC (coordinated universal time) format. The zone is *added* to the UTC to produce local time.

Any record with a date and time stamp less than or equal to *endtime* satisfies this standard.

TYPE
One or two characters specifying the types of record to be selected:

> **Type**
> > **Selected record types**
>
> **I**
> > Type 01 input records
>
> **O**
> > Type 03 output records
>
> **IO**
> > Type 01 input and type 03 output records
>
> **DI**
> > Only DUMPQ Type 4002 input records
>
> **DO**
> > Only DUMPQ Type 4002 output records
>
> **DIO**
> > Only DUMPQ Type 4002 input and output records

If TYPE is not specified, all record types acceptable to RECOVER are selected.

TYPE DI, DO, and DIO are only applicable for RECOVER jobs (RECOVERAB, RECOVERDM, and REPROCESS).

UNLOCK
All messages meeting selection criteria are to be requeued from the cold queue to the queue from which they originated for future processing.

TITLE statement

Use the TITLE statement to cause a title to be displayed your reports.

The TITLE statement is optional. If you specify the TITLE statement, the report title that you specify will appear as the third title line on all reports.

Guidelines for coding the TITLE statement

- The data for the title is enclosed in apostrophes.
- Columns 73 through 80 of the record are ignored.
- The data can be continued by not having an ending apostrophe on the first record, and starting in column one of a second record.
- Maximum length of a title is 120 characters; anything longer will be truncated.

Example TITLE statement

If you want an apostrophe within the title, two contiguous apostrophes must be used.

For example, a title of *See what is in A97's Queues*, is specified as follows:

```
title='See what is in A97''s Queues'
```

Syntax diagram for TITLE statement

▶▶─ TITLE= ── *'title data to be used in report headings'*   ─▶◀

WTOMSG statement

The WTOMSG statement creates write-to-operator (WTO) informational console messages.

WTOMSG is an optional control statement, and must be the first control statement provided.

A flag is set in the output message records to create write-to-operator informational control messages.

Syntax diagram for WTOMSG statement

▶▶─ WTOMSG ─▶◀

Parameter reference for WTOMSG statement

The following mutually exclusive parameters can be used to display the top *nn* destinations:

- TOPLASTACTIVWTO (TOPLA)
- TOPALLMSGWTO (TOPAM)
- TOPLONGMSGWTO (TOPLM)
- TOPSHORTMSGWTO (TOPSM)

Chapter 11. Browsing the message queues

The BROWSE function allows you to browse the message queues in both shared- and nonshared-queues environments.

Topics:

BROWSE function description

The BROWSE function browses and reports on the messages in the queues.

BROWSE can copy messages, or extract information from messages to produce a report. BROWSE has multiple uses; you can use the BROWSE function to perform the following tasks:

- Determine whether the IMS message queues need to be cleaned up (UNLOAD can be used later to remove unwanted messages)
- Determine whether any messages exist on the shared queues or the IMS message queues that should be requeued later or deleted by RECOVER
- Copy records for later insertion by the LOAD function

Browse logic selects records from the message queues for possible later insertion by the load function, deletion by the unload function, or for purely informational purposes. Browse logic performs the following tasks:

1. Processes control blocks that were built by IMS Queue Control Facility control processing logic from input control statements make the following determinations:

   - Determines queue selection from the QUEUETYPE= keyword of the FUNCTION statement
   - Determines record selection criteria for date and time stamps, and record type from the SELECT statement
   - Determines record selection criteria for message sources and destinations from the INCLUDE and EXCLUDE statements

2. Discards and ignores records that fail to meet selection criteria of the FUNCTION, SELECT, INCLUDE, and EXCLUDE statements

3. Writes records that do meet selection criteria of the FUNCTION, SELECT, and INCLUDE statements to the browse data set

4. Produces reports on the number and types of records that the BROWSE function selected from the message queues

5. Returns control to IMS Queue Control Facility control processing logic

The job steps that are required for the BROWSE function are as follow: BROWSE

Browse processing

IMS Queue Control Facility browse logic creates a browse data set that contains message records that are selected from the message queues. This browse data set's messages are messages that can eventually be read into the message queues by the load function.

Selection parameters

Browse logic selects messages from the message queues based on your selection criteria. You specify the selection criteria in control statements.

Selection criteria

You specify selection criteria to the browse logic with FUNCTION, SELECT, INCLUDE, and EXCLUDE statements.

Use these statements to define a subset of message records to be selected from the message queues. The SELECT, INCLUDE, and EXCLUDE statements are optional. If none of these three statements is present, all of the records from all of the message queues will be selected.

Message records can be selected based on the following criteria:

Specified in FUNCTION statement:

- Message age
- Queue type
  - ALL
  - APPC
  - COLD
  - DEADQ
  - LONG
  - LTERM
  - OTMA
  - REMOTE (remote MSNAME queue)
  - SHORT
  - SERIAL (serial transaction queue)
  - SUSPEND (suspend transaction queue)
  - TRANSACTION (transaction ready queue (remote and local))

Specified in SELECT statement:

    Beginning date and time
    Conversational messages or no conversational messages
    Ending date and time
    Log record type
    System messages or no system messages

Specified in INCLUDE or EXCLUDE statement:

    Destination, source type, or both of a message
    Destination with optional system ID
    Logical unit name
    Logical unit name destination
    Logical unit sidename
    Logical unit tpname
    OTMA tmember name

OTMA tmember name destination
OTMA tmember name source
OTMA tpipe name
Source with optional system ID
Specific IMS ID
Specific IMS ID destination
Specific IMS ID source
Either the originating unit-of-work, or processing IMS ID, or both
Either the originating unit-of-work, or processing token, or both
VTAM name

The FUNCTION statement in the AGE= keyword specifies how old a message must be to be processed, and also specifies in the QUEUETYPE= keyword, one or more message queues to be processed.

The SELECT statement specifies global criteria, a range of times, and the types of records to be selected by browse logic. Only one SELECT statement is allowed. The SELECT statement can also be used to specify whether the browse data set is created using the READ or REPORTONLY options of the ACTION= keyword.

INCLUDE and EXCLUDE statements specify selection criteria specific to message sources and destinations. INCLUDE and EXCLUDE statements are unrestricted in number.

If any INCLUDE statements are present, a record must meet all criteria for at least one INCLUDE statement to be selected by browse logic. Unspecified criteria are considered to be met.

A record that meets all criteria that is specified by an EXCLUDE statement is excluded from the BROWSE function selections even if the record meets all criteria specified by the FUNCTION, SELECT, and INCLUDE statements. Unspecified criteria are considered to be met.

Selection processing

When given control by IMS Queue Control Facility control processing logic, browse logic determines whether a browse data set should be created by determining if ACTION=(READ) was specified, or defaulted to, on the FUNCTION or SELECT statement.

If the BROWSE data set was to be created but is not present, browse logic issues a message and continues as if ACTION=(REPORTONLY) were specified.

Browse logic calls IMS requesting that a BROWSE function be performed on the message queues that you specified.

The output logic determines if the record meets the selection criteria that is specified by any SELECT and INCLUDE statements. If the record does not meet these selection criteria, the record is discarded.

If a record does meet these selection criteria, the browse logic determines if the record meets the criteria for any EXCLUDE statement. If the record meets the criteria for an EXCLUDE statement, the record is discarded. The browse logic appends a prefix to each message record before writing the record to the browse data set.

When end-of-data is signaled by IMS, browse logic produces reports on the number and type of messages that BROWSE found on the message queues and statistics on the number and types of records that BROWSE processed.

Browse logic then returns control to IMS Queue Control Facility control processing logic for clean up and normal termination.

Control statements for BROWSE function

To obtain a report and a copy of the messages on the various message queues, use the BROWSE function.

The BROWSE function lets you obtain a report and a copy of the messages on the various message queues.

To specify the messages that you want to browse from the IMS message queue, you provide control statements by way of IMS Queue Control Facility control processing in the BROWSE step. You can create these control statements by using the ISPF panels. These control statements specify the criteria to use to select messages.

A copy of each message that is selected is written to the BROWSE data set, if the BROWSE data set is present and if REPORTONLY was not specified. The original message is left on the shared queues and is available to other IMS systems.

The control statements that are recognized by the BROWSE function are:

```
* comment
ACTION
AGE
AREA
EXCLUDE
INCLUDE
QUEUETYPE
SELECT
TITLE
```

BROWSE control statement abbreviations

You can abbreviate the specification of some of the BROWSE control statements.

This table shows a list of keyword abbreviations that are allowed for BROWSE by the IMS Queue Control Facility control processing interpreter:

Table 14. BROWSE processing keyword abbreviations

Keyword	Abbreviation
ACTION	AC
AREA	AR
BROWSE	BR
LTERM	LT
OTMA	OT

FUNCTION BROWSE statement syntax

Specify the FUNCTION BROWSE control statement according to the following syntactic specifications.
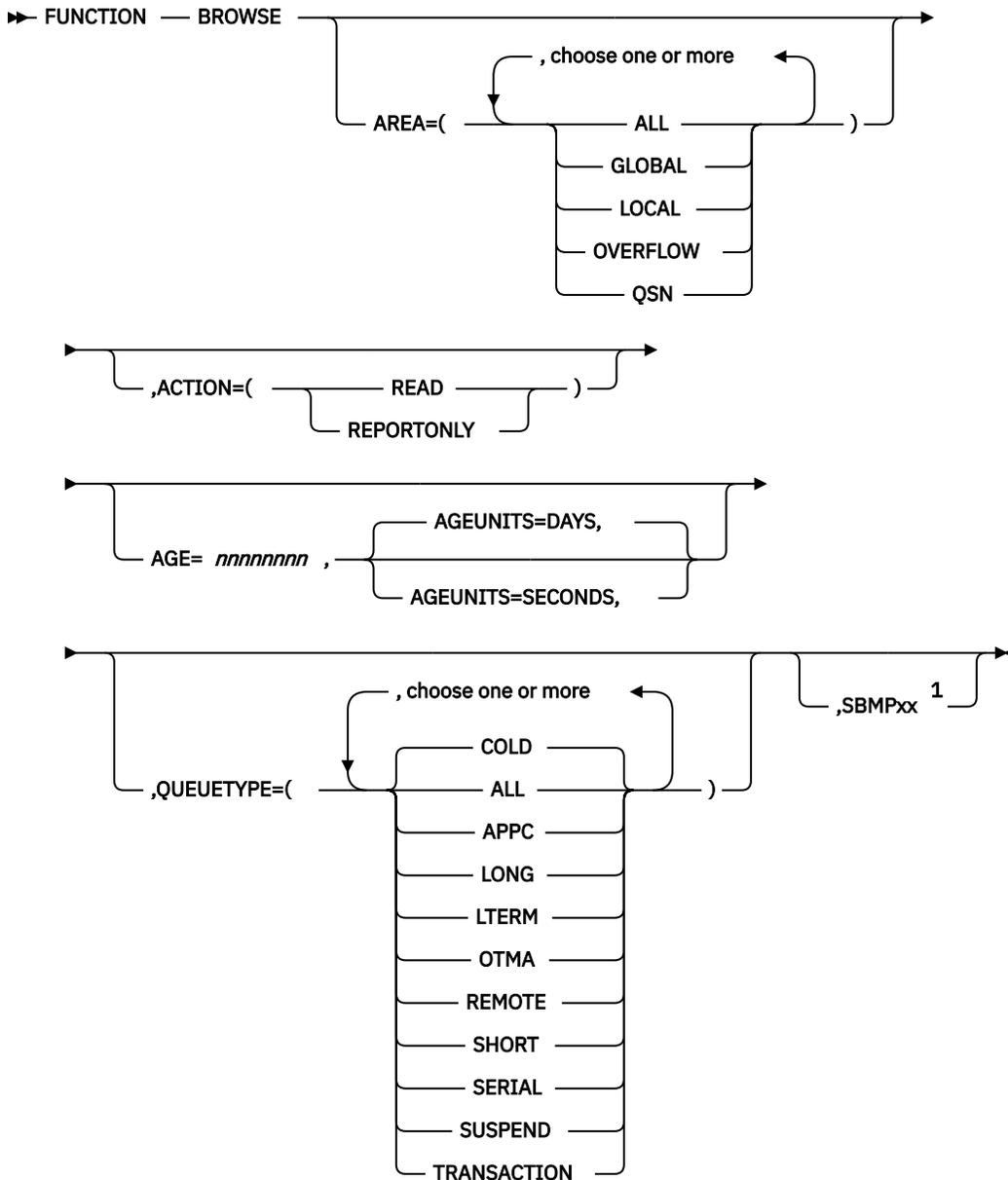
Notes:

[1] Required for all jobs that are started because a threshold was crossed.

FUNCTION BROWSE INCLUDE statement syntax

FUNCTION BROWSE can use the INCLUDE control statement to choose certain messages.

In the JCL, the INCLUDE control statement is written on a separate line after the FUNCTION BROWSE control statement.

JCL example 1

```
FUNCTION BROWSE QUEUETYPE=(ALL)
INCLUDE DESTINATION=A*
END
```

JCL example 2

```
FUNCTION BROWSE AREA=(ALL),QUEUETYPE=(ALL)
INCLUDE DESTINATION=A*,
IMSIDS=IMS1
END
```

Syntax diagram

Other options

```
                          , choose one or more
        ┌──────────────────────────────────────────┐
──►►──┬─┴─ DESTINATION= destination (sysid) ────────┴──►◄──
      │                    , choose one or more
      │              ┌──────────────────────────┐
      ├─ DESTYPE=( ──┴─┬─ APPC ──┬─┐── ) ──
      │                ├─ LTERM ─┤
      │                ├─ LTRAN ─┤
      │                ├─ MSC ───┤
      │                ├─ OTMA ──┤
      │                ├─ RTRAN ─┤
      │                ├─ TRANS ─┤
      │                └─ VSP ───┘
      │                      , choose one or more
      │                ┌──────────────────────────┐
      ├────────────────┼─ IMSDSTID= destinationimsid ─┤
      │                ├─ IMSSRCID= sourceimsid ──────┤
      │                └─ IMSIDS= imsid ──────────────┘
      │                      , choose one or more
      │                ┌──────────────────────────┐
      ├────────────────┼─ LUNAMEDST= lunamedestination ─┤
      │                ├─ LUNAMESRC= lunamesource ──────┤
      │                └─ LUNAME= luname ───────────────┘
      ├─ LUSIDENAME= lusidename ──
      ├─ LUTPNAME= lutpname ──
      ├─ MSGTYPE=( ─┬─ APPC ─┬─ ) ──
      │             ├─ MSC ──┤
      │             ├─ OTMA ─┤
      │             └─ VSP ──┘
      ├─ PERCENTDST= percentdestination ──
      ├─ RECCOUNTDST= reccountdestination ──
      ├─ SOURCE= sourcename (sysid) ──
      │                      , choose one or more
      │                ┌──────────────────────────┐
      ├─ SRCTYPE=( ────┴─┬─ APPC ──┬─┐── ) ──
      │                  ├─ LTERM ─┤
      │                  ├─ MSC ───┤
      │                  ├─ OTMA ──┤
      │                  └─ VSP ───┘
      │                      , choose one or more
      │                ┌──────────────────────────┐
      ├────────────────┼─ TMEMBERDST= tmemberdestination ─┤
      │                ├─ TMEMBERSRC= tmembersource ──────┤
      │                └─ TMEMBER= tmembername ───────────┘
      ├─ TPIPE= tpipename ──
      ├─ UOWORGID= originatingimsid ──
      ├─ UOWORGTK= originatingtoken ──
      ├─ UOWPROID= processingimsid ──
      ├─ UOWPROTK= processingtoken ──
      └─ VTAM= vtamname ──
```

FUNCTION BROWSE SELECT statement syntax

FUNCTION BROWSE can use the SELECT statement to select messages that meet particular criteria.

In the JCL, the SELECT control statement is written on a separate line after the FUNCTION BROWSE control statement.

JCL example 1

```
FUNCTION BROWSE QUEUETYPE=(ALL)
SELECT NOCONVMSG
END
```

JCL example 2

```
FUNCTION BROWSE AREA=(ALL),QUEUETYPE=(ALL)
SELECT NOCONVMSG,
NOSYSMSG
END
```

Syntax diagram
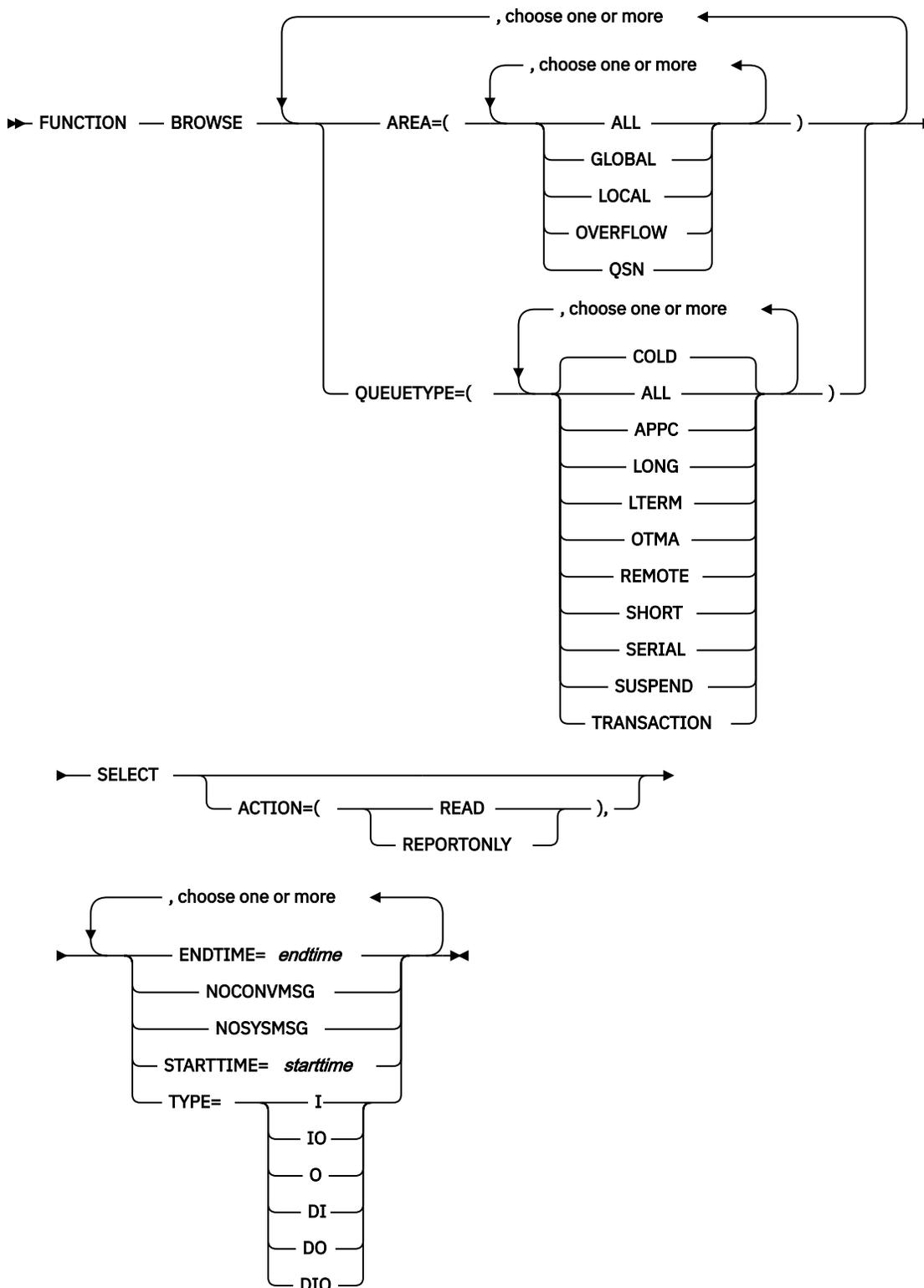
JCL for the BROWSE function

Use the BROWSE step in your JCL job stream to create reports about the message queues.

The purpose of the BROWSE step is to obtain a report about, and optionally, copies of messages on the shared queues. The BROWSE function can determine if messages exist on the cold queue.

You can run the BROWSE function in batch mode by submitting a JCL job stream.

Sample JCL to run the BROWSE function

The following example shows the JCL required to start a BROWSE step. Replace the items in italics with values that are appropriate for your environment:

```
//***********************************************************************
//* SAMPLE JCL TO EXECUTE THE BROWSE FUNCTION                           *
//***********************************************************************
//BROWSE    EXEC PGM=IQCINI0$,
//      PARM='BPEINIT=IQCBINI0,BPECFG=IQCIVPCF,IMSPLEX=IPLEX,QCFIMS=IMS1'
//STEPLIB  DD DISP=SHR,DSN=hlqual.SIQCLINK
//         DD DISP=SHR,DSN=IMSREL.RESLIB
//QCFPRINT DD SYSOUT=A
//BROWSE   DD DSN=QCF.BROWSE,UNIT=SYSDA,DISP=(NEW,CATLG),
//         DCB=(LRECL=32756,BLKSIZE=32760,RECFM=VB),
//         SPACE=(CYL,(5,5))
//SNAPDUMP DD SYSOUT=A
//QCFIN    DD DSN=USER.PRIVATE.PROCLIB(browse),UNIT=SYSDA
```

DD statements for BROWSE

STEPLIB DD
Defines the library containing the IMS Queue Control Facility program load modules and IMS libraries that are required for the browse step to run.

This DD statement is required.

QCFPRINT DD
Defines the output report data set.

This DD statement is required.

BROWSE DD
Defines the output data set. This can be used as input to the LOAD function.

If `ACTION=(READ)` is specified or defaulted to on the FUNCTION or SELECT statement, then this statement is required.

If `ACTION=(REPORTONLY)` is specified on the FUNCTION or SELECT statement, then this statement is optional.

QCFIN DD
Defines the input control statement data set.

This DD statement is required.

Example of the PROC USER.PRIVATE.PROCLIB(*browse*)

The following example shows the PROC that was specified above in the QCFIN DD statement as, USER.PRIVATE.PROCLIB(*browse*):

```
**********************************************************
* browse all the message in the queues
**********************************************************
function browse queuetype=all
end
```

Chapter 12. Loading the message queues

The LOAD function allows you to reload messages to the message queue in both shared- and nonshared-queues environments.

Topics:

LOAD function description

The LOAD function requeues messages to the message queue and produces a data report.

Use the LOAD function to load messages into one or more message queues that you specify, from a data set that is created by the BROWSE, UNLOAD, RECOVER, or SORTB function. Use the LOAD function to produce a report on the data that the LOAD function loads into the message queues.

You typically use the LOAD function for stress, regression, or application testing. You can also use LOAD as a migration aid.

When you run LOAD as a migration aid, the LOAD function requeues messages across supported IMS releases. You can load messages that are created on one supported release of IMS into another supported release of IMS if the following requirement is met. The source and destination resources (such as LTERMs, transactions, MSC names) that are involved must be defined on both IMS systems.

You can use the LOAD function to perform the following tasks:

- To requeue messages to the message queue
- To produce a data report
- To perform regression testing
- To perform application testing
- As an aid in performing fallback
- As an aid in performing migration

The load function selects records to be inserted from the LOAD data set to the message queues.

The load function performs the following tasks:

1. Processes control blocks that were built by IMS Queue Control Facility control processing logic from input control statements to make the following determinations:

   - Determines queue selection from the QUEUETYPE= keyword of the FUNCTION statement
   - Determines record selection criteria for date and time stamps, and record type from the SELECT statement
   - Determines record selection criteria for message sources and destinations from the INCLUDE and EXCLUDE statements

2. Discards or ignores records that fail to meet the selection criteria of the SELECT, INCLUDE, and EXCLUDE statements

3. Reloads that records that do meet the selection criteria of the SELECT and INCLUDE statements to the message queues

4. Produces reports on the number and types of records the LOAD function loaded to the message queues
5. Produces a report of the messages that IMS failed to reload, with the AIBREASN codes identifying the errors, and writes those messages to the SCRAPLOG data set
6. Returns control to IMS Queue Control Facility control processing logic

The job steps that are required for the load function are as follow: LOAD

Limitations of the LOAD function

When IMS abends during a QCF LOAD of messages that were queued by the IMS QUEUE TRAN command, type 01 (input) messages might be lost.

The following sequence of events can result in the loss of type 01 messages:

1. The IMS QUEUE TRAN command is issued. IMS loads the messages onto transaction *tranname* in the shared message queue.
2. The QCF UNLOAD job is run. IMS unloads the messages from the shared message queue to a temporary destination.
3. The QCF LOAD job is run. IMS starts reloading the messages onto transaction *tranname* in the shared message queue. However, before the QCF LOAD job completes, IMS abends abnormally.
4. The /ERE command is issued and IMS restarts. Any type 01 message that was not reloaded onto transaction *tranname* before the abend is lost. The IMS message queue recovery process does not support temporary destinations for type 01 messages.

Load processing

The LOAD function inserts messages to the message queues from the data set that is specified in the LOAD DD statement.

This LOAD DD data set was produced by one of the following functions:

- BROWSE
- UNLOAD
- RECOVER
- SORTB output of a RECOVERAB, RECOVERDM, or REPROCESS

Selection parameters

Based on your selection criteria, the load function selects messages from the LOAD data set to into the message queues. You specify the selection criteria in control statements that are read by the IMS Queue Control Facility control processing logic.

Selection criteria

You specify selection criteria to the load function with FUNCTION, SELECT, INCLUDE, and EXCLUDE statements. Use these statements to define the subset of message records from the LOAD data set that are to be inserted to the message queues. The SELECT, INCLUDE, and EXCLUDE statements are optional. If none of these three statements is present, all of the records in the LOAD data set will be inserted.

You can select message records based on the following criteria:

As specified in the FUNCTION statement with the following keyword and parameters:

- Queue type
  - ALL
  - APPC
  - LTERM

- OTMA
- REMOTE (remote MSNAME queue)
- SERIAL (serial transaction queue)
- SUSPEND (suspend transaction queue)
- TRANSACTION (transaction ready queue (local and remote))

As specified in the SELECT statement with the following parameters:

Beginning date and time
Conversational messages or no conversational messages
Ending date and time
Log record type
System messages or no system messages

As specified in the INCLUDE or EXCLUDE statement with the following parameters:

Destination with optional system ID
Destination type
Logical unit name
Logical unit name destination
Logical unit sidename
Logical unit source
Logical unit tpname
Message type
Source with optional system ID
Source type
TMember name
TMember destination
TMember source
TPipe name
Specific IMS ID
Specific IMS ID destination
Specific IMS ID source
Either the originating unit-of-work, or the processing IMS ID, or both
Either the originating unit-of-work, or the processing token, or both
VTAM name

The FUNCTION statement QUEUETYPE= keyword specifies one or more message queues to be processed.

The SELECT statement specifies global criteria, a range of times, and the types of records to be selected by the load function for insertion. Only one SELECT statement is allowed.

INCLUDE and EXCLUDE statements specify selection criteria specific to message sources and destinations. INCLUDE and EXCLUDE statements are unrestricted in number.

If any INCLUDE statements are present, a record must meet all criteria for at least one INCLUDE statement to be included in the load function selections. Unspecified criteria are considered to be met.

A record that meets all of the criteria that is specified by an EXCLUDE statement will be excluded from the load function selections, even if the record meets all of the criteria that is specified by the SELECT statement and an INCLUDE statement. Unspecified criteria are considered to be met.

Selection Processing

When given control by IMS Queue Control Facility control processing logic, the load function determines if a load data set is defined. If the load data set is not present, the load function terminates with a message.

The load function issues calls to IMS requesting that a Common Queue Services (CQS) LOAD function be performed.

The load function determines if the record meets the selection criteria that is specified by any SELECT and INCLUDE statements. If the record does not meet these selection criteria, the record is not inserted.

If a record does meet these selection criteria, the load function determines if the record meets the criteria for any EXCLUDE statement. If the record meets the criteria for an EXCLUDE statement, the record is not inserted.

When end-of-file is reached on the data set that is pointed to by the LOAD DD statement, the load function produces reports about the number and type of messages the load function inserted to the message queues and of the statistics on the number and types of records that the load function processed.

The load function then returns control to IMS Queue Control Facility control processing logic for clean up and normal termination.

Control statements for LOAD function

Use the LOAD function to load selected messages to the IMS message queues in both shared- and nonshared-queues environments.

The LOAD function lets you put selected messages from an input data set (that is created by the UNLOAD, BROWSE, RECOVER, or SORTB functions) to the IMS message queues.

To specify the messages that you want to load into the IMS message queue, you provide control statements to LOAD through IMS Queue Control Facility control processing in the LOAD step. You can create these control statements by using the ISPF panels. These control statements specify the criteria to use to select messages.

The control statements that are recognized by the LOAD function are:

```
* comment
CHNGDEST
CURMSGTIME
DEBUG
EXCLUDE
INCLUDE
MAXMSGCT
MAXWAIT
PACING**
QUEUETYPE
SELECT
TITLE
WAIT
```

**** - When pacing is used for a LOAD function and IMS crashes, messages might be lost because IMS does not write the data into the current OLDS buffer.

LOAD control statement abbreviations

You can abbreviate the specification of some of the LOAD control statements.

The following table shows a list of keyword abbreviations that are allowed for LOAD by the IMS Queue Control Facility control processing interpreter:

Table 15. IMS Queue Control Facility LOAD control processing keyword abbreviations

Keyword	Abbreviation
CHNGDEST	CD

Table 15. IMS Queue Control Facility LOAD control processing keyword abbreviations (continued)

Keyword	Abbreviation
DEBUG	DB
EXCLUDE	EX
INCLUDE	IN
MAXMSGCT	MM
PACING	PA
QUEUETYPE	QT
SELECT	SE

FUNCTION LOAD statement syntax

Specify the FUNCTION LOAD keywords and parameters according to the following syntactic conventions.
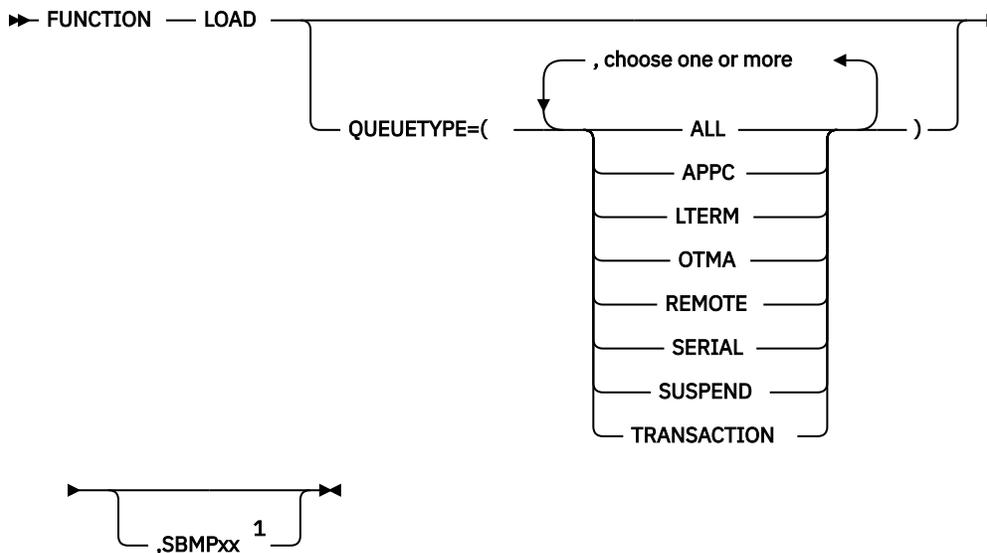
Notes:

[1] Required for all jobs that are started because a threshold was crossed.

FUNCTION LOAD INCLUDE statement syntax

The INCLUDE statement is used by FUNCTION LOAD and its keywords and parameters to further screen messages and message types.

In the JCL, the INCLUDE control statement is written on a separate line after the FUNCTION LOAD control statement.

JCL example 1

```
FUNCTION LOAD QUEUETYPE=(ALL)
INCLUDE DESTINATION=A*
END
```

JCL example 2

```
FUNCTION LOAD QUEUETYPE=(ALL)
INCLUDE DESTINATION=A*,
```

```
IMSIDS=IMS1
END
```

Syntax diagram

```
                              , choose one or more
                                   COLD
  ►►─ FUNCTION ── LOAD ── QUEUETYPE=(        ALL         )── INCLUDE ──►
                                   APPC
                                   LONG
                                   LTERM
                                   OTMA
                                   REMOTE
                                   SHORT
                                   SERIAL
                                   SUSPEND
                                   TRANSACTION

  ►─ Other options ─►◄
```

Other options

```
                          , choose one or more
      ►►─┬──────────── DESTINATION= destination (sysid) ──────────────────────┬──►◄
         │
         │                              , choose one or more
         ├── DESTYPE=( ──┬────── APPC ──────┬── ) ──
         │                ├────── LTERM ─────┤
         │                ├────── LTRAN ─────┤
         │                ├────── MSC ───────┤
         │                ├────── OTMA ──────┤
         │                ├────── RTRAN ─────┤
         │                ├────── TRANS ─────┤
         │                └────── VSP ───────┘
         │
         │                      , choose one or more
         ├──┬── IMSDSTID= destinationimsid ──┬──
         │  ├── IMSSRCID= sourceimsid ────────┤
         │  └── IMSIDS= imsid ────────────────┘
         │
         │                         , choose one or more
         ├──┬── LUNAMEDST= lunamedestination ──┬──
         │  ├── LUNAMESRC= lunamesource ────────┤
         │  └── LUNAME= luname ─────────────────┘
         ├── LUSIDENAME= lusidename ──
         ├── LUTPNAME= lutpname ──
         │                              , choose one or more
         ├── MSGTYPE=( ──┬────── APPC ──────┬── ) ──
         │                ├────── MSC ───────┤
         │                ├────── OTMA ──────┤
         │                └────── VSP ───────┘
         ├── SOURCE= sourcename (sysid) ──
         │                              , choose one or more
         ├── SRCTYPE=( ──┬────── APPC ──────┬── ) ──
         │                ├────── LTERM ─────┤
         │                ├────── MSC ───────┤
         │                ├────── OTMA ──────┤
         │                └────── VSP ───────┘
         │                           , choose one or more
         ├──┬── TMEMBERDST= tmemberdestination ──┬──
         │  ├── TMEMBERSRC= tmembersource ────────┤
         │  └── TMEMBER= tmembername ─────────────┘
         ├── TPIPE= tpipename ──
         ├── UOWORGID= originatingimsid ──
         ├── UOWORGTK= originatingtoken ──
         ├── UOWPROID= processingimsid ──
         ├── UOWPROTK= processingtoken ──
         └── VTAM= vtamname ──
```

FUNCTION LOAD SELECT statement syntax

This topic describes the FUNCTION LOAD SELECT statement syntax.

In the JCL, the SELECT control statement is written on a separate line after the FUNCTION LOAD control statement.

JCL example 1

```
FUNCTION LOAD QUEUETYPE=(ALL)
SELECT NOCONVMSG
END
```

JCL example 2

```
FUNCTION LOAD QUEUETYPE=(ALL)
SELECT NOCONVMSG,
NOSYSMSG
END
```

Syntax diagram
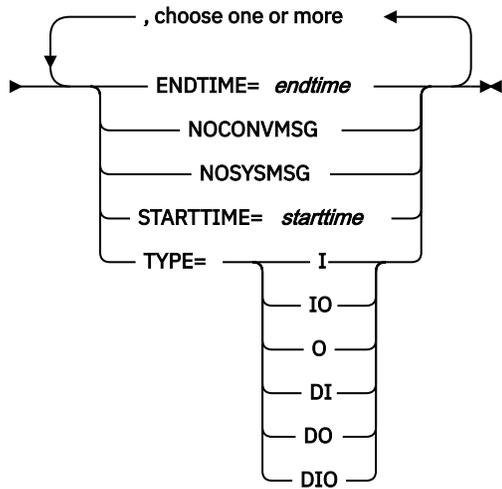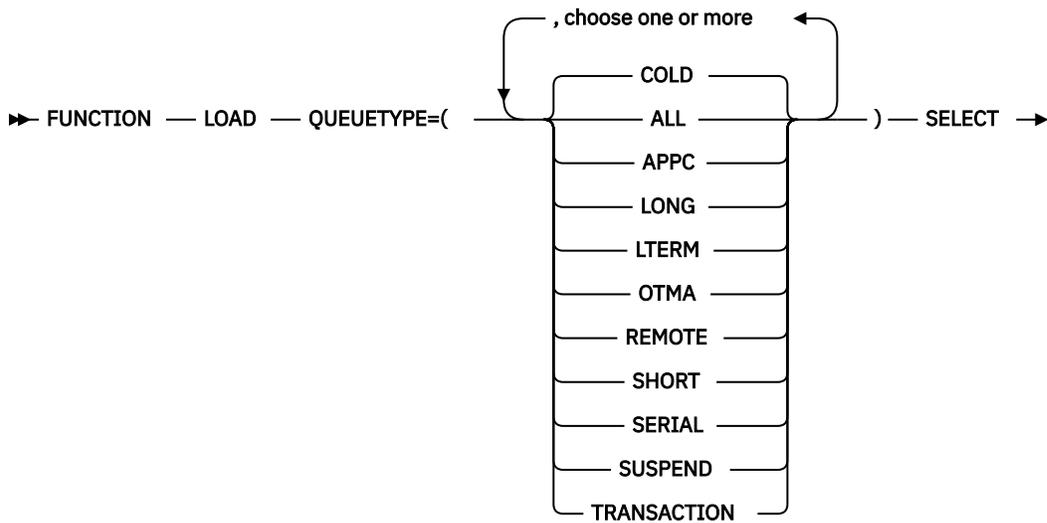
CHNGDEST statement

Use the LOAD step CHNGDEST statement to modify certain source and destination fields in message records.

The CHNGDEST statement is an optional control statement of the LOAD step.

The CHNGDEST statement lets you modify those fields in message records that indicate the source and destination of the message so that messages in a test environment can be redirected to respond to a single LTERM, TRANCODE, or TMEMBER/TPIPE.

Guidelines for coding the CHNGDEST statement

- The CHNGDEST control statement only supports local messages (this can include MSC messages where the TO and FROM systems are both the local system).

  All other message types are not supported by CHNGDEST statement and must be excluded.
- The CHNGDEST statement requires the FUNCTION, LOAD, and QUERY=ALL keywords in order to function properly, otherwise the CHNGDEST statement performs a BROWSE function.
- The following parameter combinations are valid for the CHNGDEST statement:
  - Use LTERM alone
  - Use TRAN alone
  - Use LTERM and TRAN together
  - Use TMEMBER and TPIPE together (specified on separate lines)

    Note: TMEMBER and TPIPE can only be specified in combination with each other.
  - Use TRAN and TMEMBER/TPIPE together

    Note: LTERM and TMEMBER/TPIPE are exclusive and cannot be used in combination.

Restriction: If CHNGDEST and MAXMSGCT are coded, the MAXMSGCT statement is ignored.

Syntax diagram for CHNGDEST statement
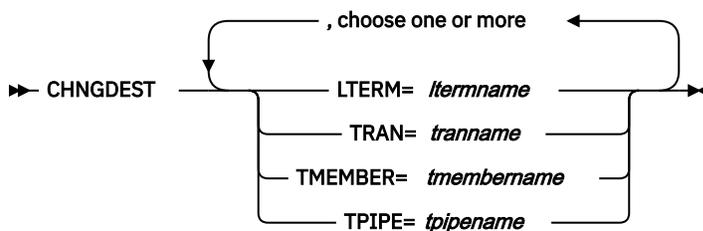
Parameter reference for CHNGDEST statement

LTERM=*ltermname*
Specifies a 1- to 8-character logical terminal name. The source name in all messages is changed to the new LTERM name, and all messages with LTERM as a destination will also have the destination name changed to the new LTERM name.

TRAN=*tranname*
Specifies a 1- to 8-character transaction code name. All messages with TRANSACTION as a destination will have the destination name changed to the new TRANCODE name.

TMEMBER=*tmembername*
Specifies a 1- to 16-character field that is the TMEMBER name for the OTMA destination.

TPIPE=*tpipename*
Specifies a 1- to 8-character field that is the TPIPE name for the OTMA destination.

If the OTMA client is sensitive to OTMA user data, you must modify DFSYIOE0. See the sample IMS Queue Control Facility IQCYIOE0 for IMS Connect in the IMS Queue Control Facility Sample Library.

CURMSGTIME statement

Use the CURMSGTIME statement to reset the message prefix times to the current load time.

The CURMSGTIME statement is an optional control statement of the LOAD function.

When you code the CURMSGTIME statement, all of the times in the message prefixes are set to the current time of the LOAD.

The CURMSGTIME statement is valid for STCK format and for UTC format times in message prefixes.

Syntax diagram for CURMSGTIME statement

▶▶— CURMSGTIME —▶◀

DEBUG statement

Use the DEBUG statement of the LOAD function only in conjunction with the PACING statement to gather documentation for problem troubleshooting.

DEBUG is an optional LOAD statement that can be used only in conjunction with the PACING statement.

DEBUG should only be used when you need to gather further documentation for problem troubleshooting purposes when directed to do so by IBM Support service.

When pacing is used with the LOAD function and IMS crashes, messages might be lost because IMS does not write the data into the current OLDS buffer.

Syntax diagram for DEBUG statement

▶▶— DEBUG ──────────────▶◀
         └── PACING ──┘

Note: A SNAPDUMP DD statement must be included in the JCL to obtain the desired output.

PACING statement

Use the PACING statement of the LOAD control statement to adjust the rate at which messages are inserted back into your message queues.

The PACING statement is an optional LOAD control statement.

If the PACING statement is present, the logic inserts the messages back into the IMS message queue at the same rate at which they originally were inserted by the application, at a slower rate, or at a faster rate, based on what is specified in the PACING control statement.

Guidelines for coding the PACING statement

- Specify a decimal number from 0.1 to 9.9; you must include the decimal point.
- A value of 1.0 will insert the messages back into the IMS message queue at the same rate at which they were originally inserted.

  A higher number, such as 4.0, slows the insertion rate proportionately (in this case, four times slower).

  A value less than 1 inserts the messages faster.
- When pacing is used with the QCF LOAD function and IMS crashes, messages might be lost because IMS does not write the data into the current OLDS buffer.

- The PACING statement can be used in conjunction with the DEBUG statement to produce the Pacing Timing report.
- You can also set a limit on the wait time using the MAXWAIT control statement.

Syntax diagram for PACING statement

►►─ PACING= ─── *x.y* ─►◄

MAXMSGCT statement

Use the MAXMSGCT statement of the LOAD function control statement to specify the maximum number of messages that you want inserted to any particular destination.

The MAXMSGCT statement is an optional LOAD control statement.

The MAXMSGCT statement allows you to specify the maximum number of messages that you want inserted for any one destination.

If the MAXMSGCT control statement is present and if the limit is reached, subsequent messages are scrapped with a return code of X'F0' and a reason code of X'FFFF'.

The logic records the return code and reason code in the QCF prefix, and writes the QCF prefix and the scrapped message to the SCRAPLOG data set.

Restriction: The MAXMSGCT statement is ignored if CHNGDEST is also coded.

Syntax diagram for MAXMSGCT statement

►►─ MAXMSGCT= ─── *ddddddd* ─►◄

where *ddddddd* is a value between 1 and 9999999.

MAXWAIT statement

Use the MAXWAIT statement of the LOAD function to set the maximum time that the PACING function should wait between inserting messages.

The MAXWAIT statement is an optional LOAD control statement.

Set the MAXWAIT statement to specify the maximum time that the PACING function should wait between inserting messages.

If MAXWAIT is present and if the time that is calculated by the PACING function exceeds the time that was supplied on the MAXWAIT control statement, the time that you supply will be used.

Syntax diagram for MAXWAIT statement

►►─ MAXWAIT= ─── *hh:mm:ss* ─►◄

Variable reference for MAXWAIT statement

The variable fields are defined as follows:

hh
    Hours, which is a value between 0 and 24.

mm
    Minutes, which is a value between 0 and 60.

ss
    Seconds, which is a value between 0 and 60.

Example specifications for MAXWAIT

- MAXWAIT

  (that indicates that the default MAXWAIT time, 2 hours, is to be used)
- MAXWAIT=hh

  (that indicates that the MAXWAIT time will be *hh* hours)
- MAXWAIT=hh:mm

  (that indicates that the MAXWAIT time will be *hh* hours and *mm* minutes)
- MAXWAIT=hh:mm:ss

  (that indicates that the MAXWAIT time will be *hh* hours,*mm* minutes, and *ss* seconds)
- MAXWAIT=:mm

  (that indicates that the MAXWAIT time will be *mm* minutes. The colon (:) is required to indicate that there are no hours specified.)
- MAXWAIT=:mm:ss

  (that indicates that the MAXWAIT time will be *mm* minutes and *ss* seconds)
- MAXWAIT=::ss

  (that indicates that the MAXWAIT time will be *ss* seconds. The two colons (::) are required to indicate that there are no hours and no minutes specified.)

WAIT statement

Use the WAIT statement of the LOAD function control statement to specify how long the PACING function waits between message insertions.

The WAIT statement is optional.

Guidelines for coding the WAIT statement

- If you issue a WAIT statement, issue a PACING statement as well, to avoid a buffer overflow.

  Specifying both statements causes IMS to enqueue the messages to the final destination on each INSERT while ignoring the parameters specified in the PACING statement.

  During this process, IMS Queue Control Facility simulates an express PCB by issuing an ISRT or PURG request for each message.

  The ISRT or PURG request puts the message on the SQ and releases the incore buffers.
- When messages are inserted without issuing a PACING statement, IMSQueue Control Facility places all messages in the incore buffers.

  If too many messages are inserted, an overflow occurs in the incore buffers.
- The WAIT statement lets you specify the fixed wait time for the PACING function between inserting messages.

  If WAIT is specified, PACING should also be specified, even though pacing parameters are ignored.
- The WAIT and MAXWAIT statements are mutually exclusive.

  WAIT is a signal to IMS to enqueue the messages to the final destination on each LOAD.
- You must restart the client with the correct set of input when using any PACING options.

Syntax diagram for WAIT statement

▶▶─ WAIT= ── *hh:mm:ss.ijklmn* ─ ▶◀

Variable reference for WAIT statement

hh
Hours, which is a value between 0 and 1.

mm
Minutes, which is a value between 0 and 60.

ss
Seconds, which is a value between 0 and 60.

ijklmn
Fractions of seconds, which can be specified as:

i
Tenths of a second

j
Hundredths of a second

k
Milliseconds

l
Tenths of a millisecond

m
Hundredths of a millisecond

n
Microseconds

Example specifications for WAIT

WAIT
Indicates that the default WAIT time 00:00:00.000000 is to be used.

WAIT=:*mm*:*ss*.*ijklmn*
Indicates that the WAIT time will be *mm* minutes, *ss* seconds, and *ijklmn* fractions of seconds.

WAIT=::*ss*.*ijklmn*
Indicates that the WAIT time will be *ss* seconds.

The two colons (::) are required to indicate that no hours or minutes are specified.

JCL for the LOAD function

Use the LOAD step to reload messages to your shared queues.

The purpose of the LOAD step is to reinsert messages to the shared queues.

You can run the LOAD function in batch mode by submitting a JCL job stream.

Sample JCL to run the LOAD function

The following example shows the JCL that is required to start a LOAD step. Replace the items in italics with values that are appropriate for your environment:

```
//***********************************************************************
//* SAMPLE JCL TO EXECUTE THE LOAD FUNCTION                             *
//***********************************************************************
//LOAD     EXEC PGM=IQCINI0$,
//     PARM='BPEINIT=IQCBINI0,BPECFG=IQCIVPCF,IMSPLEX=IPLEX,QCFIMS=IMS1'
//STEPLIB  DD  DISP=SHR,DSN=hlqual.SIQCLINK
//         DD  DISP=SHR,DSN=IMSREL.RESLIB
//*
//QCFPRINT DD SYSOUT=A
//*
//LOAD     DD DISP=SHR,DSN=QCF.UNLOAD
//*
//* SCRAPLOG IS AN OPTIONAL DD STATEMENT. IT WILL CONTAIN THE
```

```
//* REJECTED MESSAGES FROM THE LOAD STEP.
//* DCB PARAMETERS SHOULD BE THE SAME AS THOSE FOR THE LOAD
//* DATA SET.
//* HOWEVER, IT IS RECOMMENDED THAT THE MAXIMUM SIZES FOR LRECL
//* AND BLKSIZE ARE USED.
//*
//SCRAPLOG DD DSN=QCF.SCRAPLOG,UNIT=SYSDA,DISP=(NEW,CATLG),
//         DCB=(LRECL=32756,BLKSIZE=32760,RECFM=VB),
//         SPACE=(CYL,(5,5))
//*
//SNAPDUMP DD SYSOUT=A
//*
//QCFIN    DD DSN=USER.PRIVATE.PROCLIB(load),UNIT=SYSDA
```

DD statements for LOAD

STEPLIB DD
Defines the library containing the IMS Queue Control Facility program load modules and IMS libraries that are required for the load step to run. Required.

This DD statement is required.

QCFPRINT DD
Defines the output report data set.

This DD statement is required.

LOAD DD
Defines the input data set. This can be the output from BROWSE or the UNLOAD function.

This DD statement is required.

SCRAPLOG DD
Defines an output data set to receive message records that are rejected in the LOAD step.

This DD statement is optional.

SNAPDUMP DD
Defines an output data set for collecting debugging information.

This DD statement is optional. However, the statement is required if the DEBUG control statement is present.

QCFIN DD
Defines the input control statement data set.

This DD statement is required.

Example of the PROC USER.PRIVATE.PROCLIB(*load*)

The following example shows the PROC that was specified above in the QCFIN DD statement as, USER.PRIVATE.PROCLIB(*load*):

```
**********************************************************
* load all of the messages in the data set
**********************************************************
function load queuetype=all
end
```

Creating the IQCLODX0 user exit routine

Create your own LOAD function user exit routine named IQCLODX0 for different IMSPLEX systems.

Example BPE user exit routine list PROCLIB member with exit member definition

The following example shows a BPE user exit routine list PROCLIB member with exit member definition:

```
***********************************************************************
* BPE USER EXIT LIST PROCLIB MEMBER                                   *
***********************************************************************
#--------------------------------------------------------------------#
#           DEFINE LOAD USER EXIT:  IQCLODX0                          #
#--------------------------------------------------------------------#
EXITDEF(TYPE=LOADX,EXITS=(IQCLODX0))
```

Example server configuration PROCLIB member with IQCLODX0 defined

The following example shows the server configuration PROCLIB member with IQCLODX0 defined:

```
***********************************************************************
* SAMPLE QCF SERVER CONFIGURATION PROCLIB MEMBER                      *
***********************************************************************
*--------------------------------------------------------------------*
* 1- to 5-character IMSPLEX(name). This name represents the          *
* IMS Plex with which QCF is communicating and does not represent the *
* the IMS image.                                                     *
*--------------------------------------------------------------------*
IMSPLEX(IPLEX)                        /* QCF Target Member Name = IPLEX */
*--------------------------------------------------------------------*
* Security can be turned on if the AUTH=Y parameter is added.        *
* The TSO user must be defined to the MVS of the server system.      *
* This is based on the MVS name, IMS name and function being         *
* performed.                                                         *
*--------------------------------------------------------------------*
AUTH(Y|N)              /* On: indicates that SAF call will be used to
                                  validate TSO requests by function. */
```

Example server configuration for BPE with server address space defined

The following example shows the server configuration for BPE with server address space defined:

```
***********************************************************************
* SAMPLE CONFIGURATION FILE FOR BPE WITH QCF SERVER ADDRESS SPACE    *
***********************************************************************
*--------------------------------------------------------------------*
* Specify the language for messages. Syntax is:                      *
*                                                                    *
* LANG=language                                                      *
*                                                                    *
* language = 3-character code for the requested language. Currently, *
* only ENU (U.S. English) is supported.                             *
*--------------------------------------------------------------------*
LANG=ENU                                  /* Language for messages */
                                          /* (ENU = U.S. English)  */
*--------------------------------------------------------------------*
* Trace table level definitions. Syntax is:                         *
*                                                                    *
* TRCLEV=(tablename,level,component) -or-                           *
* TRCLEV=(tablename,level,component,PAGES=#pages)                   *
*                                                                    *
* tablename = name of the trace table, or * to set default for all  *
* level = tracing level: NONE, ERROR, LOW, MEDIUM, or HIGH          *
* component = owning component name (BPE or QCF)                    *
* #pages = (optional) # of 4K pages to be allocated to this trace   *
* table                                                             *
*--------------------------------------------------------------------*
#
# Definitions for BPE system traces
#
TRCLEV=(*,HIGH,BPE)                       /* Set default for all BPE  */
                                          /* traces to HIGH. Uncomment */
```

```
                                        /* lines below and fill in      */
                                        /* level to change particular */
                                        /* table levels to other than */
                                        /* HIGH.                       */
#TRCLEV=(AWE,level,BPE)                  /* AWE server trace           */
#TRCLEV=(CBS,level,BPE)                  /* Control blk srvcs trace    */
#TRCLEV=(CMD,level,BPE)                  /* Command services trace     */
#TRCLEV=(DISP,level,BPE)                 /* Dispatcher trace           */
#TRCLEV=(HASH,level,BPE)                 /* Hash table trace           */
#TRCLEV=(LATC,level,BPE)                 /* Latch trace                */
#TRCLEV=(SSRV,level,BPE)                 /* System services trace         */
#TRCLEV=(STG,level,BPE)                  /* Storage trace              */
#TRCLEV=(USRX,level,BPE)                 /* User exit services trace   */
#
# Definitions for QCF traces
#
TRCLEV=(*,HIGH,QCF)                      /* Set default for all QCF    */
                                        /* traces to HIGH. Uncomment  */
                                        /* lines below and fill in    */
                                        /* level to change particular */
                                        /* table levels to other than */
                                        /* HIGH.                       */
#TRCLEV=(ERR,level,QCF)                  /* QCF error trace            */
#TRCLEV=(INI,level,QCF)                  /* QCF initialization trace   */
#TRCLEV=(MST,level,QCF)                  /* QCF master control trace   */
#TRCLEV=(CSV,level,QCF)                  /* QCF common service trace   */
#TRCLEV=(CVC,level,QCF)                  /* QCF conversation trace     */
#TRCLEV=(REQ,level,QCF)                  /* QCF request service trace  */
#TRCLEV=(RSP,level,QCF)                  /* QCF response service trace */
#TRCLEV=(SAF,level,QCF)                  /* QCF SFA manager trace         */
#TRCLEV=(CMD,level,QCF)                  /* QCF command processor trace*/
#TRCLEV=(SCI,level,QCF)                  /* QCF SCI trace              */
EXITMBR=(IQCEXIT0,QCF)                   /* EXIT MBR DEF               */
```

Pacing reports

Only when IBM Software Support instructs you to, use the pacing reports with the DEBUG statement on the LOAD function control statement to gather diagnostic information.

When you include the DEBUG statement in conjunction with the PACING statement, the LOAD function routine produces a pacing timing report and a load pacing report .

Pacing timing report

The LOAD function Pacing timing report that is shown here has two parts: the Control records and comments, and the Pacing timing report.

Use the LOAD function Pacing timing report only to obtain diagnostic information at the request of IBM Software Support personnel.

The following sample report shows an example of a Control records and comments report section and then an example of a Pacing timing report. Detailed descriptions of the lines in the Pacing timing report are shown in the following Pacing timing report example.

```
Page    1                        IMS Queue Control Facility V3R2 (5697-N50)
Report: CQSCtrl001                         CQS Controller Routine
                 CQS Controller Control Records and Comments

                 _____
                 TITLE='QCF COLDQ TEST'
                 PACING=0.1
         IQC4079I Pacing now enabled.
                 MAXWAIT=00:00:10
         IQC4080I MAXWAIT now enabled.
                 DEBUG
```

Figure 13. CQS Controller Control Records and Comments

```
 Pacing initialized.  First record processing finished.  IMS=V15.1.0 LogDate=2020/11/05   LogTime=01:37:17.176178(B31983AEC5D72D08
  >> Next msg-schedule diff:  0000000C157F039F    real time difference:  000000000024DD40
Differential=00:02:06.709602(00000078D6F6243C) NewDiff =00:00:12.670960(0000000C157F039F) LogTime=01:39:23.885781(B31984279CCD5144
MaxWait exceeded, using user specified value. New WaitTime=00:00:10:000000(0000000989680000)
Current time before wait=15:35:50.125189     WaitTime=00:00:09.524586            InsTime=15:35:59.669176
  >> Next msg-schedule diff:  00000036091E81FF    real time difference:  000000091B66C6C7
Differential=00:07:19.894958(000001A3843AEFC0) NewDiff =00:00:43.989495(00000029F39F7E60) LogTime=01:46:43.780740(B31985CB21084104
MaxWait exceeded, using user specified value. New WaitTime=00:00:10:000000(0000000989680000)
Current time before wait=15:35:59.674028     WaitTime=00:00:10.430225            InsTime=15:36:10.106454
  >> Next msg-schedule diff:  00000036095B45F9    real time difference:  000000130FBEA989
Differential=00:00:00.009722(00000000025FA7C5) NewDiff =00:00:00.000972(00000000003CC3FA) LogTime=01:46:43.790462(B31985CB2367E8C9
Current time before wait=15:36:10.112033     No wait, differential = 0           InsTime=15:36:10.112042
  >> Next msg-schedule diff:  0000003609AA9646    real time difference:  000000131056F289
Differential=00:00:00.012690(00000000031922FC) NewDiff =00:00:00.001269(00000000004F504C) LogTime=01:46:43.803152(B31985CB26810BC5
Current time before wait=15:36:10.114465     No wait, differential = 0           InsTime=15:36:10.114473
  >> Next msg-schedule diff:  000000360A1F35AC    real time difference:  0000001313731A49
Differential=00:00:00.018659(00000000048E3A01) NewDiff =00:00:00.001865(0000000000749F66) LogTime=01:46:43.821812(B31985CB2B0F45C6
Current time before wait=15:36:10.127208     No wait, differential = 0           InsTime=15:36:10.127216
  >> Next msg-schedule diff:  000000360A8EBE33    real time difference:  000000131838D383
Differential=00:00:00.017845(00000000045B5541) NewDiff =00:00:00.001784(00000000006F8886) LogTime=01:46:43.839657(B31985CB2F6A9B07
Current time before wait=15:36:10.146787     No wait, differential = 0           InsTime=15:36:10.146796
  >> Next msg-schedule diff:  000000360AAF2FD2    real time difference:  00000013197DC4C3
Differential=00:00:00.005191(000000000144703A) NewDiff =00:00:00.000519(000000000020719F) LogTime=01:46:43.844848(B31985CB30AF0B41
Current time before wait=15:36:10.151951     No wait, differential = 0           InsTime=15:36:10.151961
  >> Next msg-schedule diff:  000000360ACE86DF    real time difference:  00000013289C7609
Differential=00:00:00.005014(0000000001396684) NewDiff =00:00:00.000501(00000000001F570D) LogTime=01:46:43.849863(B31985CB31E871C5
Current time before wait=15:36:10.213887     No wait, differential = 0           InsTime=15:36:10.213895
  >> Next msg-schedule diff:  000000360AED050C    real time difference:  00000013294CF1C9
Differential=00:00:00.004878(000000000130EDC3) NewDiff =00:00:00.000487(00000000001E7E2D) LogTime=01:46:43.854741(B31985CB33195F88
Current time before wait=15:36:10.216705     No wait, differential = 0           InsTime=15:36:10.216713
  >> Next msg-schedule diff:  000000360B1A612C    real time difference:  0000001335F92941
Differential=00:00:00.007257(0000000001C59939) NewDiff =00:00:00.000725(00000000002D5C1F) LogTime=01:46:43.861999(B31985CB34DEF8C1
Current time before wait=15:36:10.268614     No wait, differential = 0           InsTime=15:36:10.268623
  >> Next msg-schedule diff:  000000360F5C3852    real time difference:  00000013378C0341
Differential=00:00:00.174374(000000002A926781) NewDiff =00:00:00.017437(000000000441D726) LogTime=01:46:44.036374(B31985CB5F716042
Current time before wait=15:36:10.275065     WaitTime=00:00:00.010996            InsTime=15:36:10.287580
  >> Next msg-schedule diff:  000000360FB05C53    real time difference:  000000133B422505
Differential=00:00:00.013462(0000000003496805) NewDiff =00:00:00.001346(0000000000542400) LogTime=01:46:44.049836(B31985CB62BAC847
Current time before wait=15:36:10.290262     No wait, differential = 0           InsTime=15:36:10.290270
  >> Next msg-schedule diff:  000000360FD3ECCC    real time difference:  000000133CC46F85
Differential=00:00:00.005690(000000000163A4C1) NewDiff =00:00:00.000569(0000000000239079) LogTime=01:46:44.055526(B31985CB641E6D08
Current time before wait=15:36:10.296443     No wait, differential = 0           InsTime=15:36:10.296452
  >> Next msg-schedule diff:  000000360FFC299F    real time difference:  000000133D7A19C5
Differential=00:00:00.006438(000000000192603F) NewDiff =00:00:00.000643(0000000000283CD3) LogTime=01:46:44.061964(B31985CB65B0CD47
Current time before wait=15:36:10.299347     No wait, differential = 0           InsTime=15:36:10.299355
  >> Next msg-schedule diff:  0000003610221393    real time difference:  000000133DFD8485
Differential=00:00:00.006066(00000000017B2381) NewDiff =00:00:00.000606(000000000025E9F3) LogTime=01:46:44.068031(B31985CB672BF0C8
Current time before wait=15:36:10.301451     No wait, differential = 0           InsTime=15:36:10.301459
  >> Next msg-schedule diff:  00000036104C9A32    real time difference:  000000133ECF8705
Differential=00:00:00.006804(0000000001A9423D) NewDiff =00:00:00.000680(000000000002A869F) LogTime=01:46:44.074835(B31985CB68D53305
Current time before wait=15:36:10.304813     No wait, differential = 0           InsTime=15:36:10.304823
  >> Next msg-schedule diff:  00000036109C8C73    real time difference:  0000001340B10C83
```

Figure 14. Pacing Timing report

Each item in the Pacing Timing report consists of three or four lines:

The **first line** contains data that is used to decide whether or not the message should wait.

Next Msg-Schedule Diff:
> This is the difference between the current record timestamp and the first record timestamp multiplied by the pacing x.y factors. The pacing x.y factors determine what the time interval should be between the arrival of the first and the current message.

Real Time Difference=
> This is the difference between the current time and the time the first record arrived. If for some reason, the message arrives later than scheduled, there is no wait. (If the schedule difference is less than the real time difference, the message does not wait.) Both differences are in hexadecimal store clock format.

> If the time difference is less than 0, the first line displays `Negative time differential --- will not wait.`

The **second line** contains the data that is used to provide the wait time:

Differential=
> This is the differential between the current record's timestamp and the previous record's timestamp.

> • The first number is the differential in *hh:mm:ss.thmiju* format.

> • The number in parentheses is the differential in hexadecimal store clock format.

NewDiff=
> This is how long the wait will be; this is the differential multiplied by the PACING x.y factors.

- The first number is the wait time in *hh:mm:ss.thmiju* format.
- The number in parentheses is the wait time in hexadecimal store clock format.

LogTime=
  This is the time at which the message was originally queued.

- The first number is the time at which the message was logged, in *hh:mm:ss.thmiju* format.
- The number in parentheses is the log time in hexadecimal store clock format.

The **third line** is shown only if the message is on the schedule and should wait, but the calculated wait time (NewDiff) is greater than the MaxWait that you specified.

MaxWait Exceeded, Using User-Specified Value

NewWaitTime=
  This is how long the wait will be.

- The first number is the wait time in *hh:mm:ss.thmiju* format.
- The number in parentheses is the wait time in hexadecimal store clock format.

The **last line** contains three times, as follows:

Current time before wait=
  This is the time at which the wait was issued, in *hh:mm:ss.thmiju* local time.

New WaitTime=
  This is how long the wait will be. The NewDiff time is corrected with the time expired to the previous message, and with the time the current message is late on input.

  If the current message is late on input, the following text displays: `No wait, out of schedule`.

  If the WaitTime is less than or equal to 0 after the correction, the following text displays: `No wait, differential=0`.

InsTime=
  This is the time when insertion of the message starts. All three times are in *hh:mm:ss.thmiju* format.

Load Pacing Timing report

The Load Pacing Timing report is preceded by the CQS controller control records and comments.

The Load Pacing Timing report consists of CQS controller control records and comments followed by a Load Pacing Timing report. An example of the CQS controller control records and comments is shown in the following figure. After the CQS controller control records and comments, an example Load Pacing Timing report is shown.

The example Load Pacing Timing report is followed by descriptions of the lines in the report.

```
Page  1              IMS Queue Control Facility V3R2 (5697-N50)           System Date: 2020.285
Report: CQSCtrl001                       CQS Controller Routine           System Time: 12.08.43
                 CQS Controller Control Records and Comments

                 ═════════════════════════════════════════
                 title='Load Pacing Report'
                   fu lo qt=all
                   pacing=0.1
         IQC4079I PACING now enabled.
                   debug
         IQC4037I DEBUGging now enabled.
                   mw=::10
         IQC4080I MAXWAIT now enabled.
         IQC2500I CQS Load Routine
```

Figure 15. CQS Controller Control Records and Comments

A Load Pacing Timing report is shown as follows:

```
Page   2                              IMS Queue Control Facility V3R2 (5697-N50)        System Date:  2020.291
Report: Load008                               Pacing Timing Report                            System Time:  15.35.50
 Pacing initialized.  First record processing finished.  IMS=V15.1.0 LogDate=2020/11/05  LogTime=01:37:17.176178(B31983AEC5D72D08
  >> Next msg-schedule diff:  0000000C157F039F    real time difference:  000000000024DD40
 Differential=00:02:06.709602(00000078D6F6243C) NewDiff =00:00:12.670960(0000000C157F039F) LogTime=01:39:23.885781(B31984279CCD5144
 MaxWait exceeded, using user specified value. New WaitTime=00:00:10:000000(0000000989680000)
 Current time before wait=15:35:50.125189       WaitTime=00:00:09.524586                    InsTime=15:35:59.669176
  >> Next msg-schedule diff:  00000036091E81FF    real time difference:  000000091B66C6C7
 Differential=00:07:19.894958(000001A3843AEFC0) NewDiff =00:00:43.989495(00000029F39F7E60) LogTime=01:46:43.780740(B31985CB21084104
 MaxWait exceeded, using user specified value. New WaitTime=00:00:10:000000(0000000989680000)
 Current time before wait=15:35:59.674028       WaitTime=00:00:10.430225                    InsTime=15:36:10.106454
  >> Next msg-schedule diff:  00000036095B45F9    real time difference:  000000130FBEA989
 Differential=00:00:00.009722(00000000025FA7C5) NewDiff =00:00:00.000972(00000000003CC3FA) LogTime=01:46:43.790462(B31985CB2367E8C9
 Current time before wait=15:36:10.112033       No wait, differential = 0                    InsTime=15:36:10.112042
  >> Next msg-schedule diff:  0000003609AA9646    real time difference:  000000131056F289
 Differential=00:00:00.012690(00000000031922FC) NewDiff =00:00:00.001269(00000000004F504C) LogTime=01:46:43.803152(B31985CB26810BC5
 Current time before wait=15:36:10.114465       No wait, differential = 0                    InsTime=15:36:10.114473
  >> Next msg-schedule diff:  000000360A1F35AC    real time difference:  0000001313731A49
 Differential=00:00:00.018659(00000000048E3A01) NewDiff =00:00:00.001865(0000000000749F66) LogTime=01:46:43.821812(B31985CB2B0F45C6
 Current time before wait=15:36:10.127208       No wait, differential = 0                    InsTime=15:36:10.127216
  >> Next msg-schedule diff:  000000360A8EBE33    real time difference:  000000131838D383
 Differential=00:00:00.017845(00000000045B5541) NewDiff =00:00:00.001784(00000000006F8886) LogTime=01:46:43.839657(B31985CB2F6A9B07
 Current time before wait=15:36:10.146787       No wait, differential = 0                    InsTime=15:36:10.146796
  >> Next msg-schedule diff:  000000360AAF2FD2    real time difference:  00000013197DC4C3
 Differential=00:00:00.005191(000000000144703A) NewDiff =00:00:00.000519(000000000020719F) LogTime=01:46:43.844848(B31985CB30AF0B41
 Current time before wait=15:36:10.151951       No wait, differential = 0                    InsTime=15:36:10.151961
  >> Next msg-schedule diff:  000000360ACE86DF    real time difference:  00000013289C7609
 Differential=00:00:00.005014(0000000001396684) NewDiff =00:00:00.000501(00000000001F570D) LogTime=01:46:43.849863(B31985CB31E871C5
 Current time before wait=15:36:10.213887       No wait, differential = 0                    InsTime=15:36:10.213895
  >> Next msg-schedule diff:  000000360AED050C    real time difference:  00000013294CF1C9
 Differential=00:00:00.004878(000000000130EDC3) NewDiff =00:00:00.000487(00000000001E7E2D) LogTime=01:46:43.854741(B31985CB33195F88
 Current time before wait=15:36:10.216705       No wait, differential = 0                    InsTime=15:36:10.216713
  >> Next msg-schedule diff:  000000360B1A612C    real time difference:  0000001335F92941
 Differential=00:00:00.007257(0000000001C59939) NewDiff =00:00:00.000725(00000000002D5C1F) LogTime=01:46:43.861999(B31985CB34DEF8C1
 Current time before wait=15:36:10.268614       No wait, differential = 0                    InsTime=15:36:10.268623
  >> Next msg-schedule diff:  000000360F5C3852    real time difference:  00000013378C0341
 Differential=00:00:00.174374(000000002A926781) NewDiff =00:00:00.017437(000000000441D726) LogTime=01:46:44.036374(B31985CB5F716042
 Current time before wait=15:36:10.275065       WaitTime=00:00:00.010996                    InsTime=15:36:10.287580
  >> Next msg-schedule diff:  000000360FB05C53    real time difference:  000000133B422505
 Differential=00:00:00.013462(0000000003496805) NewDiff =00:00:00.001346(0000000000542400) LogTime=01:46:44.049836(B31985CB62BAC847
 Current time before wait=15:36:10.290262       No wait, differential = 0                    InsTime=15:36:10.290270
  >> Next msg-schedule diff:  000000360FD3ECCC    real time difference:  000000133CC46F85
 Differential=00:00:00.005690(000000000163A4C1) NewDiff =00:00:00.000569(0000000000239079) LogTime=01:46:44.055526(B31985CB641E6D08
 Current time before wait=15:36:10.296443       No wait, differential = 0                    InsTime=15:36:10.296452
  >> Next msg-schedule diff:  000000360FFC299F    real time difference:  000000133D7A19C5
 Differential=00:00:00.006438(000000000192603F) NewDiff =00:00:00.000643(0000000000283CD3) LogTime=01:46:44.061964(B31985CB65B0CD47
 Current time before wait=15:36:10.299347       No wait, differential = 0                    InsTime=15:36:10.299355
  >> Next msg-schedule diff:  0000003610221393    real time difference:  000000133DFD8485
 Differential=00:00:00.006066(00000000017B2381) NewDiff =00:00:00.000606(000000000025E9F3) LogTime=01:46:44.068031(B31985CB672BF0C8
 Current time before wait=15:36:10.301451       No wait, differential = 0                    InsTime=15:36:10.301459
  >> Next msg-schedule diff:  00000036104C9A32    real time difference:  000000133ECF8705
 Differential=00:00:00.006804(0000000001A9423D) NewDiff =00:00:00.000680(00000000002A869F) LogTime=01:46:44.074835(B31985CB68D53305
 Current time before wait=15:36:10.304813       No wait, differential = 0                    InsTime=15:36:10.304823
  >> Next msg-schedule diff:  00000036109C8C73    real time difference:  0000001340B10C83
```

Figure 16. Load Pacing Timing report

Each item in the Load Pacing Timing report consists of three or four lines. The **first line** contains data that is used to decide whether the message should wait.

Next Msg-Schedule Diff:
This is the difference between the current record timestamp and the first record timestamp multiplied by the pacing x.y factors. The pacing x.y factors determine what the time interval should be between the arrival of the first and the current message.

Real Time Difference=
This is the difference between the current time and the time the first record arrived. If for some reason, the message arrives later than scheduled, there is no wait. (If the schedule difference is less than the real time difference, the message does not wait.) Both differences are in hexadecimal store clock format.

If the time difference is less than 0, the first line displays `Negative time differential --- will not wait`.

The **second line** contains the data that is used to provide the wait time:

Differential=
This is the differential between the current record's timestamp and the previous record's timestamp.

- The first number is the differential in *hh:mm:ss.thmiju* format.
- The number in parentheses is the differential in hexadecimal store clock format.

NewDiff=
This is how long the wait will be; this is the differential multiplied by the PACING x.y factors.

- The first number is the wait time in *hh:mm:ss.thmiju* format.
- The number in parentheses is the wait time in hexadecimal store clock format.

LogTime=
This is the time at which the message was originally queued.

- The first number is the time at which the message was logged, in *hh:mm:ss.thmiju* format.
- The number in parentheses is the log time in hexadecimal store clock format.

The **third line** is shown only if the message is on the schedule and should wait, but the calculated wait time (NewDiff) is greater than the MaxWait that you specified.

MaxWait Exceeded, Using User-Specified Value

NewWaitTime=
Specify how long the wait will be.

- The first number is the wait time in *hh:mm:ss.thmiju* format.
- The number in parentheses is the wait time in hexadecimal store clock format.

The **last line** contains three times, as follows:

Current time before wait=
This is the time at which the wait was issued, in *hh:mm:ss.thmiju* local time.

New WaitTime=
This is how long the wait will be. The NewDiff time is corrected with the time expired to the previous message, and with the time the current message is late on input.

If the current message is late on input, the following text displays: `No wait, out of schedule`.

If the WaitTime is less than or equal to 0 after the correction, the following text displays: `No wait, differential=0`.

InsTime=
This is the time when insertion of the message starts. All three times are in *hh:mm:ss.thmiju* format.

Chapter 13. Querying the message queues

The QUERY function allows you to query the message queues in both shared- and nonshared-queues environments.

Topics:

QUERY function description

The QUERY function determines the ages of the messages on different destinations queues and produces a data report.

You can run the QUERY functions through the IMS Queue Control Facility ISPF panels, as batch JCL job streams, and there are some special IMS Queue Control Facility server commands that you can run through the z/OS **MODIFY** command.

The QUERY function examines the IMS message queues for the age of messages on various destination queues that you specified. The QUERY function lists all message queues that have messages on them, and the age of the oldest and newest message.

The QUERY function produces reports of the data that it processes. The QUERY function differs from the BROWSE function in that the QUERY function cannot produce an output data set containing the messages that satisfy the request. The output QUERY data set contains only summary information about the destinations.

You can use the QUERY function to display the top *nn* destinations (any number up to 99) that have the most messages on the messages queue.

These destinations can be listed as a user-defined message queue utilization percentage when overflow protection starts user-defined procedures to browse or unload queues, or both browse and unload queues.

You must generate control cards for the QUERY function with the new parameter in the default card members. Panels have been changed to allow generation of the new parameter.

The destinations can be listed at any time without waiting to reach a defined percentage level. A user-defined procedure must be started to process the QUERY function with the new parameter.

The QUERY function selects information about a message's age from the message queues for purely informational purposes. The QUERY function performs the following tasks:

1. Determines queue selection from the QUEUETYPE= keyword of the FUNCTION statement
2. Determines message age selection criteria for message destinations from the AGE= keyword of the FUNCTION statement.
3. Discards or ignores records that fail to meet the selection criteria of the FUNCTION, INCLUDE, and EXCLUDE statements
4. Produces reports on the number of destination queues the QUERY function queried from the message queues
5. Returns control to IMS Queue Control Facility control processing logic

The job steps that are required for the QUERY function are: QUERY

QUERY processing

The QUERY function queries information about messages that are on the message queues.

Selection parameters

Based on your selection criteria, query logic selects information about a message's age from the message queues. You specify the selection criteria in control statements.

Selection criteria

Specify the selection criteria to query logic with FUNCTION, INCLUDE, and EXCLUDE statements.

Use these statements to define a subset of the information to be selected from the message queues. The INCLUDE and EXCLUDE statements are optional. If neither of these statements is present, all age data for all destinations from the message queues is selected.

Message age data can be selected based on the following criteria:

Specified in FUNCTION statement:

- Message age
- Queue type
  - ALL
  - APPC
  - DEADQ
  - LONG
  - LTERM
  - OTMA
  - REMOTE (remote MSNAME queue)
  - SHORT
  - SERIAL (serial transaction queue)
  - SUSPEND (suspend transaction queue)
  - TRANSACTION (transaction ready queue (local and remote))

Specified in INCLUDE or EXCLUDE statement:

- Destination

The FUNCTION statement, in the AGE= keyword, specifies how old a message must be to be processed. And in the QUEUETYPE= keyword, the FUNCTION statement specifies one or more message queues to be searched.

INCLUDE and EXCLUDE statements specify selection criteria that is specific to message destinations. INCLUDE and EXCLUDE statements are unrestricted in number.

If any INCLUDE statements are present, a record must meet all criteria for at least one INCLUDE statement to be included in the query logic selections. Unspecified criteria are considered to be met.

A record that meets all criteria that is specified by an EXCLUDE statement is excluded from the query logic selections, even if the record meets all of the criteria that is specified by the FUNCTION statement and an INCLUDE statement. Unspecified criteria are considered to be met.

Selection processing

IMS Queue Control Facility query logic calls IMS requesting that a QUERY function be invoked for the message queues that you specified.

When end-of-data is signaled by IMS, IMS Queue Control Facility query logic produces reports about the number of destination queues that QUERY found on the queues, and statistics about the number of records that QUERY processed.

IMS Queue Control Facility query logic then returns control to IMS Queue Control Facility control processing logic for clean up and normal termination.

Control statements for QUERY function

Use the QUERY function control statements to filter information about the messages on the message queues.

The QUERY function lets you query the IMS message queues through the use of control statements.

The QUERY function returns the number of messages, oldest message time stamp, and newest message time stamp for each QNAME that has messages on the QUEUETYPE meeting the selection criteria.

Recommendation: Do not specify QUERY without selection criteria. Be aware that performance can be poor because every object on every queue type is read. Identifying specific queue types using the QUEUETYPE operand on the FUNCTION statement minimizes the number of queues that are accessed. You can also improve performance by identifying specific destinations using the DESTINATION operand on the INCLUDE or EXCLUDE statement.

The control statements that are recognized by the QUERY function are:

```
* comment
AGE
AREA
EXCLUDE
INCLUDE
QUEUETYPE
RCT
TOPAM
TOPLA
TOPLM
TOPSM
TITLE
```

QUERY control statement abbreviations

You can abbreviate the specification of some of the QUERY control statements.

The following table shows a list of QUERY keyword abbreviations that are allowed by the IMS Queue Control Facility control processing interpreter:

Table 16. IMS QCF control processing QUERY keyword abbreviations

Keyword	Abbreviation
AREA	AR
EXCLUDE	EX
INCLUDE	IN
QUEUETYPE	QT

FUNCTION QUERY statement syntax

The FUNCTION QUERY control statement must follow certain syntactic specifications.

The syntax of the FUNCTION QUERY control statement is:

```
►►── FUNCTION ── QUERY ──────────────────────────────────────────────►
                      └─ AGE= nnnnnnnn , ─┬─ AGEUNITS=DAYS, ──┬──┘
                                          └─ AGEUNITS=SECONDS, ─┘

►─────────────────────────────────────────────────────►
  │                    ┌─ , choose one or more ◄─┐  │
  └─ AREA=( ──┬─────────┬── ALL ──┬───────────────── ),─┘
                        ├── GLOBAL ──┤
                        ├── LOCAL ──┤
                        ├── OVERFLOW ──┤
                        └── QSN ──┘

►──────────────────────────────────────────────────────────────►
  │                        ┌─ , choose one or more ◄─┐   └─ SBMPxx, ¹ ─┘
  └─ QUEUETYPE=( ──┬────────── ALL ──────┬── ),─┘
                            ├── APPC ──┤
                            ├── LONG ──┤
                            ├── LTERM ──┤
                            ├── DEADQ ──┤
                            ├── OTMA ──┤
                            ├── REMOTE ──┤
                            ├── SHORT ──┤
                            ├── SERIAL ──┤
                            ├── SUSPEND ──┤
                            └── TRANSACTION ──┘

►──────────────────────────────────────────────►◄
  └─ RCT= Y, ─┘  ┌── TOPAM= nn, ──┐
                 ├── TOPLA= nn, ──┤
                 ├── TOPLM= nn, ──┤
                 └── TOPSM= nn ──┘
```
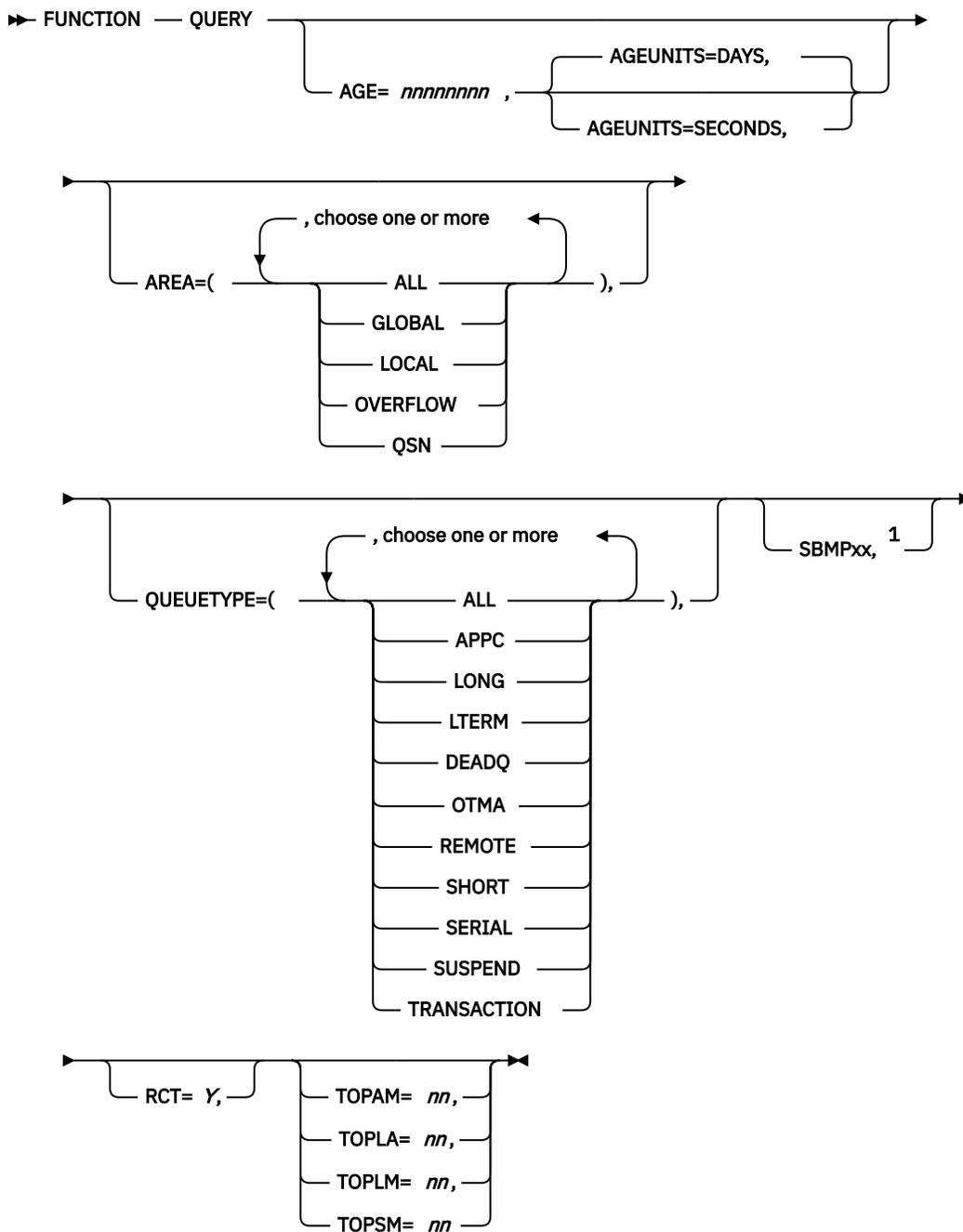
Notes:

  [1] Required for all jobs that are started because a threshold was crossed.

FUNCTION QUERY EXCLUDE statement syntax

You can use the EXCLUDE statement in conjunction with the FUNCTION QUERY statement.

In the JCL, the EXCLUDE control statement is written on a separate line after the FUNCTION QUERY control statement.

JCL example 1

```
FUNCTION QUERY QUEUETYPE=(ALL)
EXCLUDE DESTINATION=A*
END
```

JCL example 2

```
FUNCTION QUERY AREA=(ALL),QUEUETYPE=(ALL)
EXCLUDE DESTINATION=A*,
IMSIDS=IMS1
END
```

Syntax diagram
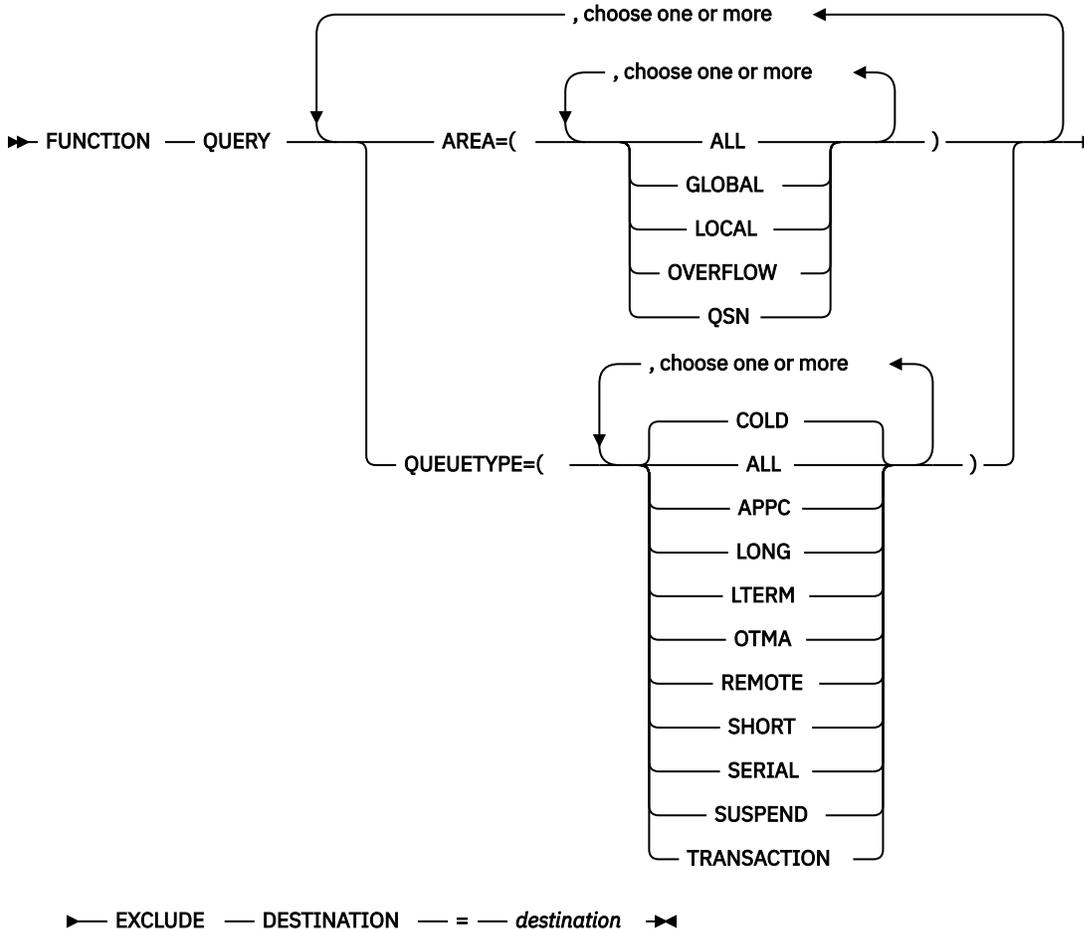
FUNCTION QUERY INCLUDE statement syntax

You can use the INCLUDE statement in conjunction with the FUNCTION QUERY statement.

In the JCL, the INCLUDE control statement is written on a separate line after the FUNCTION QUERY control statement.

JCL example 1

```
FUNCTION QUERY QUEUETYPE=(ALL)
INCLUDE DESTINATION=A*
END
```

JCL example 2

```
FUNCTION QUERY AREA=(ALL),QUEUETYPE=(ALL)
INCLUDE DESTINATION=A*,
IMSIDS=IMS1
END
```

Syntax diagram
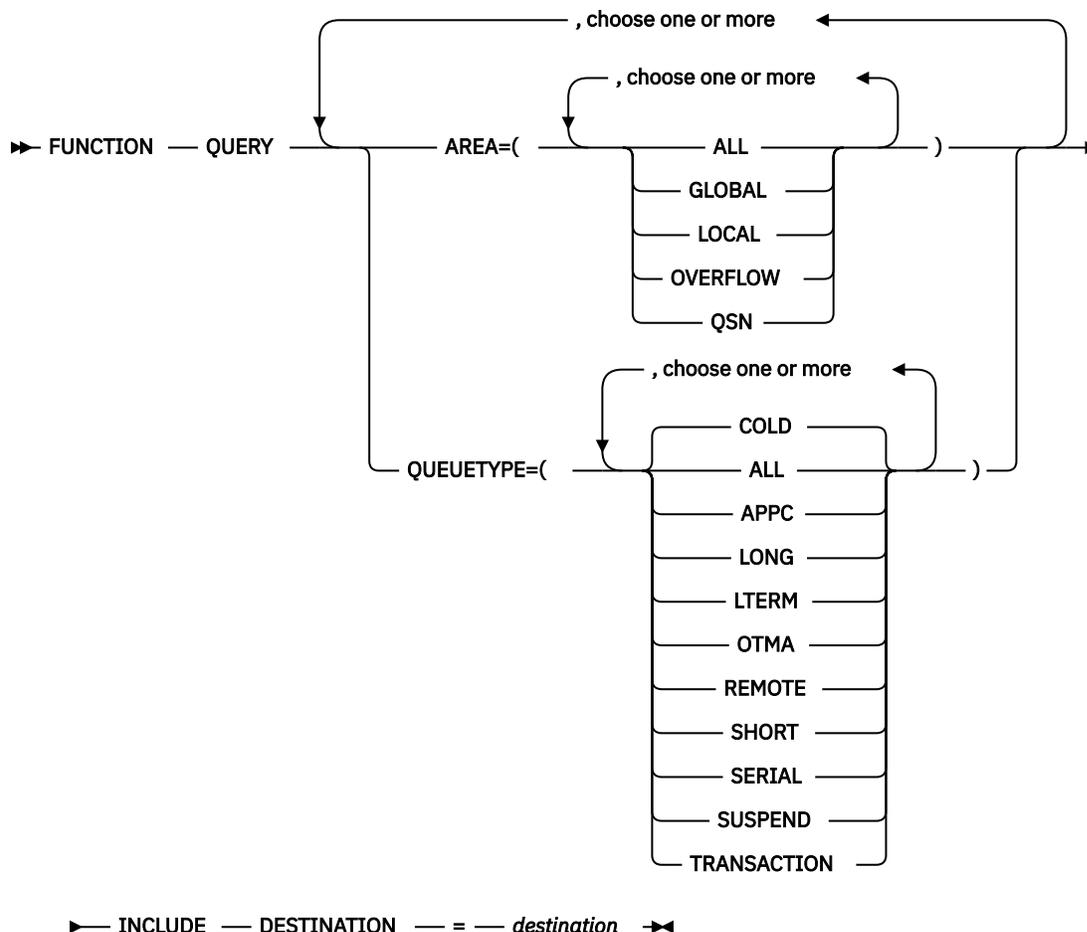
JCL for the QUERY function

The QUERY step runs to determine the ages of messages in the shared queues.

You can run the QUERY function in batch mode by submitting a JCL job stream.

Sample JCL to run the QUERY function

The following example shows the JCL that is required to start a QUERY step. Replace the items in italics with values that are appropriate for your environment:

```
//**********************************************************************
//* SAMPLE JCL TO EXECUTE THE QUERY FUNCTION                          *
//**********************************************************************
//QUERY    EXEC PGM=IQCINI0$,
//     PARM='BPEINIT=IQCBINI0,BPECFG=IQCIVPCF,IMSPLEX=IPLEX,QCFIMS=IMS1'
//STEPLIB  DD  DISP=SHR,DSN=hlqual.SIQCLINK
//         DD  DISP=SHR,DSN=IMSREL.RESLIB
//*
//QCFPRINT DD SYSOUT=A
//*
//SNAPDUMP DD SYSOUT=A
//*
//QUERY    DD   DSN=USRT002.BATCH.QUERY,
//         DSN=QCF.QUERY,UNIT=SYSDA,DISP=(NEW,CATLG),
//         UNIT=SYSDA,VOL=SER=222222,
//         DCB=(LRECL=32756,BLKSIZE=32760,RECFM=VB),
//         SPACE=(TRK,(5,1))
//*
//QCFIN    DD DSN=USER.PRIVATE.PROCLIB(query),UNIT=SYSDA
```

DD statements for QUERY

STEPLIB DD
> Defines the library containing the QCF program load modules and IMS libraries that are required for the QUERY step to run.
>
> This DD statement is required.

QCFPRINT DD
> Defines the output report data set.
>
> This DD statement is required.

QUERY DD
> Defines an output data set.
>
> This DD statement is required.

QCFIN DD
> Defines the input control statement data set.
>
> This DD statement is required.

Example of the PROC USER.PRIVATE.PROCLIB(*query*)

The following example shows the PROC that was specified above in the QCFIN DD statement as, USER.PRIVATE.PROCLIB(*query*):

```
**********************************************************
* query all the messages with age=2
**********************************************************
function query queuetype=all
   age=2
end
```

Showing messages queue utilization and last active destinations

You can use the QUERY function to show short and long messages queue utilization and last active destinations.

For nonshared queues, IMS Queue Control Facility can display the top *nn* destinations using the short or the long message queue, both short and long messages queues, or last active destinations.

For shared queues, IMS Queue Control Facility can display the top *nn* destinations using the messages queues or last active destinations.

The following mutually exclusive parameters can be used to display the top *nn* destinations:

- TOPLASTACTIVWTO (TOPLA)
- TOPALLMSGWTO (TOPAM)
- TOPLONGMSGWTO (TOPLM)
- TOPSHORTMSGWTO (TOPSM)

WTO output for top last active destinations

You can get WTO (wait to operator) output by using the TOPLA parameter.

The following figure shows sample WTO output when the TOPLA parameter is used:

```
11.03.25 STC00227  +IQC3522I Top           9            last Active   Destinations
                                                        LastActive    LongB/ShrtB
11.03.25 STC00227  +IQC3523I Dest        QT  06/01/10-14:11:27:9    0.2%/0.0%
11.03.25 STC00227  +IQC3524I AD201L5     LT  06/01/10-14:11:27:9    0.2%/0.0%
11.03.25 STC00227  +IQC3524I NDS01B      LT  06/01/10-14:11:27:9    0.2%/0.0%
11.03.25 STC00227  +IQC3524I RTERM07     LT  06/01/10-14:11:27:9    0.2%/0.0%
11.03.25 STC00227  +IQC3524I UTRL        LT  06/01/10-14:11:27:9    0.2%/0.0%
11.03.25 STC00227  +IQC3524I T3270L2     LT  06/01/10-14:11:27:9    0.2%/0.0%
11.03.25 STC00227  +IQC3524I AD201L4     LT  06/01/10-14:11:27:9    0.2%/0.0%
11.03.25 STC00227  +IQC3524I T2741       LT  06/01/10-14:11:27:9    0.2%/0.0%
11.03.25 STC00227  +IQC3524I S2741P1     LT  06/01/10-14:11:27:9    0.2%/0.0%
11.03.25 STC00227  +IQC3524I AD202F      LT  06/01/10-14:11:27:9    0.2%/0.0%
```

WTO output for top long messages destinations

You can use the TOPLM parameter to obtain WTO output for top long messages destinations.

The following figure shows sample WTO output when the TOPLM parameter is used.

```
11.03.25 STC00228  +IQC3516I Top           6         Long Queue  Destinations
11.05.49 STC00228  +IQC3517I Dest        QT           LongBufs    LastActive
11.05.49 STC00228  +IQC3518I TSEUD01     LT           36/3.1%     01/10/30-06:58:02:6
11.05.49 STC00228  +IQC3518I DESRZA70,A7CICHBT
11.05.49 STC00228  +IQC3518I -DFSASYNCAP              32/ 2.7%    01/10/31-02:29:40:6
11.05.49 STC00228  +IQC3518I CLIENT1-
11.05.49 STC00228  +IQC3518I T3270LC OT              24/ 2.0%    01/11/02-10:19:04:1
11.05.49 STC00228  +IQC3518I T0910122 LT             24/ 2.0%    01/10/31-02:33:36:9
11.05.49 STC00228  +IQC3518I IMSNET.L62MVS1
11.05.49 STC00228  +IQC3518I -DFSASYNCAP              16/ 1.3%    01/10/31-02:33:36:9
11.05.49 STC00228  +IQC3518I VTAGB588 LT             16/ 1.3%    01/10/31-02:33:36:9
```

Chapter 14. Recovering shared queue messages

To recover messages after an abnormal termination in a shared-queues environment, you can use the RECOVER function or the RECOVERAS procedure.

When an abnormal termination occurs and a cold start is performed, use the RECOVER function to recover messages on the cold queue.

When the messages on the cold queue are lost, follow the RECOVERAS procedure to recover messages from the logs.

Topics:

RECOVER function description

Use the RECOVER function to select cold shared queue records for reprocessing, for deletion, and for possible reinsertion.

In a shared-queues environment, cold starts do not erase messages from the shared queues. Also, it is impossible in the shared queue environment for a single IMS to keep track of the status of a message that is inserted into the shared queues, because another IMS can remove and process that message.

For these just stated reasons, IMS does not log messages to the IMS log during checkpoint processing. DUMPQ and SNAPQ checkpoints are not done, and IMS Queue Control Facility cannot perform RECOVERAB or RECOVERDM processing.

However, IMS Queue Control Facility can recover messages on the cold shared queue that were in process when IMS abended and was then cold started. You can perform this recovery with the RECOVER function.

The RECOVER function keyword invokes the IMS Queue Control Facility recover logic that selects records from the cold shared queue for reprocessing, for deletion, for possible later insertion by the load function logic, or for purely informational purposes. IMS Queue Control Facility recover logic performs the following tasks:

1. Processes control blocks that were built by IMS Queue Control Facility from input control statements:
   - Determines record selection criteria from the date and time stamps, and the record type from the SELECT statement
   - Determines record selection criteria from message sources, and destinations from the INCLUDE and EXCLUDE statements
2. Writes to the RECOVER data set records that do meet the selection criteria of the SELECT, INCLUDE, and EXCLUDE statements
3. Requeues and unlocks, or deletes messages that meet the selection criteria of the SELECT, INCLUDE, and EXCLUDE statements
4. Discards records that fail to meet the selection criteria of the SELECT, INCLUDE, and EXCLUDE statements
5. Produces reports on the number and types of messages that the RECOVER function recovered from the cold shared queue
6. Returns control to IMS Queue Control Facility control processing

The job steps that are required for the RECOVER function are as follow: RECOVER

When to use the RECOVER function

Use the RECOVER function to recover messages that were left on the cold shared queue after an IMS cold start was necessary following an abnormal IMS termination (an abend).

If emergency restart fails, you must cold start IMS. You must also perform database backout or recovery to avoid data integrity problems.

The recovery function that is available in a shared-queues environment is RECOVER.

Specify the RECOVER keyword on a FUNCTION control statement to run the RECOVER function.

1. Shut down IMS using the **/CHE DUMPQ** command
2. Restart IMS
3. IMS ends abnormally
4. Restart IMS
5. An **/ERE** failure occurs
6. Back out updates or recover the database
7. Cold start IMS
8. Run the RECOVER function

Recovery processing in shared queues

Use the recovery logic to select and then process messages from the cold shared queue.

Recovery in a shared-queues environment is done by the IMS Queue Control Facility recovery logic. IMS Queue Control Facility recovery logic selects and then processes messages from the cold shared queue. You can use IMS Queue Control Facility recovery logic to perform the following tasks:

• DELETE messages from the cold shared queue
• UNLOCK or requeue messages for a later run
• Write messages to a RECOVER data set
• Report on the messages that RECOVER processed from the cold shared queue

The recover data set contains a copy of messages that can eventually be read into the shared queues by the load function logic. The data set could be used to subsequently perform stress, regression, or application testing.

Selection parameters

IMS Queue Control Facility recovery logic selects and removes messages from the cold shared queue based on your selection criteria. You specify the criteria with control statements that are read by IMS Queue Control Facility processing logic.

Selection criteria

You specify selection criteria to IMS Queue Control Facility recovery logic with FUNCTION, SELECT, INCLUDE, and EXCLUDE statements.

These statements define a subset of the message records to be selected from the cold shared queue. The FUNCTION statement is required. The other three statements are optional.

If no SELECT, INCLUDE, or EXCLUDE statements are present, all of the records from the cold shared queue are selected.

Message records can be selected based on the following criteria:

Specified in FUNCTION statement:

> Queue type of `COLD`

Specified in SELECT statement:

> Beginning date and time
> Ending date and time
> Log record type
> Start number
> System messages or no system messages
> Conversational messages or no conversational messages

Specified in INCLUDE or EXCLUDE statement:

> Destination with optional system ID
> Destination type
> Logical unit name
> Logical unit name destination
> Logical unit sidename
> Logical unit source
> Logical unit tpname
> Message type
> Source with optional system ID
> Source type
> TMember name
> TMember destination
> TMember source
> TPipe name
> Specific IMS ID
> Specific IMS ID destination
> Specific IMS ID source
> Either the originating unit-of-work, or the processing IMS ID, or both
> Either the originating unit-of-work, or the processing token, or both
> VTAM name

The FUNCTION statement specifies that the cold shared queue is to be processed. You can also use the FUNCTION statement to specify whether the recover data set is to be created using the READ or REPORTONLY options of the `ACTION=` keyword.

The FUNCTION statement is required to specify that the function is RECOVER.

The QUEUETYPE= keyword is optional as only the cold shared queue can be recovered.

The SELECT statement can also be used to specify whether the recover data set is to be created using the READ or REPORTONLY options of the `ACTION=` keyword. The SELECT statement specifies global criteria, the range of times, and the types of records to be selected by IMS Queue Control Facility recovery logic. Only one SELECT statement is allowed.

INCLUDE and EXCLUDE statements specify selection criteria about message sources and destinations. There can be any number of INCLUDE and EXCLUDE statements.

If any INCLUDE statements are present, a record must meet all criteria for at least one INCLUDE statement to be included in the recovery logic selections. Unspecified criteria are considered to be met.

A record that meets all of the criteria that is specified by an EXCLUDE statement is excluded by IMS Queue Control Facility recovery logic, even if the record meets all of the criteria that is specified by the SELECT statement and an INCLUDE statement. Unspecified criteria are considered to be met.

Selection processing

When given control by IMS Queue Control Facility control processing logic, recovery logic determines whether a recover data set is required; recovery logic does this by determining if `ACTION=READ` was specified, or defaulted to, on the FUNCTION or SELECT statement.

If the recover data set is required but not present, recovery logic issues a message and continues as if `ACTION=REPORTONLY` were specified.

IMS Queue Control Facility recover logic calls IMS requesting that a Common Queue Services (CQS) RECOVER function be performed on the cold shared queue.

Each record is processed by record processing logic that is specific for that record type.

The output logic calls the selection checker to determine if the record meets the selection criteria that was specified by the SELECT and INCLUDE statements.

- If the record does not meet these selection criteria, the record is bypassed and remains on the cold queue.
- If a record does meet these selection criteria, the output logic determines if the record meets the criteria for any EXCLUDE statement.
- If the record meets the criteria for an EXCLUDE statement, the record is discarded and remains on the cold queue.

When end-of-data is signaled by IMS, IMS Queue Control Facility recovery logic produces the following information:

- Reports on the number and type of messages that RECOVER selected on the cold shared queue
- Statistics on the number and types of control statements processed by IMS Queue Control Facility control processing logic

IMS Queue Control Facility recovery logic then returns control to IMS Queue Control Facility control processing logic for clean up and normal termination.

Control statements for RECOVER function

The RECOVER function lets you delete or requeue selected messages from the cold shared queue.

A copy of each processed message is written to the RECOVER data set, if the RECOVER data set is present, and if REPORTONLY is not specified.

The control statements that are recognized by the RECOVER function are:

```
* comment
EXCLUDE
FUNCTION
INCLUDE
SELECT
TITLE
```

RECOVER control statement abbreviations

You can abbreviate the specification of some of the RECOVER control statements

The following table shows a list of RECOVER keyword abbreviations that are allowed by the IMS Queue Control Facility control processing interpreter:

Table 17. IMS Queue Control Facility control processing RECOVER keyword abbreviations

Keyword	Abbreviation
EXCLUDE	EX

Table 17. IMS Queue Control Facility control processing RECOVER keyword abbreviations (continued)	
Keyword	**Abbreviation**
FUNCTION	FU
INCLUDE	IN
SELECT	SE

FUNCTION RECOVER statement syntax

The FUNCTION RECOVER control statement must follow certain syntactic specifications.

The syntax of the FUNCTION RECOVER control statement is:
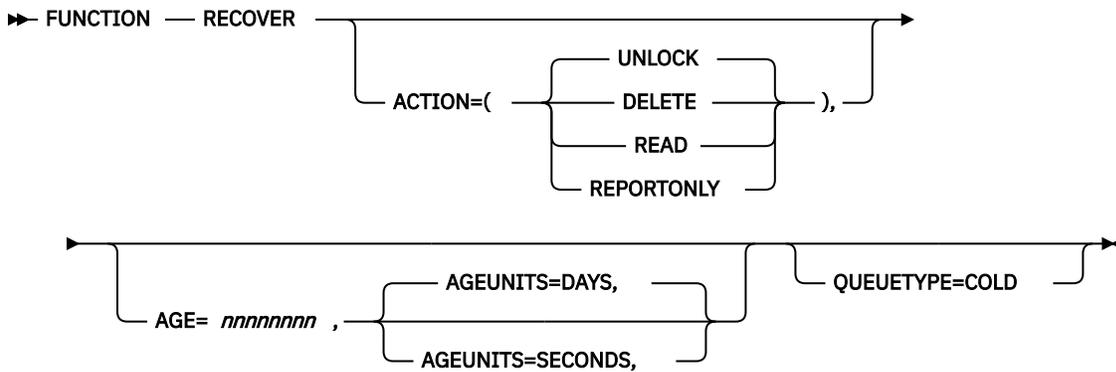
Note: The ACTION= keyword can also be specified as a global control on the SELECT control statement.

FUNCTION RECOVER INCLUDE statement syntax

The FUNCTION RECOVER INCLUDE statement must follow certain syntactic specifications.

In the JCL, the INCLUDE control statement is written on a separate line after the FUNCTION RECOVER control statement.

JCL example 1

```
FUNCTION RECOVER QUEUETYPE=COLD
INCLUDE DESTINATION=A*,
IMSIDS=IMS1
END
```
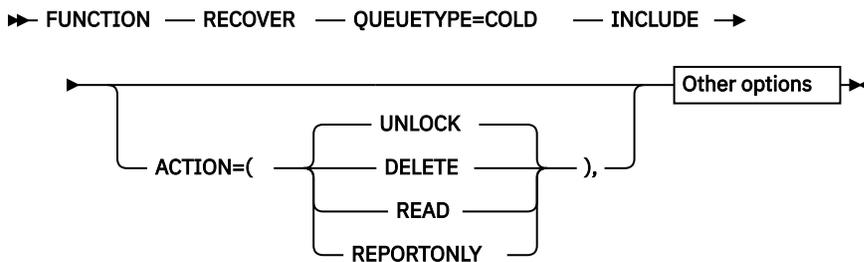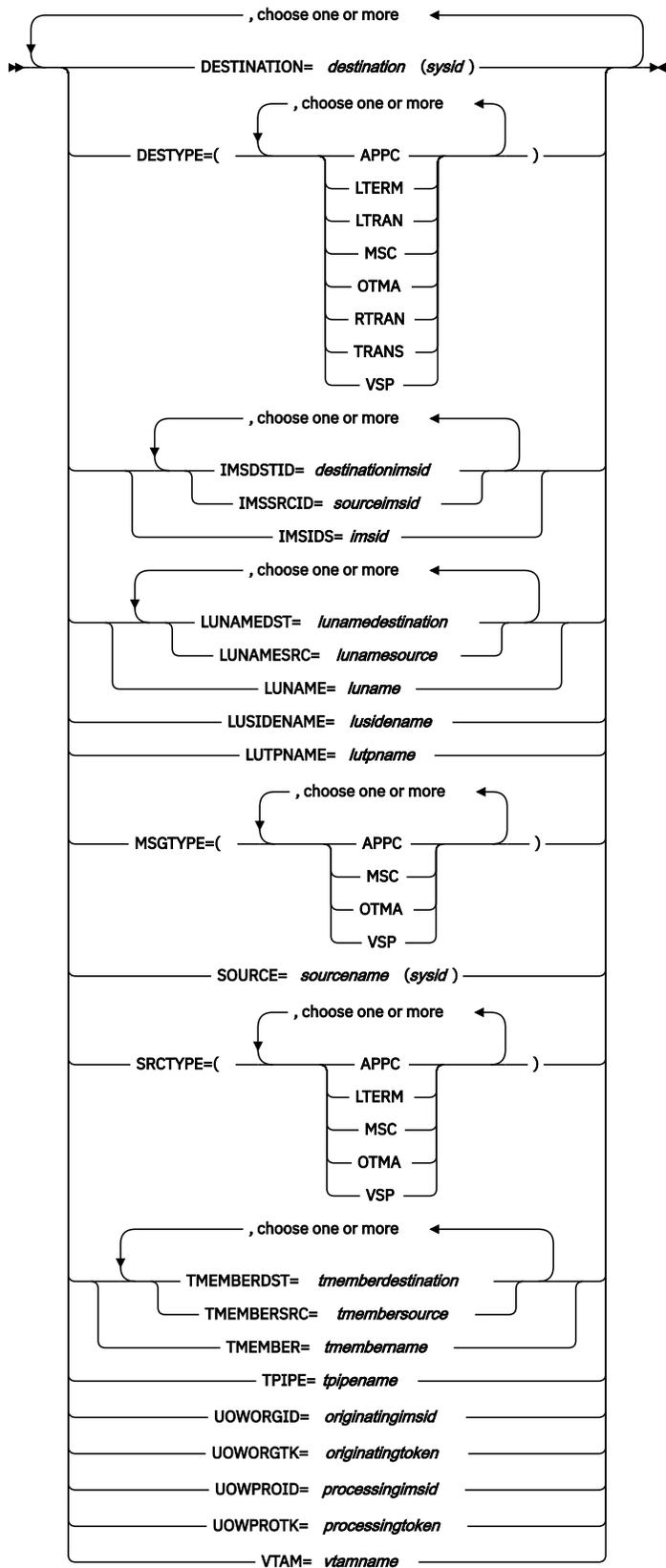
Syntax diagram

Other options

```
                                    ┌──────── , choose one or more ◄────────┐
►►─┬─────────────── DESTINATION= destination (sysid) ─────────────────────────►◄
   │                        ┌──── , choose one or more ◄────┐
   ├─ DESTYPE=( ─┬──┬─────── APPC ────────┬──┬── ) ──
   │             │  ├─── LTERM ───┤
   │             │  ├─── LTRAN ───┤
   │             │  ├─── MSC ─────┤
   │             │  ├─── OTMA ────┤
   │             │  ├─── RTRAN ───┤
   │             │  ├─── TRANS ───┤
   │             │  └─── VSP ─────┘
   │                  ┌──── , choose one or more ◄────┐
   ├─┬── IMSDSTID= destinationimsid ──┤
   │ ├── IMSSRCID= sourceimsid ───┤
   │ └─── IMSIDS= imsid ───┘
   │                  ┌──── , choose one or more ◄────┐
   ├─┬── LUNAMEDST= lunamedestination ──┤
   │ ├── LUNAMESRC= lunamesource ───┤
   │ └── LUNAME= luname ───┘
   ├── LUSIDENAME= lusidename ──
   ├── LUTPNAME= lutpname ──
   │             ┌──── , choose one or more ◄────┐
   ├─ MSGTYPE=( ─┬──┬─── APPC ───┬──┬── ) ──
   │             │  ├── MSC ───┤
   │             │  ├── OTMA ──┤
   │             │  └── VSP ───┘
   ├── SOURCE= sourcename (sysid) ──
   │             ┌──── , choose one or more ◄────┐
   ├─ SRCTYPE=( ─┬──┬─── APPC ───┬──┬── ) ──
   │             │  ├── LTERM ──┤
   │             │  ├── MSC ────┤
   │             │  ├── OTMA ───┤
   │             │  └── VSP ────┘
   │                  ┌──── , choose one or more ◄────┐
   ├─┬── TMEMBERDST= tmemberdestination ──┤
   │ ├── TMEMBERSRC= tmembersource ───┤
   │ └── TMEMBER= tmembername ───┘
   ├── TPIPE= tpipename ──
   ├── UOWORGID= originatingimsid ──
   ├── UOWORGTK= originatingtoken ──
   ├── UOWPROID= processingimsid ──
   ├── UOWPROTK= processingtoken ──
   └── VTAM= vtamname ──
```

FUNCTION RECOVER SELECT statement syntax

The FUNCTION RECOVER SELECT statement must follow certain syntactic specifications.

In the JCL, the SELECT control statement is written on a separate line after the FUNCTION RECOVER control statement.

JCL example 1

```
FUNCTION RECOVER QUEUETYPE=COLD
SELECT ACTION=(READ),
NOCONVMSG
END
```

Syntax diagram
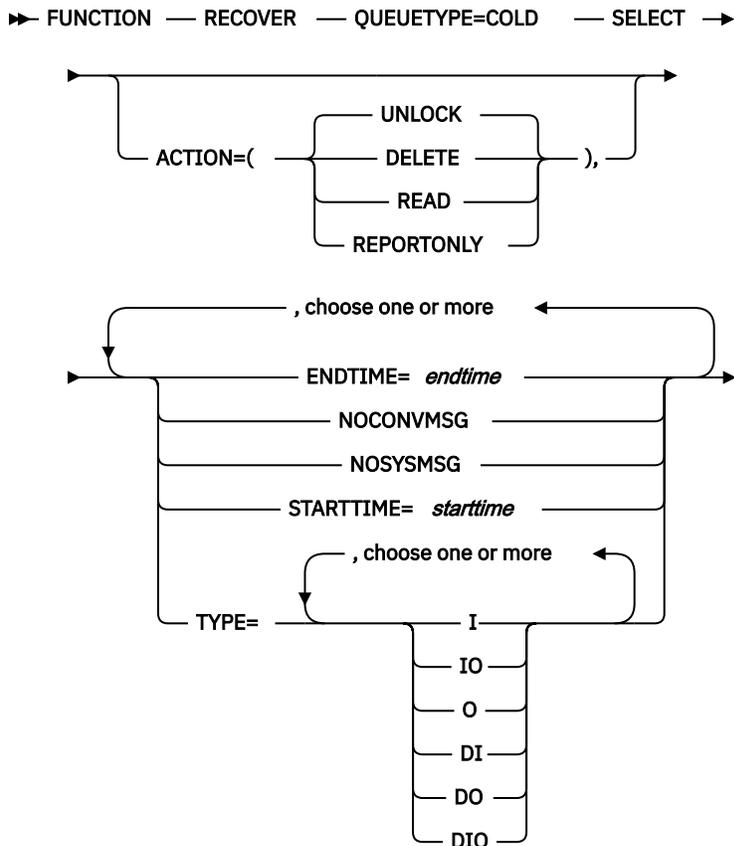
JCL for the RECOVER function

The RECOVER step is performed to delete or requeue messages from the cold queue that belongs to the shared queue.

Use the RECOVER procedure to requeue messages that have been left on the IMS cold shared queue because of an IMS abend.

You can run the RECOVER function in batch mode by submitting a JCL job stream.

The following figure shows the system operation flow for the RECOVER procedure:

```
┌─────────────────┐
│  IMS cold share │
│     queue       │
└─────────────────┘
         │
         ▼
┌──────────────────────────────────────────────────────┐
│                   #1 Recover                           │
│                                                        │
│  - Determine which messages are to be selected         │
│  - Delete from cold queue messages specified as DELETE │
│  - Requeue to cold queue messages specified as UNLOCK  │
│  - Write to RECOVER data set messages specified as READ│
│  - Print recover report                                │
└──────────────────────────────────────────────────────┘
         │
         ▼
┌─────────────────┐
│   IMS shared    │
│  message queue  │
└─────────────────┘
```
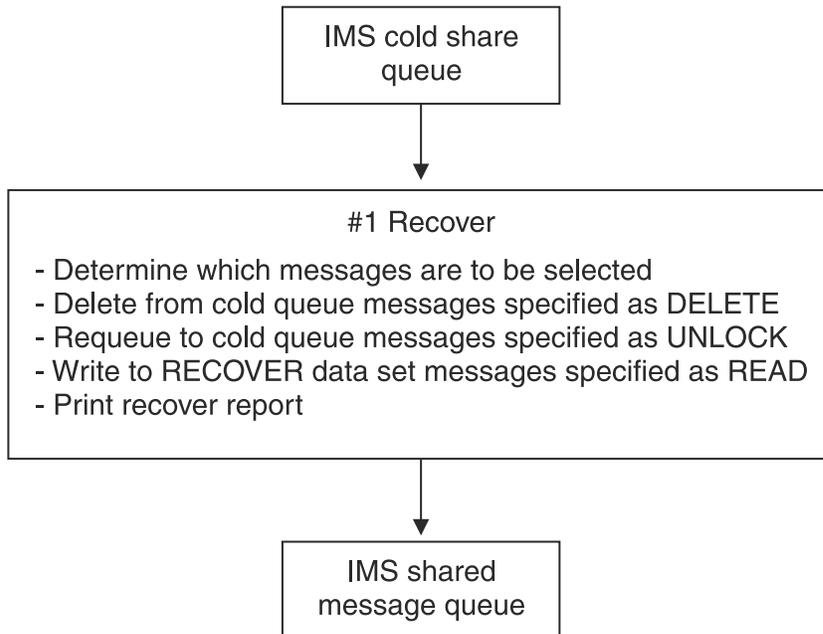
Figure 17. RECOVER procedure: system operation flow

Sample JCL to run the RECOVER function

The following example shows the JCL that is required to start a RECOVER step. Replace the items in italics with values that are appropriate for your environment:

```
//***********************************************************************
//* SAMPLE JCL TO EXECUTE THE RECOVER FUNCTION                          *
//***********************************************************************
//RECOVER  EXEC PGM=IQCINI0$,
//     PARM='BPEINIT=IQCBINI0,BPECFG=IQCIVPCF,IMSPLEX=IPLEX,QCFIMS=IMS1'
//STEPLIB  DD  DISP=SHR,DSN=hlqual.SIQCLINK
//         DD  DISP=SHR,DSN=IMSREL.RESLIB
//*
//QCFPRINT DD SYSOUT=A
//*
//RECOVER  DD DSN=QCF.RECOVER,UNIT=SYSDA,DISP=(NEW,CATLG),
//         DCB=(LRECL=32756,BLKSIZE=32760,RECFM=VB),
//         SPACE=(CYL,(5,5))
//*
//SNAPDUMP DD SYSOUT=A
//*
//QCFIN    DD DSN=USER.PRIVATE.PROCLIB(recover),UNIT=
```

DD statements for RECOVER

STEPLIB DD
Defines the library containing the IMS Queue Control Facility program load modules and IMS libraries that are required for the RECOVER function to run.

This DD statement is required.

QCFPRINT DD
Defines the output report data set.

This DD statement is required.

RECOVER DD
Defines an output data set. This can be used as input to the LOAD function.

If ACTION=(READ) is specified or defaulted to on the FUNCTION or SELECT statement, this statement is required.

If `ACTION=(REPORTONLY)` is specified on the FUNCTION or SELECT statement, this statement is optional.

QCFIN DD

Defines the input control statement data set.

This DD statement is required.

Example of the PROC USER.PRIVATE.PROCLIB(*recover*)

Following is an example of the PROC that was specified above in the QCFIN DD statement as, USER.PRIVATE.PROCLIB(*recover*):

```
***********************************************************
* recover all messages from the cold queue
***********************************************************
title 'Recover all messages from the Cold Queue'
function recover queuetype=(cold)
select action(read,unlock)
end
```

RECOVERAS procedure description

Use the RECOVERAS procedure to extract messages and other information from log records after an abnormal termination and a cold start in a shared-queues environment. You can use the RECOVERAS procedure to locate checkpoints and recover unprocessed messages starting from the checkpoint.

You can use the RECOVERAS procedure when messages are lost because an IMS cold start is necessary following an abend.

The RECOVERAS procedure extracts the following types of log records:

X'67D0' log record

Contains messages that IMS cannot put on the shared queues.

X'4040' log record

Contains UOWs for the messages that IMS cannot put on the shared queues.

X'3F' log record

Contains UOWs for the messages that IMS puts on the shared queues.

If an emergency restart fails, IMS must be cold started. Database batch backout or database recovery is required to avoid data integrity problems.

The following list typically precedes your use of RECOVERAS:

1. IMS is shut down or an IMS abend occurs

2. IMS is restarted

3. An IMS abend occurs

4. IMS is restarted

5. An /ERE failure occurs

6. The user backs out updates or recovers the database

7. The user issues a cold start of IMS

8. The user runs RECOVERAS

The following job steps are required for RECOVERAS:

1. SELECT

   During the SELECT job step, program IQCSELCT analyzes the execution parameters, prepares the LIST.LOG command for DBRC, reads the allocated logs, and extracts several sets of records to output files. In the SELECT step, you specify a restart checkpoint at a time before the shared queues entered the overflow condition.

The following control statement shows the format for the SELECT job step using a specified restart checkpoint:

```
CHKPT LAST,imsid,RS,D=yyyy.ddd,T=hh:mm:ss,DT=CP
FUNCTION RECOVERAS
```

imsid
　　The 4-byte IMS ID

yyyy.ddd
　　The date of the restart checkpoint according to the year (*yyyy*) and day (*ddd*).

hh:mm:ss
　　The time of the restart checkpoint in hours (*hh*), minutes (*mm*), and seconds (*ss*).

After the parameters are checked, the logs data sets are allocated, starting with the log that was created at the time of the specified checkpoint.

The following control statement shows the format for the SELECT job step using the automatic checkpoint:

```
CHKPT LAST,imsid,RS
FUNCTION RECOVERAS
```

imsid
　　The 4-byte IMSID

After the parameters are checked, the logs data sets are allocated, starting with the log at the start of the third before the last checkpoint.

The logs are read and several output files are created:

LOGOUT
　　Contains all the input and output messages that were created after the specified checkpoint.

SCRAPSEL
　　Contains the input and output messages that were created before the specified checkpoint (based on selection criteria and recoverability of the messages).

CKPTOFIL
　　Contains the formatted records created from the X'67D0', X'4040' and X'3F' log records.

2. SORT1

   In the SORT1 job step, CKPTOFIL is sorted by the UOWs of each message and by the number of the log record. For more information, see sample member IQCRCAS in the *hlq*.SIQCSAMP library file.

3. SORT2

   In the SORT2 job step, LOGOUT is sorted by the UOWs of each message. For more information, see sample member IQCRCAS in the *hlq*.SIQCSAMP library file.

4. CANCEL

   During the CANCEL job step, program IQCCANC1 reads the sorted files and selects only the messages with the missing X'3F' records. Selected messages are put in the CANCLOUT file to be loaded in the last LOAD step.

   The CANCEL job step ends with RC=0 if all UOWs with the missing X'3F' records are found in the sorted file with the logged messages. The CANCEL job step ends with RC=4 if some UOWs are not found in the sorted file with the logged messages. In the case that the CANCEL job step ends with RC=4, you must use an earlier restart checkpoint to recover the messages queues.

5. LOAD

   For information about the LOAD job step, see .

JCL for RECOVERAS procedure in shared queues

Refer to sample member IQCRCAS in the *hlq*.SIQCSAMP library file for an example of the RECOVERAS procedure. The sample member includes comments that explain how to customize the JCL and parameters for allocating the required data sets.

Chapter 15. Recovering nonshared queues messages

IMS Queue Control Facility uses different recovery procedures in the nonshared and shared queue environments.

Topics:

Recovery procedure overview

In a nonshared-queues environment, two procedures exist for recovering messages by using information in the PRISLD and/or PRIOLDS: RECOVERAB and RECOVERDM. The procedure you choose depends on how IMS terminated.

- Use RECOVERAB after a cold start following abnormal termination.
- Use RECOVERDM after a cold start following normal termination.

(RECOVERAB and RECOVERDM are keywords you specify in FUNCTION control statements. The keywords are read by the IMS Queue Control Facility select logic in the SELECT job step.)

RECOVERAB and RECOVERDM are both multistep requeuing procedures. Program IQCSELCT reads messages from the PRISLD and/or PRIOLDS during the SELECT step. IQCCANCL analyzes and cancels messages in the CANCEL step of RECOVERAB procedures.

Messages are sorted into their correct order for processing in the SORT*x* steps, and reloaded to the IMS online program which places them back on the message queue in the LOAD step.

IQCSELCT attaches DSPURX00 to list RECON information for the IMSID and analyzes the output as follows:

- Scans the PRISLD list and finds PRISLDs with the last valid checkpoint for function type and all following PRISLDs with log records for this function type.

  For RECOVERAB, the last valid checkpoint is the last DUMPQ, PURGE or SNAPQ. The log DSNs are PRISLDs containing the selected checkpoints and all subsequent PRISLDs.

  For RECOVERDM, the last valid checkpoint is DUMPQ or PURGE. Log DSNs are PRISLDs containing the selected checkpoint.

- Scans the PRIOLD list and orders PRIOLD data sets in start time sequence.
- Merges the list of PRISLDs and the list of PRIOLDs, replacing PRISLDs with PRIOLDs if they are for the same time interval.

  If several PRIOLDs are archived on one PRISLD, they will replace the PRISLD data set.

  The PRISLD data set is not replaced if there is a gap in stop/start times of PRIOLDs or if the stop time of a PRISLD is not found in the stop time sequence of the PRIOLDs.

- Adds PRIOLDs with later start times at the end of the list. If there are no PRISLDs, the list will consist of only PRIOLDs.
- Allocates and concatenates the data sets in the list, reads the concatenated data sets and finds the last valid checkpoint (depending on the requested function).

Checklist for rebuilding queues

In order to rebuild queues, perform the following steps:

1. Determine which procedure to use: RECOVERDM, RECOVERAB, or (to reprocess messages) REPROCESS.
2. If possible, wait until archiving jobs have archived PRIOLDs.
3. Start rebuilding procedures immediately after IMS is cold started.

   RECOVERAB and REPROCESS select messages from inside and after the selected checkpoint.

   If the procedures start after new messages are sent to IMS, these messages can be reinserted.
4. Use IQCSELECT CHKPT control statement with the LAST parameter to call RECON data sets and to perform dynamic allocation of PRISLDs and PRIOLDs.
5. Selection proceeds as follows:
   - Control card processing takes place first to determine the base checkpoint ID, program function, and selection criteria.
   - Allocated logs (PRISLDS and PRIOLDS) are searched for the base checkpoint.

     If you use the RECOVERDM procedure, the CHKPT must be a DUMPQ or PURGE.

     If you use the RECOVERAB procedure, the CHKPT can be either DUMPQ or PURGE or SNAPQ.
   - Depending on the procedure, the correct log record types are selected:
     - RECOVERDM selects only 4002 records; all records are reloaded to rebuild the queues.
     - RECOVERAB selects 4002, 01, 03, 30, 33, 34, 35, and 36 records.
     - REPROCESS selects 01 and 03 records.

RECOVERAB procedure description

The RECOVERAB procedure recovers messages in a nonshared-queues environment after an abnormal termination and a cold start.

Use RECOVERAB to locate checkpoints and recover unprocessed messages starting from the last DUMPQ, PURGE, or SNAPQ checkpoint. Use RECOVERAB when messages have been lost because an IMS cold start was necessary following an abnormal IMS termination (ABEND).

If emergency restart fails, IMS must be cold started. Database batch backout or database recovery is required to avoid data integrity problems.

The following list shows the sequence of events that are necessary to run RECOVERAB processing:

1. Shut down IMS using **/CHE DUMPQ**
2. Restart IMS
3. IMS ABEND
4. Restart IMS
5. **/ERE** failure
6. Back out updates or recover the database
7. Cold start IMS
8. Run RECOVERAB

RECOVERAB can also be used as a fallback aid in the nonshared-queues environment, to requeue messages across supported IMS releases.

Messages that were created on one supported release of IMS can be inserted into another supported release of IMS if the source and destination resources (LTERMs, transactions, MSC names, and other resources) are defined on both systems.

The following job steps are required for RECOVERAB:

- DFNDS
- SELECT
- DLTDS
- SORTA
- CANCEL
- SORTB
- LOAD

RECOVERDM procedure description

The RECOVERDM procedure recovers messages in a nonshared-queues environment after a normal termination and a cold start.

Use RECOVERDM to locate checkpoints, recover unprocessed messages from a DUMPQ or PURGE checkpoint that is logged at IMS termination with a **/CHE DUMPQ** or **/CHE PURGE** command.

Use RECOVERDM when messages have been lost because a cold start of IMS was necessary following a normal termination (such as, when an IMS system redefinition has been performed).

The following list shows the sequence of events that are necessary to run RECOVERDM processing:

1. Shut down IMS using **/CHE DUMPQ**
2. Optional system maintenance
3. Cold start IMS
4. Run RECOVERDM

RECOVERDM can also be used as a migration aid in the nonshared queue environment, to requeue messages across supported IMS releases; IMS 8.1 or later. Messages created on one supported release of IMS can be inserted into another supported release of IMS if the source and destination resources (such as LTERMs, transactions, MSC names, and other resources) are defined on both systems.

The following job steps are required for RECOVERDM:

- DFNDS
- SELECT
- DLTDS
- SORTB
- LOAD

Recovery logic components

In the nonshared-queues environment, IMS Queue Control Facility recovery consists of three logic components and requires the use of a sort package.

The three logic components are as follows:

Select logic
Select logic selects messages from the IMS SLDSs (based on your specifications) for reinsertion into the IMS message queue.

Cancel logic

Cancel logic is used in RECOVERAB processing only.

Cancel logic is run after the select logic and before the load function.

Cancel logic selects from the select logic output only those messages that were not processed by IMS.

Load function

The load function is always run as the last step of recovery processing.

The purpose of the load function is to reinsert the messages that were selected by the select logic or cancel logic into the IMS message queue.

Select logic

Use the SELECT control statement for the LOAD job step to select records for processing and reinsertion to the message queue.

IMS Queue Control Facility select logic selects records from an input SLDS for processing and reinsertion into the IMS message queue by the LOAD job step.

Select logic chooses the records by processing control statements that allow the select logic to make the following determinations:

- Identifies the base checkpoint from the CHKPT control statement.

  Identifies the logs, dynamically allocates the logs, and identifies the base checkpoint from the CHKPT control statement, if you specify the CHKPT control statement format with the LAST option.

- Determines program function from the FUNCTION control statement, if you specify the simple CHKPT control statement format without the LAST option.

  Determines the program function from the CHKPT control statement, if you specify the CHKPT control statement without the LAST option.

- Determines log record selection criteria for date and time stamps, log sequence number, and record type from the SELECT control statement

- Determines log record selection criteria for message sources and destinations from the INCLUDE and EXCLUDE control statements

- Locates the base checkpoint in the input SLDS

- Passes control to the correct program function logic to select the correct record types

- Processes the IMS system log records using type-specific record processing logic

- Writes records that fail to meet the selection criteria of SELECT and INCLUDE control statements to the SCRAPSEL data set

- Writes records that *do* meet the selection criteria of EXCLUDE control statements to the SCRAPSEL data set

- Writes successfully processed records to the LOGOUT data set to be further processed by subsequent job steps

- Terminates at end-of-file in the input SLDS

Cancel logic

Use the CANCEL logic for the LOAD function to read log records and scrap canceled messages to the SCRAPCAN data set.

IMS Queue Control Facility cancel logic reads log records from an input CANCLIN data set. The CANCLIN data set records have been sorted so that:

- Records with the same DRRN or Universal Time Stamp are contiguous.

- Queue manager processing auxiliary records precede message records with the same DRRN or Universal Time Stamp.

- Primary and secondary message records are in their normal order in relation to each other.

IMS Queue Control Facility cancel logic processes all records from the CANCLIN data set and determines if the message has been previously processed by IMS, or if the message is a duplicate or canceled message.

If the message has been processed by IMS, or is a duplicate or canceled message, cancel logic scraps the message by writing the message to the SCRAPCAN data set; otherwise cancel logic keeps the message by writing the message to the CANCLOUT data set.

Control statements for RECOVERAB and RECOVERDM functions

The RECOVERAB and RECOVERDM functions do not have their own like-named control statements as most of the other functions.

Both the processing logic for CANCEL and SELECT job steps (used by RECOVERAB and RECOVERDM) perform the following tasks:

- Interpret each statement
- Validate the data
- Build control blocks
- Issue error messages
- Run the appropriate logic

Control statements for CANCEL job step

Use the CANCEL job step to filter and save certain messages.

The CANCEL job step is part of the RECOVERAB procedure. The CANCEL job step filters out all the messages that have not been processed successfully and saves the valid messages in the CANCLOUT data set.

The CANCEL job step operates in the nonshared queue environment.

The following optional control statements are recognized by the CANCEL job step:

    * comment
    TITLE

Add a QCFIN DD statement to the CANCEL job step JCL if the comment or TITLE control statements are to be read by the CANCEL job step.

Control statements for SELECT job step

Use the SELECT job step to specify which messages that you want read into the nonshared message queue.

To specify the messages that you want read into the message queue in the nonshared-queues environment, provide control statements to the select logic in the SELECT job step.

These control statements specify the base checkpoint, the program function (which is also the name of the procedure to be followed), and the selection criteria to be used to select the messages.

Follow these guidelines for coding the control statements:

- The control statements can contain no operands or any number of operands in any order. First operand starts in column 10.
- Duplicate operands for the same statement are not allowed.
- One or more operands can be specified per control statement, separated by commas.
- Additional operands can be specified on continuation lines by following the previous operand with a comma and beginning the next operand in column 10 of the next line.

The following control statements are recognized by the SELECT job step:

    * comment
    FUNCTION
    INCLUDE
    EXCLUDE
    SELECT
    CHKPT
    NOWTOMSG
    TITLE
    WTOMSG

FUNCTION control statement syntax

The FUNCTION control statement must follow certain syntactic specifications.

```
>>-- FUNCTION --+-- RECOVERAB --+-- , --+----------------+--><
                '-- RECOVERDM --'       +-- AGE= nnn -----+
                                        '-- MAXAGE= nnn --'
```

INCLUDE control statement syntax

The INCLUDE control statement must follow certain syntactic specifications.

```
                                    , choose one or more
                              ┌──────────────────────────────◄──────┐
►►─ INCLUDE ──┬───────────────┴─ DESTINATION= destination (sysid) ──┴──►◄
              │                       , choose one or more
              │                   ┌────────────────────────◄─┐
              ├─ DESTYPE=( ───────┴─┬─ APPC ──┬──────────────┴─ ) ─┤
              │                     ├─ LTERM ─┤
              │                     ├─ LTRAN ─┤
              │                     ├─ MSC ───┤
              │                     ├─ OTMA ──┤
              │                     ├─ RTRAN ─┤
              │                     ├─ TRANS ─┤
              │                     └─ VSP ───┘
              ├─ LUNAME= luname ─────────────────┤
              ├─ LUNAMEDST= lunamedestination ───┤
              ├─ LUNAMESRC= lunamesource ────────┤
              ├─ LUSIDENAME= lusidename ─────────┤
              ├─ LUTPNAME= lutpname ─────────────┤
              │                   , choose one or more
              │               ┌────────────────────────◄─┐
              ├─ MSGTYPE=( ────┴─┬─ APPC ──┬─────────────┴─ ) ─┤
              │                  ├─ MSC ───┤
              │                  ├─ OTMA ──┤
              │                  └─ VSP ───┘
              ├─ SOURCE= sourcename (sysid) ─────┤
              │                   , choose one or more
              │               ┌────────────────────────◄─┐
              ├─ SRCTYPE=( ────┴─┬─ APPC ──┬─────────────┴─ ) ─┤
              │                  ├─ LTERM ─┤
              │                  ├─ MSC ───┤
              │                  ├─ OTMA ──┤
              │                  └─ VSP ───┘
              ├─ TMEMBER= tmembername ───────────┤
              ├─ TMEMBERDST= tmemberdestination ─┤
              ├─ TMEMBERSRC= tmembersource ──────┤
              ├─ TPIPE= tpipename ───────────────┤
              └─ VTAM= vtamname ─────────────────┘
```

SELECT control statement syntax

The SELECT control statement must follow certain syntactic specifications.

```
                      , choose one or more
                  ┌──────────────────────────┐
                  ▼                          │
►►─ SELECT ───┬──────── ENDNUM= endnum ───────┬──►◄
              │         ENDTIME= endtime       │
              │           NOCONVMSG            │
              │            NOSYSMSG            │
              │         STARTNUM= startnum     │
              │         STARTTIME= starttime   │
              │                                │
              │            , choose one or more│
              │        ┌───────────────────┐   │
              │        ▼                   │   │
              └─ TYPE= ──┬──── DI ──────────┴───┘
                         ├─ DIO ─
                         ├─ DO ──
                         ├─ I ───
                         ├─ IO ──
                         └─ O ───
```
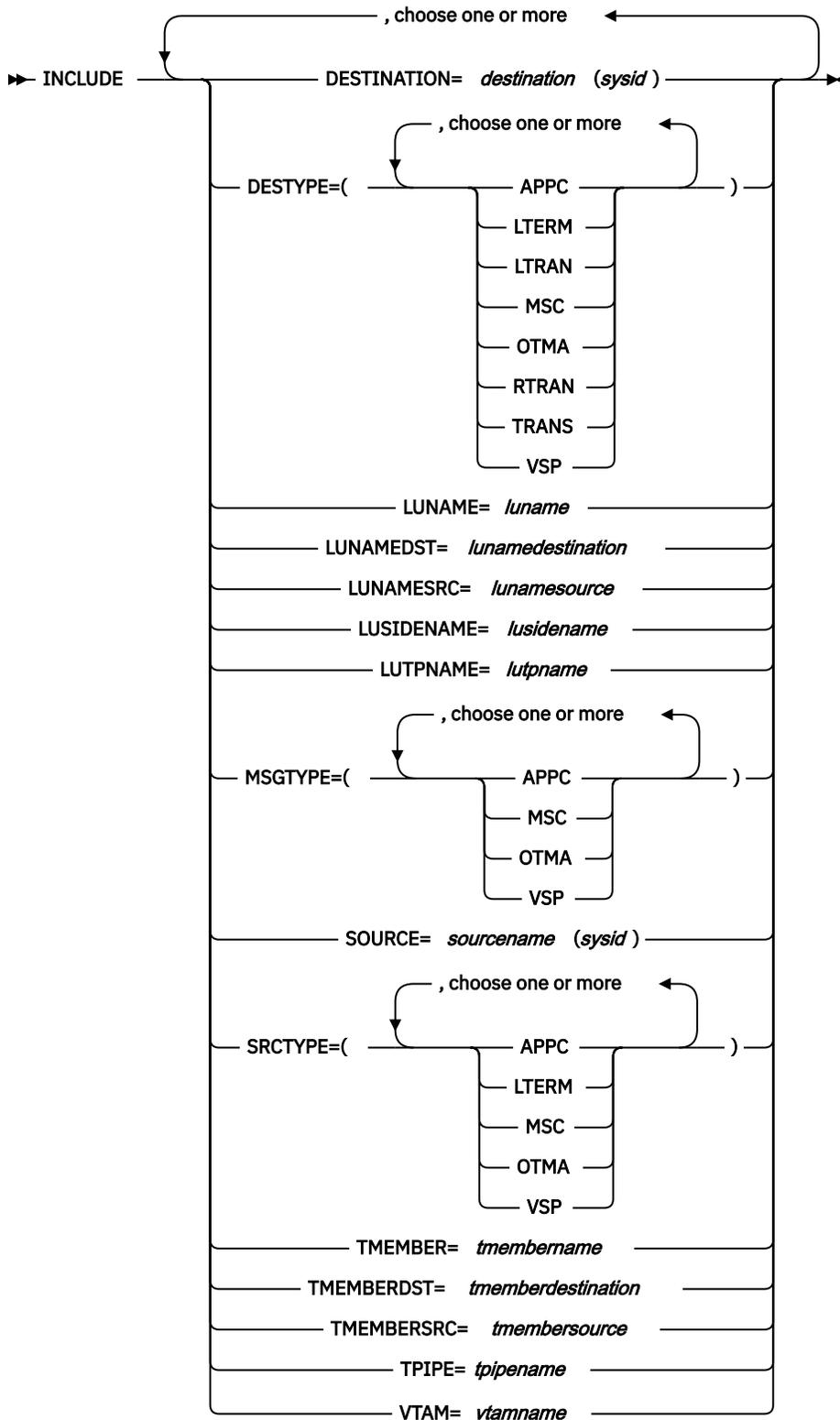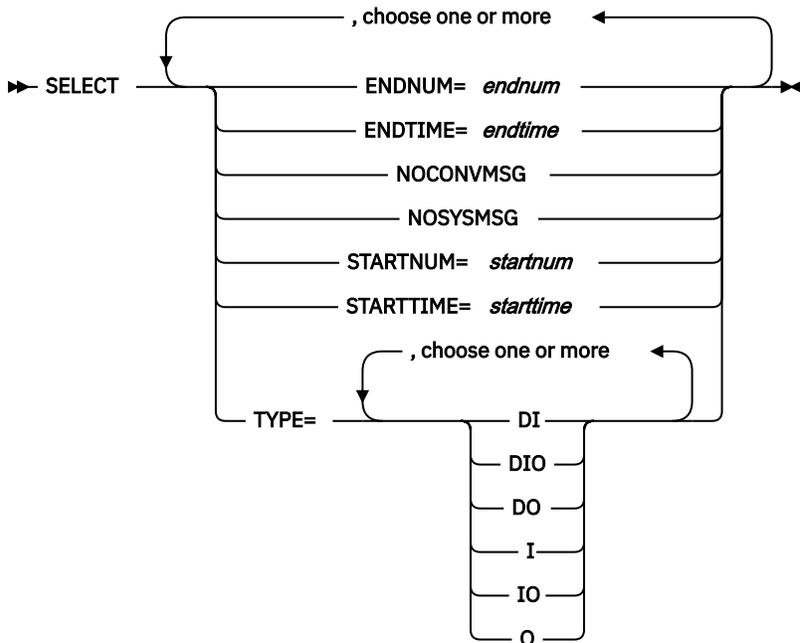
CHKPT control statement syntax

The CHKPT control statement must follow certain syntactic specifications.

The CHKPT control statement specifies the base checkpoint. The base checkpoint is the checkpoint that the SELECT step uses as a starting point for selecting messages from the input log data set.

The syntax of the CHKPT statement is:

```
►►─ CHKPT ── D= yyyy.ddd,  T= hh:mm:ss.thmiju,   Z= shh:mm ─►◄
```

or

```
►►─ CHKPT ── D= yyyy.ddd,  T= hh:mm:ss ─►◄
```

In the first example:

> *yyyy.ddd* is the date
> *hh:mm:ss.thmiju* is the time
> *shh:mm* is the zone

Together, these three comprise the Universal Time Stamp. This represents the IMS checkpoint that is to be used as the base checkpoint.

- *yyyy* = year
- *ddd* = day
- *hh*= hour
- *mm* = minute
- *ss* = second
- *t*= tenths of a second
- *h* = hundredths of a second
- *m* = milliseconds

- *i*= tenths of a millisecond
- *j* = hundredths of a millisecond
- *u* = microseconds
- *s* = sign + or - from Greenwich
- *hh* = hour offset from Greenwich
- *mm*= minutes offset from Greenwich

Because most installations are in an hourly time zone, the zone can be represented simply as the sign (if negative or in other words, if West of GMT (Greenwich mean time.)) and the offset.

For example, San Jose, CA is Z=-7 or Z=-8; Uithoorn in the Netherlands is Z=1 or Z=2; GMT is Z=0. If minutes are required, such as for Gander, Newfoundland, the leading zero can be omitted (for example, Z=-3:30).

All date and time stamps are in UTC (coordinated universal time) format. The zone is added to the UTC to produce local time.

In the second example, *yyyy.ddd* is the date, and *hh:mm:ss* is the local time as the local time appears in the IMS checkpoint timestamp message DFS994I.

Another option for the CHKPT control statement is to specify a value of 0 or 00000000000, starting at column 10. This allows SELECT to select the first valid checkpoint that SELECT encounters for that function type as the base checkpoint according to the following rules.

- For RECOVERAB, the first DUMPQ or SNAPQ checkpoint is selected.
- For RECOVERDM, the first DUMPQ checkpoint is selected.
- For REPROCESS, the first checkpoint is selected.

CHKPT control statement syntax, with zero values as input:

```
►►─ CHKPT ─────────── 0 ───────────►◄
                └─ 00000000000 ─┘
```

The alternate syntax of the CHKPT control statement lets you run SELECT with automatic checkpoint location, automatic log selection, and dynamic allocation.

To use this alternate syntax of CHKPT, add DD statements for automatic checkpoint location, automatic log selection, and dynamic allocation to the SELECT job step in the RECOVERAB, RECOVERDM, and REPROCESS procedures.

These DD statements include:

- RECON data sets
- IMSDALIB
- Sample log LOGIN

The alternate syntax of this version of the CHKPT control statement is:

```
►►─ CHKPT ── LAST, IMSID, FUNCTION,D=YYYY.DDD,T=HH:MM:SS      ►◄
```

Parameter reference for CHKPT control statement

LAST
    Indicates that the last valid checkpoint for the procedure type should be used.

IMSID
    The 4-byte IMSID.

FUNCTION
    One of the following procedure types:

AB
    RECOVERAB procedure

DM
    RECOVERDM procedure

RE
    REPROCESS procedure

D=*YYYY.DDD*
    For RECOVERDM and RECOVERAB, this parameter is optional and D=*YYYY.DDD* is the start date for the LIST.LOG RECON command. If a value for D= is omitted, D=*YYYY.DDD* defaults to the current date minus 7 days.

    For REPROCESS, this parameter is required and D=*YYYY.DDD* is the time of the selected checkpoint.

T=*HH:MM:SS*
    For RECOVERDM and RECOVERAB, this parameter is optional and T=HH:MM:SS is the start time for the LIST.LOG RECON command. If a value for T= is omitted, T=*HH:MM:SS* defaults to 00:00:00.

    For REPROCESS, this parameter is required and T=*HH:MM:SS* is the time of the selected checkpoint.

Job steps for RECOVERAB and RECOVERDM

You can run RECOVERAB and RECOVERDM in batch mode by submitting JCL job streams.

Some JCL steps are common to RECOVERAB and RECOVERDM and some are specific to one or the other.

If the subheading contains the procedure name, the JCL step is specific to that procedure.

If the subheading does not contain the procedure name, the JCL step is generic to both RECOVERAB and RECOVERDM procedures.

DFNDS job step

Use the DFNDS job step to define the VSAM data set that is used to store temporary message queues.

The DFNDS job step defines the VSAM data set that is used in the RECOVERDM and RECOVERAB procedures to store the message queue contents temporary.

The following example shows a sample JCL for the DFNDS job step:

```
//DFNDS    EXEC PGM=IDCAMS
//SYSPRINT DD    SYSOUT=A
//SYSIN    DD  *
//* DEFINE FOR VSAM DATA SET SHOULD BE MODIFIED TO MEET YOUR
//* SYSTEM REQUIREMENTS.  RECSZ PARAMETER, THE AVERAGE RECORD SIZE
//* CAN BE SET TO THE LRECL OF THE IMS SHORT MESSAGE
//* QUEUE AND THE MAX VALUE MUST BE EQUAL TO OR GREATER THAN
//* IMS LONG MESSAGE QUEUE.
//* NOTE:
//*     IT IS RECOMMENDED THAT THE MAXIMUM RECORDSIZE BE SET
//*     TO THE MAXIMUM ALLOWED BY VSAM; IE, 32761.
//*     IT IS RECOMMENDED THAT THE CONTROLINTERVALSIZE BE SET
//*     TO THE MAXIMUM ALLOWED BY VSAM; IE, 32768.
    DELETE QCF.V3M1
    SET MAXCC=0
    DEFINE CLUSTER (NAME(QCF.V3M1) -
              STORAGECLASS(PSTANDRD) -
              MANAGEMENTCLASS(PNOBACK) -
              IXD KEYS(4 0) -
              RECSZ(500 32761) -
              SPEED IMBED REPLICATE -
              REUSE SHAREOPTIONS(3 3) ) -
        DATA   (NAME(QCF.V3M1.DATA) -
              CONTROLINTERVALSIZE(32768) -
              CYL(100,20)) -
        INDEX  (NAME(QCF.V3M1.INDEX) -
              CONTROLINTERVALSIZE(512) -
              TRACKS(1,1))
//*
```

DD statements for DFNDS job step

SYSPRINT DD
> Defines the output message data set. Required.

SYSIN DD
> Defines the input control statement data set. Required.

The maximum value for the RECSZ parameter for the DEFINE of the VSAM data set must be equal to or greater than the IMS long message queue, plus twenty bytes (four bytes for variable-blocked data sets, eight bytes for the log sequence number, and eight bytes for the time stamp)

However, to accommodate possible future changes, it is recommended that the maximum value for RECSZ be set to 32761, the maximum allowed by VSAM, and the CONTROLINTERVALSIZE be set to 32768, the maximum allowed for that parameter.

DLTDS job step

Use the DLTDS job step to delete the VSAM data set that is used to store temporary message queues.

The DLTDS job step deletes the VSAM data set that is used in the RECOVERDM and RECOVERAB procedures to store the message queue contents temporary.

The following example shows the sample JCL for the DLTDS job step:

```
//DLTDS    EXEC PGM=IDCAMS
//SYSPRINT DD    SYSOUT=A
//SYSIN    DD  *
    DELETE (QCF.V3M1)
//*
```

DD statements for DLTDS job step

SYSPRINT DD
> Defines the output message data set. Required.

SYSIN DD
> Defines the input control statement data set. Required.

SELECT job step using specified checkpoint

Use the SELECT job step to run the select logic. Based on the parameters that you specify, the SELECT job step selects messages from the SLDSs for reinsertion into the IMS message queue.

Example JCL for SELECT job step

The following example shows SELECT job step JCL and control statements:

```
//********************************************************************
//*                                                                *
//*                 SELECT JOB STEP                                *
//*                                                                *
//********************************************************************
//SELECT   EXEC PGM=IQCSELCT,REGION=0M
//STEPLIB  DD   DISP=SHR,DSN=hlqual.SIQCLINK
//QCFIN    DD  *
CHKPT    00000000000
FUNCTION RECOVERAB
*
*-* OR *-*
*
CHKPT D=YYYY.DDD,T=HH:MM:SS
FUNCTION RECOVERAB
*
//*
//* THIS IS A SAMPLE OF MULTIPLE LOG TAPES WHERE EACH LOG HAS A
//* DIFFERENT DATA SET NAME, THUS REQUIRING SEPARATE DD STATEMENTS.
//*
```

```
//LOGIN    DD  DISP=OLD,UNIT=TAPE,LABEL=(1,SL),DSN=LOG1,
//            VOL=SER=IMSLG1
//        DD  DISP=OLD,UNIT=TAPE,LABEL=(1,SL),DSN=LOG2,
//            VOL=SER=IMSLG2
//*
//MSGQ     DD  DSN=QCF.V3M1,
//            AMP=('BUFND=20,BUFNI=10'),
//            DISP=(OLD,KEEP,KEEP)
//SCRAPSEL DD  DSN=QCF.SCRAPSEL,UNIT=SYSDA,SPACE=(CYL,(5,5)),
//            DCB=(LRECL=32756,BLKSIZE=32760,RECFM=VB),
//            DISP=(NEW,CATLG)
//*
//*   THE LRECL AND BLKSIZE FOR THE SNAPQ DATA SET SHOULD
//*   EQUAL THE LRECL AND BLKSIZE FOR THE LOGIN DATA SET.
//*   HOWEVER, IT IS RECOMMENDED THAT THE MAXIMUM SIZES, LRECL=32756
//*   AND BLKSIZE=32760, BE USED.
//*
//SNAPQ    DD  UNIT=SYSDA,DISP=NEW,SPACE=(CYL,(5,5)),
//            DSN=&&SNAPQ,
//            DCB=(LRECL=32756,BLKSIZE=32760,RECFM=VB)
//*
//*   THE LRECL OF THE LOGOUT DATA SET MUST BE AT LEAST 256
//*   BYTES LONGER THAN THE LRECL OF THE LOGIN DATA SETS
//*   TO ACCOMMODATE THE DFSMRQPF PREFIX.
//*   HOWEVER, IT IS RECOMMENDED THAT THE MAXIMUM SIZES, LRECL=32756
//*   AND BLKSIZE=32760, BE USED.
//*
//LOGOUT   DD  DSN=&&LOGOUT,UNIT=SYSDA,SPACE=(CYL,(5,5)),
//            DCB=(LRECL=32756,BLKSIZE=32760,RECFM=VB),
//            DISP=(NEW,PASS)
//QCFPRINT DD  SYSOUT=A
//*
//SNAPDUMP DD  SYSOUT=A
//*
//SORTINA  DD  DSN=&&SORTINA,UNIT=SYSDA,SPACE=(TRK,(1,1)),
//          DCB=(LRECL=80,RECFM=FB,BLKSIZE=800),DISP=(NEW,PASS)
//SORTINB  DD  DSN=&&SORTINB,UNIT=SYSDA,SPACE=(TRK,(1,1)),
//          DCB=(LRECL=80,RECFM=FB,BLKSIZE=800),DISP=(NEW,PASS)
```

DD statements for SELECT job step

STEPLIB DD
Defines the library containing the QCF program load modules. Required.

QCFIN DD
Defines the input control statement data set. Required.

LOGIN DD
Defines the IMS system log data set to be used for input. Required.

MSGQ DD
Defines a temporary VSAM data set (created by the DFNDS step) where SELECT recreates the message queue in the RECOVERDM and RECOVERAB procedures. Required for RECOVERDM and RECOVERAB procedures.

SCRAPSEL DD
Defines a data set used to collect records not selected for reinsertion into the IMS message queue, either because of selection criteria or because of problems with the records. Optional.

SNAPQ DD
Defines a temporary data set for the RECOVERAB procedure when using a SNAPQ checkpoint. SELECT saves log records processed during SNAPQ for processing after the type 4002 records have been read. Required for the RECOVERAB procedure when using a SNAPQ checkpoint.

LOGOUT DD
Defines the data set where SELECT collects the records that SELECT has selected for reinsertion into the IMS message queue. Required.

QCFPRINT DD
Defines the output report data set. Required.

SORTINA DD
Defines an output data set that can be used as input to the SORTA step. This data set will contain the correct sort control cards based on the IMS release level.

These sort control cards are for the IBM sort program product. If another sort package is used, you might need to generate the necessary control statements.

The SELECT step in the SELECT step JCL above shows the scratch data set allocated to UNIT=SYSDA; a more practical allocation would be to VIO. However, the naming conventions for VIO are installation dependent; in other words, UNIT=VIO might or might not work at an installation (some installations use UNIT=SYSVIO, UNIT=TEMPDISK, and so on).

SORTINB DD
Defines an output data set that will be used as input to the SORTB step. This data set will contain the correct sort control cards based on the IMS release level.

These sort control cards are for the IBM sort program product. If you use another sort package, you might need to generate the necessary control statements.

Note: The example SELECT step shows the scratch data set that is allocated to UNIT=SYSDA; a more practical allocation would be to VIO. However, the naming conventions for VIO are environment dependent, so UNIT=VIO might not work in your environment (some environments use UNIT=SYSVIO, UNIT=TEMPDISK, or other naming conventions).

SELECT job step using automatic checkpoint

You can run the select processing logic with CHKPT=LAST to specify automatic checkpoint location, automatic log selection, and dynamic allocation.

The command format is shown as follows:

```
CHKPT LAST,IMSID,FUNCTION,D=yyyy.ddd,T=hh:mm:ss
```

where the following values are true:

LAST
Specifies that QCF should use the last valid checkpoint for the function type.

IMSID
Specifies the 4-byte IMS id.

FUNCTION
Specifies either AB for RECOVERAB or DM for RECOVERDM.

D=*yyyy.ddd*
Specifies the start date for **LIST.LOG.RECON** command. D=*yyyy.ddd* is optional. If omitted, the date defaults to the current date minus 7 days.

T=*hh:mm:ss*
Specifies the start time for **LIST.LOG RECON** command T=*hh:mm:ss* is optional. If omitted, the time defaults to 00:00:00.

Using this format, SELECT calls DSPURX00 and analyzes the output to find the last valid checkpoint for the function type and all PRIOLDs with log records for the function type.

For RECOVERAB, the last valid checkpoint is the last DUMPQ or PURGE or DSNAPQ. The log DSNs are all PRIOLDs containing the selected checkpoints and the next PRIOLDs.

For RECOVERDM, the last valid checkpoint is DUMPQ or PURGE. The log DSNs are all PRISLDs containing the selected checkpoints and the next PRISLDs.

SELECT dynamically allocates all necessary PRISLD data sets and concatenates them when building the // LOGIN DD.

You can run the SELECT job step in batch mode by submitting JCL job streams.

Example JCL for SELECT job step

Sample JCL for processing select logic is shown in the following sample. Add the DD statements to the IQCSELCT step in the RECOVERAB and RECOVERDM functions that are necessary for automatic checkpoint location, log selection, and allocation.

```
//****************************************************************************/
//*                                                                         */
//*      SELECT STEP                                                        */
//*                                                                         */
//****************************************************************************/
//SELECT EXEC PGM=IQCSELCT,REGION=0M
//STEPLIB DD DISP=SHR,DSN=hlqual.SIQCLINK
//        DD DISP=SHR,DSN=IMS.RESLIB
//IMSDABLIB DD DISP=SHR,DSN=&ismdalib
//SYSIN    DD DSN=&sysin,UNIT=SYSDA,
//         DISP=SHR,
//         SPACE=(TRK,(1,1))
//RECON1   DD DISP=SHR,DSN=&recon1
//RECON2   DD DISP=SHR,DSN=&recon2
//QCFIN  DD *
CHKPT   LAST,IMA8,AB,D=2003.103
FUNCTION RECOVERAB
//*
//MSGQ   DD DSN=hlqual,
//      AMP=('BUFND=20,BUFNI=10'),
//      DISP=(OLD,KEEP,KEEP)
//SCRAPSEL DD DUMMY
//SNAPDUMP DD DUMMY
//*
//*THE LRECL AND BLKSIZE FOR SNAPQ DATA SET SHOULD EQUAL LOGINS.
//*
//SNAPQ  DD UNIT=systda,DISP=NEW,SPACE=(CYL,(5,5)),
//       DCB=(LRECL=4000,BLKSIZE=4004)
//*
//*THE LRECL OF THE LOGOUT DATA SET MUST BE AT LEAST 256
//*BYTES LONGER THEN THE LRECL OF THE LOGIN DATA SETS,
//*TO ACCOMMODATE MRPREFIX.
//*
//*
//LOGOUT  DD DSN=&&LOGOUT,UNIT=systda,SPACE=(CYL,(5,5)),
//      DCB=(LRECL=4066,BLKSIZE=4070,RECFM=VB),
//      DISP=(NEW,PASS)
//SORTINA DD DSN=&&SORTINA,UNIT=systda,SPACE=(TRK,(1,1)),
//      DCB=(LRECL=80,BLKSIZE=800,RECFM=FB),
//       DISP(NEW,PASS)
//SORTINB DD DSN=&&SORTINB,UNIT=systda,SPACE=(trk,(1,1)),
//      DCB=(LRECL=80,BLKSIZE=800,RECFM=FB),
//       DISP(NEW,PASS)
//QCFPRINT DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//*
```

DD statements for SELECT job step

CHKPT

The format of the CHKPT control statement does not require //LOGIN DD; //LOGIN DD is ignored if present. //STEPLIB DD should point to QCF and IMS load module libraries.

IMSDALIB DD

Specifies dynamic allocation of RECON data sets.

SYSIN DD

The standard input data set for DBRC commands; if not provided, SYSIN is dynamically allocated.

SYSPRINT DD

The standard output data set for DBRC commands; if not provided, SYSPRINT is dynamically allocated.

RECON1/RECON2 DD

If not provided, RECON1 and RECON2 are dynamically allocated.

SYSOUT DD

Contains information from allocation/look up for the checkpoint process:

- List of checkpoints on the first PRISLD
- List of dynamically allocated LOGs

Example output from specifying //SYSOUT DD in the SELECT job step

```
____   ************************************************************************
____   *   START LOGIN DSN:
____   *     VND0442.PM25358.NFTPFILE.SLDSCKPT.T084312
____   *   CHECKPOINTS ON START LOGIN DSN:
____   *    CHKPT  D=2002.018,T=07:40:20.172581,z=+01:00
____   *    CHKPT  D=2002.018,T=08:50:07.962918,z=+01:00
____   *   END OF CHECKPOINTS ON START LOGIN DSN
____   ************************************************************************
____   ************************************************************************
____   *   LOGIN DSNS IN USE-DYNAMIC ALLOCATION
____   *     VND0442.PM25385.NFTPFILE.SLDSCKPT.T084312
____   *     VND0442.PM05309.NFTPFILE
____   *  END OF DYNAMIC ALLOCATION
____   ************************************************************************
```

SORTA job step

The SORTA job step sets up the sort order for the CANCEL job step.

The SORTA job step is required between the SELECT job step and the CANCEL job step in the RECOVERAB procedure.

The SORTIN input to SORTA is the LOGOUT data set from the SELECT step.

The SYSIN input to the SORTA step can be coded by you, or if you included the SORTINA data set in the SELECT step, you can use that data set.

The output from SORTA is passed to the CANCEL job step to be used as the CANCLIN input data set.

The SORTA sort fields are:

1. Unit-of-work, column 53, length 32, binary, ascending order
2. The QCF-assigned sequence number, column 97, length 4, binary, descending order
3. The QCF-assigned secondary record number, column 87, binary, ascending order

Example: SORTA job step sort control statements

The following example shows SORTA control statements:

```
SORT FIELDS=(53,32,BI,A,97,4,BI,D,87,2,BI,A),FILSZ=E2000
RECORD TYPE=V
END
```

This sort order ensures that the message records will be processed by CANCEL logic in the proper order.

Example: SORTA job step JCL

The following example SORTA job step JCL shows a SORTA step. This step uses the output of SELECT as the SYSIN to SORT.

```
//*****************************************************************
//*                                                              *
//*                SORTA JOB STEP                                *
//*                                                              *
//*****************************************************************
//SORTA    EXEC PGM=SORT,REGION=0K,COND=(0,NE,SELECT),
//         PARM='CORE=MAX'
//SORTLIB  DD   DSN=SYS1.SORTLIB,DISP=SHR
//SORTIN   DD   DSN=*.SELECT.LOGOUT,DISP=(OLD,DELETE)
//SORTOUT  DD   DSN=&&SORTOUT,UNIT=SYSDA,SPACE=(CYL,(5,5)),
//         DCB=(LRECL=32756,BLKSIZE=32760,RECFM=VB),
//         DISP=(NEW,PASS)
```

```
//SYSOUT   DD   SYSOUT=A
//SORTWK01 DD   UNIT=SYSDA,SPACE=(CYL,(20,10))
//SORTWK02 DD   UNIT=SYSDA,SPACE=(CYL,(20,10))
//SORTWK03 DD   UNIT=SYSDA,SPACE=(CYL,(20,10))
//SORTWK04 DD   UNIT=SYSDA,SPACE=(CYL,(20,10))
//SYSIN    DD   DSN=&&SORTINA,DISP=(OLD,PASS)
```

CANCEL job step

Use the CANCEL job step to select (from the SELECT job step) just those messages that have not been successfully processed by IMS.

The CANCEL job step is run between the SELECT and LOAD job steps in the RECOVERAB procedure.

The purpose of the CANCEL step is to select specific output from the SELECT job step. That output includes only those messages that have not been successfully processed by IMS.

Example: CANCEL job step JCL

The following example shows sample JCL to run the CANCEL job step.

```
//**********************************************************************
//*                                                                    *
//*                CANCEL JOB STEP                                      *
//*                                                                    *
//**********************************************************************
//CANCEL  EXEC PGM=IQCCANCL,COND=((0,NE,SELECT),(0,NE,SORTA)),
//             REGION=0M
//STEPLIB  DD   DISP=SHR,DSN=hlqual.SIQCLINK
//CANCLIN  DD   DISP=(OLD,DELETE),DSN=*.SORTA.SORTOUT
//CANCLOUT DD   DSN=QCF.CANCLOUT,UNIT=SYSDA,DISP=(NEW,PASS),
//         DCB=(LRECL=32756,BLKSIZE=32760,RECFM=VB)
//SCRAPCAN DD   DSN=QCF.SCRAPCAN,UNIT=SYSDA,DISP=(NEW,CATLG),
//         DCB=(LRECL=32756,BLKSIZE=32760,RECFM=VB)
//QCFPRINT DD   SYSOUT=A
//SNAPDUMP DD SYSOUT=A
//*
```

DD statements for CANCEL job step

STEPLIB DD
Defines the library containing the IMS Queue Control Facility program load modules. Required.

CANCLIN DD
Defines the input data set, which was the output from the SELECT job step and sorted by the SORTA job step. Required.

CANCLOUT DD
Defines the output data set, which will contain the message records to be passed (through the SORTB job step) to the LOAD step for reinsertion into the IMS message queue. Required.

SCRAPCAN DD
Defines an output data set for collecting message records rejected by the CANCEL job step. Optional.

QCFPRINT DD
Defines the output report data set. Required.

SORTB job step

The SORTB job step is required in several MS Queue Control Facility procedures to sort the input to the SELECT job step for RECOVERAB, RECOVERDM, and REPROCESS.

The SORTB job step is required in several MS Queue Control Facility procedures to sort the input to the SELECT step.

The SORTIN input to SORTB is the LOGOUT data set from the SELECT step in the RECOVERDM and REPROCESS procedures.

The SORTIN input to SORTB is the CANCLOUT data set from the CANCEL step in the RECOVERAB procedure.

You can code the SYSIN input to the SORTB job step, or if you included the SORTINB data set in the SELECT job step, you can use that data set in the RECOVERAB procedure. The output from SORTB is passed to the SELECT job step to be used as the ISRTFIL input data set.

The SORTB sort fields are:

1. Universal Time Stamp, column 41, length 12, binary, ascending order
2. The QCF-assigned sequence number, column 97, length 4, binary, ascending order
3. Unit of work, column 53, length 32, binary, ascending order
4. The QCF-assigned secondary record number, column 87, length 2, binary, ascending order

The following sample shows sample SORTB control statements:

SORTB job step sort control statements

This sort order ensures that the message records are inserted into the IMS message queues in the proper chronological order.

```
SORT FIELDS=(41,12,BI,A,97,4,BI,A,53,32,BI,A,87,2,BI,A),FILSZ=E2000
RECORD TYPE=V
END
```

SORTB job step JCL

The following example shows a sample SORTB job step (this step uses the output of SELECT as the SYSIN to SORT).

```
//********************************************************************
//*                                                                  *
//*               SORTB JOB STEP                                     *
//*                                                                  *
//********************************************************************
//SORTB    EXEC PGM=SORT,REGION=0K,
//         COND=((0,NE,SELECT),(0,NE,SORTA),(0,NE,CANCEL)),
//         PARM='CORE=MAX'
//SORTLIB  DD   DSN=SYS1.SORTLIB,DISP=SHR
//SORTIN   DD   DSN=*.CANCEL.CANCLOUT,DISP=(OLD,DELETE)
//SORTOUT  DD   DSN=&&SORTOUT,UNIT=SYSDA,SPACE=(CYL,(5,5)),
//         DCB=(LRECL=32756,BLKSIZE=32760,RECFM=VB),
//         DISP=(NEW,PASS)
//SYSOUT   DD   SYSOUT=A
//SORTWK01 DD   UNIT=SYSDA,SPACE=(CYL,(20,10))
//SORTWK02 DD   UNIT=SYSDA,SPACE=(CYL,(20,10))
//SORTWK03 DD   UNIT=SYSDA,SPACE=(CYL,(20,10))
//SORTWK04 DD   UNIT=SYSDA,SPACE=(CYL,(20,10))
//SYSIN    DD   DSN=&&SORTINB,DISP=(OLD,PASS)
```

LOAD job step

The purpose of the LOAD job step is to reinsert messages to the message queues. The LOAD step runs as an IMS Queue Control Facility batch program.

LOAD job step JCL

The following example shows the JCL that is required to start a LOAD step. Replace the items in italics with values that are appropriate for your environment.

```
//********************************************************************
//*                                                                  *
//*               LOAD JOB STEP                                      *
//*                                                                  *
//********************************************************************
```

```
//LOAD      EXEC PGM=IQCINI0$,
//      PARM='BPEINIT=IQCBINI0,BPECFG=IQCIVPCF,IMSPLEX=IPLEX,QCFIMS=IMS1'
//STEPLIB  DD  DISP=SHR,DSN=hlqual.SIQCLINK
//         DD  DISP=SHR,DSN=IMSREL.RESLIB
//*
//QCFPRINT DD SYSOUT=A
//*
//LOAD      DD DISP=SHR,DSN=QCF.UNLOAD
//*
//* SCRAPLOG IS AN OPTIONAL DD STATEMENT. IT WILL CONTAIN THE
//* REJECTED MESSAGES FROM THE LOAD STEP.
//* DCB PARAMETERS SHOULD BE THE SAME AS THOSE FOR THE LOAD
//* DATA SET.
//* HOWEVER, IT IS RECOMMENDED THAT THE MAXIMUM SIZES FOR LRECL
//* AND BLKSIZE ARE USED.
//*
//SCRAPLOG DD DSN=QCF.SCRAPLOG,UNIT=SYSDA,DISP=(NEW,CATLG),
//         DCB=(LRECL=32756,BLKSIZE=32760,RECFM=VB),
//         SPACE=(CYL,(5,5))
//*
//SNAPDUMP DD SYSOUT=A
//*
//QCFIN DD DSN=USER.PRIVATE.PROCLIB(load),UNIT=SYSDA
```

DD statements for LOAD job step

STEPLIB DD
> Defines the library containing the IMS Queue Control Facility program load modules and IMS libraries that are required for the load step to run. Required.

QCFPRINT DD
> Defines the output report data set. Required.

LOAD DD
> Defines the input data set. This can be the output from BROWSE or the UNLOAD function. Required.

SCRAPLOG DD
> Defines an output data set to receive message records that are rejected in the LOAD step. Optional.

SNAPDUMP DD
> Defines an output data set for collecting debugging information. Optional. Required if the DEBUG control statement is present.

QCFIN DD
> Defines the input control statement data set. Required.

Running JCL for RECOVERAB

Use RECOVERAB to recover the messages that were processed before the abend that caused a cold start.

If emergency restart fails following an IMS abend, an IMS cold start is necessary. Messages lost from the queues because of this circumstance can be recovered by the RECOVERAB procedure.

If after a cold start IMS abends before a DUMPQ or SNAPQ checkpoint is taken and IMS must again be restarted from a cold start, QCF can use the cold start checkpoint as the base checkpoint; RECOVERAB will recover the messages processed before the abend.

The following figure shows the system operation flow for the RECOVERAB procedure:

```
                    ┌─────────────────────────┐
                    │    IMS log data sets     │
                    │     (OLDS or SLDS)       │
                    └─────────────────────────┘
                                 │
                                 ▼
```

#1 SELECT

- Locate base checkpoint on LOGIN data sets
- Select messages and their queue manager processing
  records, beginning at the selected checkpoint
- Write selected messages to LOGOUT data set
- Write discarded messages to SCRAPSEL data set
- Print select report

#2 SORTA

- Read SORTIN data set (output of SELECT)
- Sort messages and their queue manager processing
  records into groups
- Write groups to SORTOUT data set

#3 CANCEL

- Read CANCLIN data set (output of SORTA)
- Analyze message activity for each message and delete
  messages that have been successfully processed
- Write discarded messages to SCRAPCAN data set
- Write unprocessed messages to CANCLOUT data set
- Print cancel report

#4 SORTB

- Read SORTIN data set (output of SELECT)
- Sort messages into chronological order
- Write messages to SORTOUT data set

#5 LOAD

- Read LOAD data set (output of SORTB)
- Reload messages to IMS
- Write discarded messages to SCRAPLOG data set
- Print insert report

```
                    ┌─────────────────────────┐
                    │      IMS message         │
                    │      queue data set      │
                    └─────────────────────────┘
```

Figure 18. RECOVERAB procedure: system operation flow

Example: RECOVERAB procedure JCL

The following example JCL runs RECOVERAB.

```
//IQCRCAB  JOB (ACCOUNTINFO),NORMAL JOBCARD INFO
//********************************************************************
//*                                                                  *
//*          SAMPLE EXECUTION JCL - RECOVERAB                        *
//*                                                                  *
//********************************************************************
//*                                                                  *
//*          DEFINE FOR VSAM DATA SET TO STORE MESSAGE QUEUES        *
//*                                                                  *
```

```
//**********************************************************************
//DFNDS     EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=A
//* DEFINE FOR VSAM DATA SET SHOULD BE MODIFIED TO MEET YOUR
//* SYSTEM REQUIREMENTS. RECSZ PARAMETER, THE AVERAGE RECORD SIZE,
//* CAN BE SET TO THE LRECL OF THE IMS SHORT MESSAGE
//* QUEUE AND THE MAX VALUE MUST BE EQUAL TO OR GREATER THAN THE
//* IMS LONG MESSAGE QUEUE.
//* NOTE:
//*      IT IS RECOMMENDED THAT THE MAXIMUM RECORDSIZE BE SET
//*      TO THE MAXIMUM ALLOWED BY VSAM, IE, 32761.
//*      IT IS RECOMMENDED THAT THE CONTROLINTERVALSIZE BE SET
//*      TO THE MAXIMUM ALLOWED BY VSAM, IE, 32768.
//SYSIN    DD  *
   DEFINE CLUSTER (NAME(QCF.V3M1) -
              STORAGECLASS(PSTANDRD) -
              MANAGEMENTCLASS(PNOBACK) -
              IXD KEYS(4 0) -
              RECSZ(500 32761) -
              SPEED IMBED REPLICATE -
              REUSE SHAREOPTIONS(3 3)) -
       DATA    (NAME(QCF.V3M1.DATA) -
              CONTROLINTERVALSIZE(32768) -
              CYL(100,20)) -
       INDEX   (NAME(QCF.V3M1.INDEX) -
              CONTROLINTERVALSIZE(512) -
              TRACKS(1,1))
//**********************************************************************
//*                                                                    *
//*              SELECT STEP                                           *
//*                                                                    *
//**********************************************************************
//SELECT   EXEC PGM=IQCSELCT,REGION=0M
//STEPLIB  DD  DISP=SHR,DSN=hlqual.SIQCLINK
//QCFIN    DD  *
CHKPT    97359070707
FUNCTION RECOVERAB
//*
//* THIS IS A SAMPLE OF MULTIPLE LOG TAPES WHERE EACH LOG HAS A
//* DIFFERENT DATA SET NAME, THUS REQUIRING SEPARATE DD STATEMENTS.
//*
//LOGIN    DD  DISP=OLD,UNIT=TAPE,LABEL=(1,SL),DSN=LOG1,
//             VOL=SER=IMSLG1
//         DD  DISP=OLD,UNIT=TAPE,LABEL=(1,SL),DSN=LOG2,
//             VOL=SER=IMSLG2
//*
//MSGQ     DD DSN=QCF.V3M1,
//             AMP=('BUFND=20,BUFNI=10'),
//             DISP=(OLD,KEEP,KEEP)
//SCRAPSEL DD DUMMY
//*
//*   THE LRECL AND BLKSIZE FOR SNAPQ DATA SET SHOULD EQUAL LRECL
//*   AND BLKSIZE FOR LOGIN DATA SET.
//*   HOWEVER, IT IS RECOMMENDED THAT THE MAXIMUM SIZES, LRECL=32756
//*   AND BLKSIZE=32760, BE USED.
//*
//SNAPQ    DD  UNIT=SYSDA,DISP=NEW,SPACE=(CYL,(5,5)),
//             DSN=&&SNAPQ,
//             DCB=(LRECL=32756,BLKSIZE=32760,RECFM=VB)
//*
//*   THE LRECL OF THE LOGOUT DATA SET MUST BE AT LEAST 256
//*   BYTES LONGER THAN THE LRECL OF THE LOGIN DATA SET
//*   TO ACCOMMODATE THE DFSMRQPF PREFIX.
//*   HOWEVER, IT IS RECOMMENDED THAT THE MAXIMUM SIZES, LRECL=32756
//*   AND BLKSIZE=32760, BE USED.
//*
//LOGOUT   DD  DSN=&&LOGOUT,UNIT=SYSDA,SPACE=(CYL,(5,5)),
//             DCB=(LRECL=32756,BLKSIZE=32760,RECFM=VB),
//             DISP=(NEW,PASS)
//SORTINA  DD  DSN=&&SORTINA,UNIT=SYSDA,SPACE=(TRK,(1,1)),
//             DCB=(LRECL=80,RECFM=FB,BLKSIZE=800),DISP=(NEW,PASS)
//SORTINB  DD  DSN=&&SORTINB,UNIT=SYSDA,SPACE=(TRK,(1,1)),
//             DCB=(LRECL=80,RECFM=FB,BLKSIZE=800),DISP=(NEW,PASS)
//SNAPDUMP DD  SYSOUT=A
//QCFPRINT DD  SYSOUT=A
//**********************************************************************
//*                                                                    *
//*              DELETE OF VSAM DATA SET                               *
//*                                                                    *
//**********************************************************************
```

```
//DLTDS    EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=A
//SYSIN    DD   *
    DELETE (QCF.V3M1)
//************************************************************************
//*                                                                     *
//*             SORTA JOB STEP                                          *
//*                                                                     *
//************************************************************************
//SORTA    EXEC PGM=SORT,REGION=0K,COND=(0,NE,SELECT),
//               PARM='CORE=MAX'
//SORTLIB  DD   DISP=SHR,DSN=SYS1.SORTLIB
//SORTIN   DD   DISP=(OLD,DELETE),DSN=*.SELECT.LOGOUT
//SORTOUT  DD   DSN=&&SORTOUT,UNIT=SYSDA,SPACE=(CYL,(5,5)),
//               DCB=(LRECL=32756,BLKSIZE=32760,RECFM=VB),
//               DISP=(NEW,PASS)
//SYSOUT   DD   SYSOUT=A
//SORTWK01 DD   UNIT=SYSDA,SPACE=(CYL,(20,10),,CONTIG)
//SORTWK02 DD   UNIT=SYSDA,SPACE=(CYL,(20,10),,CONTIG)
//SORTWK03 DD   UNIT=SYSDA,SPACE=(CYL,(20,10),,CONTIG)
//SORTWK04 DD    UNIT=SYSDA,SPACE=(CYL,(20,10))
//SYSIN    DD   DSN=&&SORTINA,DISP=(OLD,PASS)
//*
//************************************************************************
//*                                                                     *
//*             CANCEL JOB STEP                                         *
//*                                                                     *
//************************************************************************
//CANCEL   EXEC PGM=IQCCANCL,COND=((0,NE,SELECT),(0,NE,SORTA)),
//               REGION=0M
//STEPLIB  DD   DISP=SHR,DSN=hlqual.SIQCLINK
//CANCLIN  DD   DISP=(OLD,DELETE),DSN=*.SORTA.SORTOUT
//CANCLOUT DD   DSN=&&CANCLOUT,UNIT=SYSDA,SPACE=(CYL(5,5)),
//               DCB=(LRECL=32756,BLKSIZE=32760,RECFM=VB)
//SCRAPCAN DD   DSN=QCF.SCRAPCAN,UNIT=SYSDA,DISP=(NEW,CATLG),
//               DCB=(LRECL=32756,BLKSIZE=32760,RECFM=VB)
//QCFPRINT DD   SYSOUT=A
//SNAPDUMP DD SYSOUT=A


//************************************************************************
//*                                                                     *
//*             SORTB JOB STEP                                          *
//*                                                                     *
//************************************************************************
//SORTB    EXEC PGM=SORT,REGION=0K,PARM='CORE=MAX',
//               COND=((0,NE,SELECT),(0,NE,SORTA),(0,NE,CANCEL))
//SORTLIB  DD   DISP=SHR,DSN=SYS1.SORTLIB
//SORTIN   DD   DISP=(OLD,DELETE),DSN=*.CANCEL.CANCLOUT
//SORTOUT  DD   DSN=&&SORTOUT,UNIT=SYSDA,SPACE=(CYL,(5,5)),
//               DCB=(LRECL=32756,BLKSIZE=32760,RECFM=VB),
//               DISP=(NEW,PASS)
//SYSOUT   DD   SYSOUT=A
//SORTWK01 DD   UNIT=SYSDA,SPACE=(CYL,(20,10),,CONTIG)
//SORTWK02 DD   UNIT=SYSDA,SPACE=(CYL,(20,10),,CONTIG)
//SORTWK03 DD   UNIT=SYSDA,SPACE=(CYL,(20,10),,CONTIG)
//SORTWK04 DD    UNIT=SYSDA,SPACE=(CYL,(20,10))
//SYSIN    DD   DSN=&&SORTINB,DISP=(OLD,PASS)
//*
//************************************************************************
//*                                                                     *
//*             LOAD JOB STEP                                           *
//*                                                                     *
//************************************************************************
//LOAD     EXEC PGM=IQCINI0$,
//     PARM='BPEINIT=IQCBINI0,BPECFG=IQCIVPCF,IMSPLEX=IPLEX,QCFIMS=IMS1'
//STEPLIB  DD   DISP=SHR,DSN=hlqual.SIQCLINK
//         DD   DISP=SHR,DSN=IMSREL.RESLIB
//*
//QCFPRINT DD SYSOUT=A
//*
//LOAD     DD DISP=SHR,DSN=&&SORTOUT
//*
//* SCRAPLOG IS AN OPTIONAL DD STATEMENT. IT WILL CONTAIN THE
//* REJECTED MESSAGES FROM THE LOAD STEP.
//* DCB PARAMETERS SHOULD BE THE SAME AS THOSE FOR THE LOAD
//* DATA SET.
//* HOWEVER, IT IS RECOMMENDED THAT THE MAXIMUM SIZES FOR LRECL
//* AND BLKSIZE ARE USED.
//*
//SCRAPLOG DD DSN=QCF.SCRAPLOG,UNIT=SYSDA,DISP=(NEW,CATLG),
//               DCB=(LRECL=32756,BLKSIZE=32760,RECFM=VB),
```

```
//          SPACE=(CYL,(5,5))
//*
//SNAPDUMP DD SYSOUT=A
//*
//QCFIN DD DSN=USER.PRIVATE.PROCLIB(load),UNIT=SYSDA
```

Running JCL for RECOVERDM

Submit the RECOVERDM JCL to recover messages that were lost from the queues because an IMS cold start was necessary following system maintenance.

The RECOVERDM procedure recovers messages lost from the queues because an IMS cold start was necessary following system maintenance.

The following figure shows the system operation flow for the RECOVERDM procedure:

IMS log data sets
(OLDS or SLDS)

#1 SELECT
- Locate base checkpoint on LOGIN data sets
- Select 4002 log records from DUMPQ or PURGE CHKPT
- Write selected messages to LOGOUT data set
- Write discarded messages to SCRAPSEL data set
- Print select report

#2 SORTB
- Read SORTIN data set (output of SELECT)
- Sort messages into chronological order
- Write messages to SORTOUT data sets

#3 LOAD
- Read Load data set (output of SORTB)
- Reinsert messages to IMS
- Write discarded messages to SCRAPLOG data set
- Print insert report

IMS message
queue data set

Figure 19. RECOVERDM procedure: system operation flow

Example: RECOVERDM procedure JCL

The following example JCL runs RECOVERDM when programs DFSUARC0 and IQCFCKPT have been used to automatically find the checkpoint.

```
//IQCRCDM  JOB (ACCOUNTINFO),NORMAL JOBCARD INFO
//***************************************************************
//*                                                            *
//*           SAMPLE EXECUTION JCL - RECOVERDM                  *
//*                                                            *
//***************************************************************
//*                                                            *
//*           DEFINE FOR VSAM DATA SET TO STORE MESSAGE QUEUES  *
//*                                                            *
//***************************************************************
```

```
//DFNDS    EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=A
//* DEFINE FOR VSAM DATA SET SHOULD BE MODIFIED TO MEET YOUR
//* SYSTEM REQUIREMENTS. RECSZ PARAMETER, THE AVERAGE RECORD SIZE,
//* CAN BE SET TO THE LRECL OF THE IMS SHORT MESSAGE
//* QUEUE AND THE MAX VALUE MUST BE EQUAL TO OR GREATER THAN
//* IMS LONG MESSAGE QUEUE.
//* NOTE:
//*      IT IS RECOMMENDED THAT THE MAXIMUM RECORDSIZE BE SET
//*      TO THE MAXIMUM ALLOWED BY VSAM, IE, 32761.
//*      IT IS RECOMMENDED THAT THE CONTROLINTERVALSIZE BE SET
//*      TO THE MAXIMUM ALLOWED BY VSAM, IE, 32768.
//SYSIN    DD  *
   DEFINE CLUSTER (NAME(QCF.V3M1) -
               STORAGECLASS(PSTANDRD) -
               MANAGEMENTCLASS(PNOBACK) -
               IXD KEYS(4 0) -
               RECSZ(500 32761) -
               SPEED IMBED REPLICATE -
               REUSE SHAREOPTIONS(3 3)) -
        DATA    (NAME(QCF.V3M1.DATA) -
               CONTROLINTERVALSIZE(32768) -
               CYL(100,20)) -
        INDEX   (NAME(QCF.V3M1.INDEX) -
               CONTROLINTERVALSIZE(512) -
               TRACKS(1,1))
//***********************************************************************
//*                                                                    *
//*            SELECT STEP                                             *
//*                                                                    *
//***********************************************************************
//SELECT  EXEC PGM=IQCSELCT,REGION=0M
//STEPLIB  DD  DISP=SHR,DSN=hlqual.SIQCLINK
//QCFIN    DD  *
CHKPT    92359070707
FUNCTION RECOVERDM
//*
//* THIS IS A SAMPLE OF MULTIPLE LOG TAPES WHERE EACH LOG HAS A
//* DIFFERENT DATA SET NAME, THUS REQUIRING SEPARATE DD STATEMENTS.
//*
//LOGIN    DD  DISP=OLD,UNIT=TAPE,LABEL=(1,SL),DSN=LOG1,
//             VOL=SER=IMSLG1
//         DD  DISP=OLD,UNIT=TAPE,LABEL=(1,SL),DSN=LOG2,
//             VOL=SER=IMSLG2
//*
//MSGQ     DD  DSN=QCF.V3M1,
//             AMP=('BUFND=20,BUFNI=10'),
//             DISP=(OLD,KEEP,KEEP)
//*
//*   THE LRECL OF THE LOGOUT DATA SET MUST BE AT LEAST 256
//*   BYTES LONGER THAN THE LRECL OF THE LOGIN DATA SET
//*   TO ACCOMMODATE THE DFSMRQPF PREFIX.
//*   HOWEVER, IT IS RECOMMENDED THAT THE MAXIMUM SIZES, LRECL=32756
//*   AND BLKSIZE=32760, BE USED.
//*
```

```
//LOGOUT   DD  DSN=&&LOGOUT,UNIT=SYSDA,SPACE=(CYL,(5,5)),
//             DCB=(LRECL=32756,BLKSIZE=32760,RECFM=VB),
//             DISP=(NEW,PASS)
//SORTINB  DD  DSN=&&SORTINB,UNIT=SYSDA,SPACE=(TRK,(1,1)),
//             DCB=(LRECL=80,RECFM=FB,BLKSIZE=800),DISP=(NEW,PASS)
//QCFPRINT DD  SYSOUT=A
//SNAPDUMP DD  SYSOUT=A
//***********************************************************************
//*                                                                    *
//*            DELETE OF VSAM DATA SET                                 *
//*                                                                    *
//***********************************************************************
//DLTDS    EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=A
//SYSIN    DD  *
   DELETE (QCF.V3M1)
//***********************************************************************
//*                                                                    *
//*            SORTB JOB STEP                                          *
//*                                                                    *
//***********************************************************************
//SORTB   EXEC PGM=SORT,REGION=0K,PARM='CORE=MAX',
//             COND=((0,NE,SELECT))
//SORTLIB  DD  DISP=SHR,DSN=SYS1.SORTLIB
//SORTIN   DD  DISP=(OLD,DELETE),DSN=*.SELECT.LOGOUT
```

```
//SORTOUT  DD  DSN=&&SORTOUT,UNIT=SYSDA,SPACE=(CYL,(5,5)),
//          DCB=(LRECL=32756,BLKSIZE=32760,RECFM=VB),
//          DISP=(NEW,PASS)
//SYSOUT   DD  SYSOUT=A
//SORTWK01 DD  UNIT=SYSDA,SPACE=(CYL,(20,10),,CONTIG)
//SORTWK02 DD  UNIT=SYSDA,SPACE=(CYL,(20,10),,CONTIG)
//SORTWK03 DD  UNIT=SYSDA,SPACE=(CYL,(20,10),,CONTIG)
//SORTWK04 DD   UNIT=SYSDA,SPACE=(CYL,(20,10))
//SYSIN    DD  DSN=&&SORTINB,DISP=(OLD,PASS)
//*
//***********************************************************************
//*                                                                     *
//*              LOAD JOB STEP                                          *
//*                                                                     *
//***********************************************************************
//LOAD     EXEC PGM=IQCINI0$,
//    PARM='BPEINIT=IQCBINI0,BPECFG=IQCIVPCF,IMSPLEX=IPLEX,QCFIMS=IMS1'
//STEPLIB  DD  DISP=SHR,DSN=hlqual.SIQCLINK
//          DD  DISP=SHR,DSN=IMSREL.RESLIB
//*
//QCFPRINT DD SYSOUT=A
//*
//LOAD     DD DISP=SHR,DSN=&&SORTOUT
//*
//* SCRAPLOG IS AN OPTIONAL DD STATEMENT. IT WILL CONTAIN THE
//* REJECTED MESSAGES FROM THE LOAD STEP.
//* DCB PARAMETERS SHOULD BE THE SAME AS THOSE FOR THE LOAD
//* DATA SET.
//* HOWEVER, IT IS RECOMMENDED THAT THE MAXIMUM SIZES FOR LRECL
//* AND BLKSIZE ARE USED.
//*
//SCRAPLOG DD DSN=QCF.SCRAPLOG,UNIT=SYSDA,DISP=(NEW,CATLG),
//          DCB=(LRECL=32756,BLKSIZE=32760,RECFM=VB),
//          SPACE=(CYL,(5,5))
//*
//SNAPDUMP DD SYSOUT=A
//*
//QCFIN DD DSN=USER.PRIVATE.PROCLIB(load),UNIT=SYSDA
```

Batch processing with skeletal JCL

You can use the IMS Queue Control Facility batch logic to create RECOVERAB and RECOVERDM jobs from skeletal JCL.

Store the skeletal JCL in the QCFJCLIN DD library using member names IMSID*xx*AB and IMSID*xx*DM.

The RECOVERAB and RECOVERDM jobs that are created are stored in the QCFJCLOT DD library and have the same names as those that were used for the input skeletal JCL.

The IMS Queue Control Facility batch logic performs the following processing functions:

• Analyzes execution parameters, allocates all necessary data sets, and prepares a LIST.LOG run command for DBRC.

• Attaches the RECON utility program and analyzes the output to find the last valid checkpoint for the function type and to find all PRISLDs with log records for the function type.

• To determine the last valid checkpoint, the batch logic uses the following logic:

  – For RECOVERAB, the last valid checkpoint is the last DUMPQ, PURGE, or DSNAPQ. The log data sets consist of all PRISLDs that contain the selected checkpoints, plus all of the following PRISLDs.

  – For RECOVERDM, the last valid checkpoint is DUMPQ or PURGE. The log data sets consist of all PRISLDs that contain the selected checkpoints.

• Reads the skeletal JCL for the function type and the IMSID from the skeletal JCL library, replaces or creates the LOGIN DD concatenation statement if that statement is missing with all selected PRISLDs data sets in the SELECT step, and replaces or creates the CHKPT command statement if the CHKPT command statement is missing for the last valid checkpoint in the SELECT step.

• Stores the new JCL in the output JCL library using the name of the input skeletal JCL.

Sample skeletons to create RECOVERAB and RECOVERDM JCL

You can use sample JCL skeletons to create you RECOVERAB and RECOVERDM JCL.

Sample skeletons to create JCL to run batch processing logic are in the sample library that is delivered as part of the IMS Queue Control Facility package (*hlq*.SIQCSAMP).

The following member names are used for the sample library skeletons:

IQCA##AB
  Sample skeleton for the RECOVERAB procedure

IQCA##DM
  Sample skeleton for the RECOVERDM procedure

Use the sample skeletons to create JCL skeletons for every IMS system that requires RECOVERDM and RECOVERAB.

Store your skeletal JCL in the QCFJCLIN library using member names IMSID*xxfunction_type*.

The RECOVERAB and RECOVERDM jobs that are created during the batch run are stored in the QCFJCLOT library using the same names as those that were used for the input skeletal JCL.

The batch processing logic performs the following actions:

- Analyzes run parameters, allocates all necessary data sets, and prepares a LIST.LOG run command for DBRC.
- Attaches the RECON utility program (DSPURX00) and analyzes the output to locate the last valid checkpoint for the function type and to find all PRISLDs with log records for the function type.
- Reads the skeletal JCL for the function type and the IMSID from the skeletal JCL library, replaces or creates (if missing) the LOGIN DD concatenation with all selected PRISLDs data sets in the SELECT step, and replaces or creates (if missing) the CHKPT command statement for the last valid checkpoint in the SELECT step.
- Stores the new JCL in the output JCL library using the name of the input skeletal JCL.

Running IQCABAT0 from sample skeletons

You must run IQCABAT0 to prepare RECOVERAB and RECOVERDM JCL from sample skeletons.

You can use the batch processing logic JCL, IQCABAT0 to create RECOVERAB and RECOVERDM jobs from skeletal JCL. Store your skeletal JCL in the QCFJCLIN library using member names IMSID*xxfunction_type*.

The RECOVERAB and RECOVERDM jobs that are created are stored in the QCFJCLOT library using the same names as those that were used for the input skeletal JCL.

The batch processing logic JCL member, IQCABAT0 performs the following actions:

- Analyzes run parameters, allocates all necessary data sets, and prepares a LIST.LOG run command for DBRC.
- Attaches the RECON utility program (DSPURX00) and analyzes the output to locate the last valid checkpoint for the function type and to find all PRISLDs with log records for the function type:
  - For RECOVERAB, the last valid checkpoint is the last DUMPQ, PURGE, or DSNAPQ. The log data sets consist of all PRISLDs that contain the selected checkpoints, plus all following PRISLDs.
  - For RECOVERDM, the last valid checkpoint is DUMPQ or PURGE. The log data sets consist of all PRISLDs that contain the selected checkpoints.
- Reads the skeletal JCL for the function type and the IMSID from the skeletal JCL library, replaces or creates (if missing) the LOGIN DD concatenation with all selected PRISLDs data sets in the SELECT step, and replaces or creates (if missing) the CHKPT command statement for the last valid checkpoint in the SELECT step.
- Stores the new JCL in the output JCL library using the name of the input skeletal JCL. Member IQCARUN in *hlq*.SIQCSAMP contains the sample JCL for executing batch processing logic.

The following JCL is used for running batch processing logic:

```
//*****************************************************************/
//* JOB NAME = IQCARUN                                          */
//*                                                             */
//* DESCRIPTIVE NAME = SAMPLE EXECUTION JCL FOR                 */
//*                    THE IMS QUEUE CONTROL FACILITY           */
//*                                                             */
//* FUNCTION = EXECUTE A JOB TO PREPARE RECOVERAB/RECOVERDM     */
//* PROCEDURES USING RECON LISTING AND JCL SKELETONS            */
//*                                                             */
//*     QCF VERSION 3 RELEASE 2                                 */
//*     LICENSED MATERIALS - PROPERTY OF IBM                    */
//*     5697-N50 COPYRIGHT IBM CORP. 2000, 2014                 */
//*     ALL RIGHTS RESERVED.                                    */
//*     US GOVERNMENT USERS RESTRICTED RIGHTS -                 */
//*     USE, DUPLICATION OR DISCLOSURE RESTRICTED               */
//*     BY GSA ADP SCHEDULE CONTRACT WITH IBM CORP.             */
//*                                                             */
//*****************************************************************/
//*                                                             */
//IQCARUN   JOB (ACCOUNTINFO),NORMAL JOBCARD INFO
//*****************************************************************/
//*     EXEC IQCABAT0                                           */
//*     &prm1 - IMSID -  character, 4bytes                      */
//*     &prm2 - FUNCTION - character ,2 bytes, AB/DM/RE         */
//*     &prm3 - START DATE/TIME D=YYYY.DDD,T=HH:MM:SS           */
//*  for FUNCTIONS AB and DM-&prm3 is optional                 */
//*  it can be used to reduce the LIST.LOG output              */
//*  &prm3 should be earlier than the last PRISLD start time   */
//*   if present -RECON LIST.LOG  works with FROMDATE=&prm3     */
//*   if missing -RECON LIST.LOG  works without FROMTIME        */
//*  for FUNCTION RE -&prm3 is required                         */
//*   it is the start checkpoint date time for reprocess       */
//*    RECON LIST.LOGS works without FROMTIME                   */
//*****************************************************************/
//STEP1    EXEC PGM=IQCABAT0,PARM='&prm1,&prm2,&prm3'
//STEPLIB  DD DISP=SHR,DSN=hlqual.SIQCLINK
//         DD DISP=SHR,DSN=IMSREL.RESLIB
//         DD DISP=SHR,DSN=SYS1.LINKLIB
//*****************************************************************/
//* IMSDALIB DD is optional-used for RECON1 and RECON2          */
//* allocation.                                                 */
//* RECON1 and RECON2 are allocated using information           */
//* from  sources in next order:                                */
//*    - DD information in JCL (RECON1 and RECON2 DD)           */
//*    - IMSDALIB (RECON1 and RECON1 members)                  */
//*    - STEPLIB  (RECON1 and RECON2 members)                  */
//*****************************************************************/
//IMSDALIB DD DISP=SHR,DSN=imsdalib
//*
//SYSUDUMP DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//*
//*****************************************************************/
//* SYSIN    DD is optional                                     */
//* if not provided, SYSIN    is dynamically allocated          */
//* SYSIN is the input control cards file for DSPURX00          */
//*****************************************************************/
//SYSIN    DD DSN=sysin,UNIT=SYSDA,
//         DISP=SHR,
//         SPACE=(TRK,(1,1))
//*


//*****************************************************************/
//* SYSPRINT DD is optional                                     */
//* if not provided, SYSPRINT is dynamically allocated          */
//* SYSPRINT is the output file from LIST.LOG command           */
//*****************************************************************/
//SYSPRINT DD DSN=sysprint,UNIT=SYSDA,
//         DISP=SHR,
//         SPACE=(CYL,(10,10))
//*
//*****************************************************************/
//* RECON1 and RECON2 DD are optional                           */
//* if not provided, they are dynamically allocated             */
//* using information in IMSDALIB and STEPLIB                    */
//****************************************************************
//RECON1   DD DISP=SHR,DSN=recon1
//RECON2   DD DISP=SHR,DSN=recon2
```

```
//*
//****************************************************************/
//* QCFJCLIN is PO data set with JCL skeletons                  */
//* for IMSIDs and FUNCTIONS (AB/DM)                            */
//****************************************************************/
//QCFJCLIN DD DSN=qcfjclin,DISP=SHR
//*
//****************************************************************/
//* QCFJCLOT is PO data set to store output JCL                 */
//* for IMSIDs and FUNCTIONS (AB/DM)                            */
//****************************************************************/
//QCFJCLOT DD DSN=qcfjclot,DISP=SHR
//*
//
```

EXEC statement for IQCABAT0

The EXEC statement for IQCABAT0 tells the system which job to run, the IMSID, function type, and start date and time.

The EXEC statement must specify PGM=IQCABAT0.

The EXEC statement specify the following parameters:

- IMSID

- Function type (AB or DM)

- Start date and time, based on the function type:

  For RECOVERAB and RECOVERDM, the start date and time is optional and indicates the date and time to start the LIST.LOG command.

  The default date and time is the current date, minus 7 days.

IQCABAT0 DD statements

The DD statements for IQCABAT0 for RECOVERAB and RECOVERD specify the data sets that are needed by the IQCABAT0 JCL member for RECOVERAB and RECOVERDM.

The STEPLIB DD statement of IQCABAT0 job the must contain the following parameters:

- *hlq*.SIQCLINK

- IMS RESLIB

- SYS1.LINKLIB

The IMSDALIB DD statement is used to dynamically allocate RECON data sets.

The SYSIN DD statement is the standard input data set for DBRC commands. If this statement is not provided, SYSIN is dynamically allocated.

The SYSPRINT DD statement is the standard output data set for DBRC commands. If this statement is not provided, SYSPRINT is dynamically allocated.

The RECON1 and RECON2 DD statements need to be provided. If these statements are not provided, RECON1 and RECON2 are dynamically allocated.

The QCFJCLIN DD library statement contains the skeletal JCL for RECOVERAB and RECOVERDM functions for each IMSID.

The QCFJCLOT DD library statement is used to store the new RECOVERAB and RECOVERDM JCL.

Example: Using the batch processing logic to prepare RECOVERDM JCL for IMSID IMA8:

- Skeletal JCL is created in the QCFJCLIN data set with the name IMA8*xx*DM. Run the IQCARUN routine and the parameters in the EXEC statement are PARM='IMA8,DM' (D=and T= are optional.)

- Member IMA8*xx*DM in the QCFJCLOT data set contain the prepared LOGIN DD and CHKPT statements.

IQCA*xx*AB
   Member IQCA*xx*AB in *hlq*.SIQCSAMP contains the sample skeletal JCL for function RECOVERAB.

IQCA*xx*DM
>   Member IQCA*xx*DM in *hlq*.SIQCSAMP contains the sample skeletal JCL for function RECOVERDM.

Locating the DUMPQ checkpoint for RECOVERDM

Use the program IQCFCKPT to create a job stream to automatically locate the DUMPQ checkpoint so that you can submit a RECOVERDM job to restart IMS after a cold start.

You can use the program IQCFCKPT to create a job stream that will automatically locate the DUMPQ checkpoint. You can then submit a RECOVERDM job to restart IMS after a cold start.

The IMS Log Archive utility (DFSUARC0) creates a user-defined data set containing all DUMPQ log records required for IMS Queue Control Facility to recreate the IMS message queue after a cold start.

IQCFCKPT reads through the data set of IMS log records created by DFSUARC0 and creates the required CHECKPOINT command for the last DUMPQ checkpoint in the data set. IQCFCKPT also creates the appropriate checkpoint control card for the last SNAPQ simple checkpoint or reset checkpoint in the file.

Use the following process to locate the checkpoint and then restart after an IMS cold start.

* Run DFSUARC0 to capture all the DUMPQ records IMS Queue Control Facility needs for running the RECOVERDM procedure
* Run IQCFCKPT to identify and create the IMS Queue Control Facility checkpoint control card
* Run SELECT to select all IMS log records necessary to recover the message queue
* Run SORT to put the records in the proper sequence for insertion to the message queue
* Execute LOAD to the selected, sorted messages back on the message queue

The following example RECOVERDM procedure shows sample JCL to locate DUMPQ checkpoint:

```
//RECOVER  JOB 'RECOVER AFTER DUMPQ',MSGCLASS=H,MSGLEVEL=(1,1),
//         TIME=1440,CLASS=S,USER=USRT001,NOTIFY=USRT001
//*
/*ROUTE    PRINT THISCPU/IMSTST13
//*
//*            Auto Restart after IMS DUMPQ
//*            ---------------------------
//*
//*********************************************************************
//*            DELETE OLD DATASETS                                    *
//*********************************************************************
//DELETE   EXEC PGM=IEFBR14
//SYSPRINT DD  SYSOUT=H
//DD1    DD  DSN=USRT001.ARCHLOG.CPFODMPQ,UNIT=SYSDA,DISP=(MOD,DELETE),
//        SPACE=(TRK,(1))
//DD2    DD  DSN=USRT001.ARCHLOG.DFSSLOGP,UNIT=SYSDA,DISP=(MOD,DELETE),
//        SPACE=(TRK,(1))
//DD3    DD  DSN=USRT001.ARCHLOG.DFSMRQ,UNIT=SYSDA,DISP=(MOD,DELETE),
//        SPACE=(TRK,(1))
//DD4    DD  DSN=USRT001.ARCHLOG.MSGQ,UNIT=SYSDA,DISP=(MOD,DELETE),
//        SPACE=(TRK,(1))
//DD5    DD  DSN=USRT001.ARCHLOG.SNAPQ,UNIT=SYSDA,DISP=(MOD,DELETE),
//        SPACE=(TRK,(1))
//DD6    DD  DSN=USRT001.ARCHLOG.LOGOUT,UNIT=SYSDA,DISP=(MOD,DELETE),
//        SPACE=(TRK,(1))
//DD7    DD  DSN=USRT001.ARCHLOG.SCRAPSEL,UNIT=SYSDA,DISP=(MOD,DELETE),
//        SPACE=(TRK,(1))
//DD8    DD  DSN=USRT001.ARCHLOG.SORTOUTB,UNIT=SYSDA,DISP=(MOD,DELETE),
//        SPACE=(TRK,(1))
//DD9    DD  DSN=USRT001.ARCHLOG.SCRAPLOG,UNIT=SYSDA,DISP=(MOD,DELETE),
//        SPACE=(TRK,(1))
//DD10 DD  DSN=USRT001.ARCHLOG.ISRTFIL,UNIT=SYSDA,DISP=(MOD,DELETE),
//        SPACE=(TRK,(1))
//*
//*********************************************************************
//*          Run IMS Log Archive Utility to copy DUMPQ records.     *
//*********************************************************************
//ARCHIVE EXEC PGM=DFSUARC0,PARM='DBRC=NO'
//*        PARM='SYS1,DBRC=NO'
//STEPLIB  DD  DISP=SHR,DSN=IMSBLD.I71RTS52.ARESLIB
//*TEPLIB  DD  DISP=SHR,DSN=IMSBLD.I810TS25.ARESLIB
//RECON1   DD  DSN=IMSTESTS.DSHR.RECON1,DISP=SHR
//RECON2   DD  DSN=IMSTESTS.DSHR.RECON2,DISP=SHR
```

```
//RECON3   DD  DSN=IMSTESTS.DSHR.RECON3,DISP=SHR
//*
//DFSSLDSP DD  DISP=SHR,DSN=MRQT.IMS81R.MULTI.DUMPQ,
//             VOL=SER=IMSDCL,UNIT=SYSDA
//*
//DFSSLOGP DD  DSN=USRT001.ARCHLOG.DFSSLOGP,DISP=(,CATLG),
//             DCB=(RECFM=VB,LRECL=22524,BLKSIZE=22528),
//             UNIT=SYSDA,SPACE=(CYL,(1,1),RLSE)
//DFSMRQ   DD  DSN=USRT001.ARCHLOG.DFSMRQ,DISP=(,CATLG),
//             DCB=(RECFM=VB,LRECL=22524,BLKSIZE=22528),
//             UNIT=SYSDA,SPACE=(CYL,(1,1),RLSE)
//*
//SYSPRINT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//SYSIN    DD   *
 COPY  DDNOUT1(DFSMRQ) -
    RECORD (O(5) T(X) V(4001) L(2) C(E)) -
    RECORD (O(5) T(X) V(4002) L(2) C(E)) -
    RECORD (O(5) T(X) V(4098) L(2) C(E)) -
    RECORD (O(5) T(X) V(4099) L(2) C(E))
 /*

//**********************************************************************
//*           Locate DUMPQ checkpoint on user-generated log          *
//**********************************************************************
//FINDCK  EXEC PGM=IQCFCKPT,REGION=0M
//STEPLIB  DD  DISP=SHR,DSN=IMSVS.QCFV3.LOADLIB
//SYSUDUMP DD  SYSOUT=*
//SYSPRINT DD  SYSOUT=*
//CKPTIFIL DD  DSN=MRQT.IMS81R.MULTI.DUMPQ,UNIT=SYSDA,
//             VOL=SER=IMSDCL,DISP=SHR
//CKPTOFIL DD  SYSOUT=*,DCB=(RECFM=FA,LRECL=137)
//CPFOSIMP DD  SYSOUT=*,DCB=(RECFM=F,LRECL=80)
//CPFORSRT DD  SYSOUT=*,DCB=(RECFM=F,LRECL=80)
//CPFODMPQ DD  DSN=USRT001.ARCHLOG.CPFODMPQ,
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=0),
//             UNIT=SYSDA,SPACE=(TRK,(1)),
//             DISP=(,CATLG)
//CPFOSNPQ DD  SYSOUT=*,DCB=(RECFM=F,LRECL=80)
//CPFOQDMP DD  SYSOUT=*,DCB=(RECFM=F,LRECL=80)
//*
//**********************************************************************
//*           DEFINE VSAM MSGQ FILE                                  *
//**********************************************************************
//MSGQALL  EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=H
//SYSIN    DD   *
  DELETE USRT001.ARCHLOG.CLUSTER
  SET MAXCC=0
  DEFINE CLUSTER (NAME(USRT001.ARCHLOG.CLUSTER) -
                  VOL(000000) -
                  IXD KEYS(4 0) -
                  RECSZ(500 30000)  -
                  SPEED IMBED REPLICATE -
                  REUSE SHAREOPTIONS(3 3) ) -
        DATA    (NAME(USRT001.ARCHLOG.DATA) -
                 CONTROLINTERVALSIZE(32760) -
                 CYL(01,05) ) -
        INDEX   (NAME(USRT001.ARCHLOG.INDEX) -
                 CONTROLINTERVALSIZE(512) -
                 TRACKS(10,10) )
 /*
//*
//**********************************************************************
//*           QCF SELECT STEP                                        *
//**********************************************************************
//SELECT  EXEC PGM=IQCSELCT,REGION=0M
//STEPLIB  DD  DISP=SHR,DSN=IMSVS.QCFV1.LOADLIB
//*                                 DATA FROM IMS LOG ARCHIVE UTILITY
//LOGIN    DD  DSN=USRT001.ARCHLOG.DFSMRQ,DISP=SHR
//LOGOUT   DD  DSN=USRT001.ARCHLOG.LOGOUT,UNIT=SYSDA,
//             DCB=(LRECL=32756,BLKSIZE=32760,RECFM=VB),
//             DISP=(,CATLG),VOL=SER=222222,SPACE=(CYL,(1,1),RLSE)
//MSGQ     DD  DSN=USRT001.ARCHLOG.CLUSTER,
//             AMP=('BUFND=20,BUFNI=10'),
//             DISP=(OLD,KEEP,KEEP)
//SCRAPSEL DD  DUMMY
//SORTINB  DD  DSN=&&SORTINB,SPACE=(TRK,(1,1)),DISP=(NEW,PASS),
//             UNIT=SYSVIO,
//             DCB=(LRECL=80,RECFM=FB,BLKSIZE=800)
//SYSUDUMP DD  SYSOUT=H
```

```
//QCFIN     DD  DSN=USRT001.ARCHLOG.CPFODMPQ,DISP=SHR
//          DD  DSN=IMSVS.ARCHLOG.QCFIN,DISP=SHR
//QCFPRINT DD  SYSOUT=H
//SNAPDUMP DD  SYSOUT=H
```

```
//********************************************************************
//*      SORT QCF SELECTED LOG RECORDS TO  IN COLD STARTED IMS      *
//********************************************************************
//SORTB    EXEC PGM=SORT,REGION=0M,PARM='CORE=MAX',
//         COND=(0,NE,SELECT)
//SORTLIB  DD  DISP=SHR,DSN=SYS1.SORTLIB
//SORTIN   DD  DISP=SHR,DSN=USRT001.ARCHLOG.LOGOUT,
//             UNIT=SYSDA,VOL=SER=222222
//SORTOUT  DD  DSN=USRT001.ARCHLOG.ISRTFIL,
//             UNIT=SYSDA,SPACE=(TRK,(5,2)),
//             DCB=(LRECL=32756,BLKSIZE=32760,RECFM=VB),
//             DISP=(NEW,CATLG),VOL=SER=000000
//SYSOUT   DD  SYSOUT=H
//SORTWK01 DD  UNIT=SYSVIO,SPACE=(CYL,(1,1))
//SORTWK02 DD  UNIT=SYSVIO,SPACE=(CYL,(1,1))
//SORTWK03 DD  UNIT=SYSVIO,SPACE=(CYL,(1,1))
//SORTWK04 DD  UNIT=SYSVIO,SPACE=(CYL,(1,1))
//SYSIN    DD  DSN=&&SORTINB,DISP=(OLD,PASS)
//*
//********************************************************************
//*           SORTED QCF SELECTED DUMPQ LOG RECORDS
//********************************************************************
//LOAD     EXEC PGM=IQCINI0$,
//     PARM='BPEINIT=IQCBINI0,BPECFG=IQCIVPCF,IMSPLEX=IPLEX,QCFIMS=IMS1'
//STEPLIB  DD  DISP=SHR,DSN=hlqual.SIQCLINK
//         DD  DISP=SHR,DSN=IMSREL.RESLIB
//*
//QCFPRINT DD SYSOUT=A
//*
//LOAD     DD DISP=SHR,DSN=QCF.UNLOAD
//*
//* SCRAPLOG IS AN OPTIONAL DD STATEMENT. IT WILL CONTAIN THE
//* REJECTED MESSAGES FROM THE LOAD STEP.
//* DCB PARAMETERS SHOULD BE THE SAME AS THOSE FOR THE LOAD
//* DATA SET.
//* HOWEVER, IT IS RECOMMENDED THAT THE MAXIMUM SIZES FOR LRECL
//* AND BLKSIZE ARE USED.
//*
//SCRAPLOG DD DSN=QCF.SCRAPLOG,UNIT=SYSDA,DISP=(NEW,CATLG),
//         DCB=(LRECL=32756,BLKSIZE=32760,RECFM=VB),
//         SPACE=(CYL,(5,5))
//*
//SNAPDUMP DD SYSOUT=A
//*
//QCFIN DD DSN=USER.PRIVATE.PROCLIB(load),UNIT=SYSDA
```

In step FINDCK, program IQCFCKPT prepares the CHECKPOINT command that is used by the RECOVERDM function. This statement is stored in data set DSN=USRT001.ARCHLOG.CPFODMPQ (DDNAME CPFODMPQ in the same step).

The CPFODMPQ data set, concatenated with the IMSVS.ARCHLOG.QCFIN data set, is the QCFIN for the IQCSELCT program in the SELECT step.

In the IMSVS.ARCHLOG.QCFIN data set, you should prepare the FUNCTION and SELECT statements.

The following example shows how to do this for IMSVS.ARCHLOG.QCFIN:

```
FUNCTION RECOVERDM
SELECT   TYPE=DIO
```

Replacing the user exit routines

You can replace the dummy user exit routines by relinking the individual load modules.

Unlike IMS, which LOADs its user exit routines, IMS Queue Control Facility has some of its user exit routines link-edited into the appropriate load modules.

These exit routines are called by the similarly-named functions. The exit routines let you perform the following tasks:

- Include or exclude messages from processing, useful if you have special needs not satisfied by the INCLUDE/EXCLUDE control statements.

  You might want, for example, to delete messages over a certain age.
- Modify parts of a message before requeuing the message.

  You might, for example, want to change the destination SYSID from one system to another based on the content of the message.
- Requeue messages marked nonrecoverable.

  Transaction messages from an APPC synchronous conversation session are flagged by IMS as nonrecoverable, regardless of whether the transaction is recoverable.

  But if these messages were sent to a recoverable IMS transaction, IMS logs the user segments.

  This process makes it possible for IMS Queue Control Facility to recover these messages even though the nonrecoverable flag is set.

To replace the dummy user exit routine, you must relink the individual load modules.

These user exit routines that are link-edited, and their corresponding load modules are as follows:

```
IQCCANCL - IQCCANX0
IQCSELCT - IQCSELX0
```

Replacing IQCCANX0

You can replace IQCCANX0 by supplying your own Cancel user exit routine.

The following example code shows how you could supply your own Cancel user exit routine (IQCCANX0):

```
//LINKCAN  EXEC PGM=IEWL,PARM='LIST,LET,XREF'
//QCFPRINT DD   SYSOUT=A
//SYSUT1   DD   &&LUT1,SPACE=(CYL,(10,10)),UNIT=VIO,DISP=(,PASS)
//SYSLMOD  DD   DSN=hlqual.SIQCLINK,DISP=SHR
//OBJ      DD   DSN=QCF.IQCCANX0.OBJ,DISP=SHR
//SYSLIN   DD   *
 INCLUDE OBJ
 INCLUDE SYSLMOD(IQCCANCL)
 ENTRY   IQCCANCL
 NAME    IQCCANCL(R)
/*
```

The data set pointed to by the OBJ DD statement can contain either an object file replacement of IQCCANX0 or a link-edited load module of IQCCANX0.

Replacing IQCSELX0

You can replace IQCSELX0 by supplying your own Select user exit routine.

The following example code shows how you could supply your own Select user exit routine (IQCSELX0):

```
//LINKSEL  EXEC PGM=IEWL,PARM='LIST,LET,XREF'
//QCFPRINT DD   SYSOUT=A
//SYSUT1   DD   &&LUT1,SPACE=(CYL,(10,10)),UNIT=VIO,DISP=(,PASS)
//SYSLMOD  DD   DSN=hlqual.SIQCLINK,DISP=SHR
//OBJ      DD   DSN=QCF.IQCSELX0.OBJ,DISP=SHR
//SYSLIN   DD   *
 INCLUDE OBJ
 INCLUDE SYSLMOD(IQCSELCT)
 ENTRY   IQCSELCT
 NAME    IQCSELCT(R)
/*
```

The data set that the OBJ DD statement points to can contain either an object file replacement of IQCSELX0 or a link-edited load module of IQCSELX0.

Chapter 16. Reprocessing messages

The REPROCESS function reprocesses messages in both the shared- and nonshared-queues environments after an application error has occurred.

Topics:

REPROCESS function description

Use the REPROCESS function to reinsert messages that have been processed into the shared and nonshared message queues.

If an application program has processed messages incorrectly because of a logic error, you can reprocess the messages after the program error is corrected.

⚠️ **Attention:** Database batch backout or database recovery is required before using the REPROCESS function to avoid double updating.

You can locate checkpoints and mass-insert transactions into the IMS message queue by concatenating several system logs.

REPROCESS can be used for stress, regression, or application testing an IMS system in both the shared and nonshared queue environment.

Reprocess logic components

In the nonshared-queues environment, IMS Queue Control Facility reprocess consists of two logic components and requires the use of a sort package.

The two logic components are as follows:

Select logic
Select logic selects messages from the IMS SLDSs (based on your specifications) for reinsertion into the IMS message queue.

Load function
The load function is always run as the last step of recovery processing.

The purpose of the load function is to reinsert the messages that were selected by the select logic or cancel logic into the IMS message queue.

Reprocess processing (select logic)

Use the REPROCESS select logic to create a data set of messages that are read into the message queue.

IMS Queue Control Facility select logic creates a LOGOUT data set containing message records that are selected from the IMS SLDSs.

The LOGOUT data set that the select logic creates, provides the messages that will eventually be read into the IMS message queue.

For the REPROCESS function, SELECT logic selects records from an input SLDS for eventual reinsertion into the IMS message queue by the load function. SELECT selects the records by doing the following tasks:

1. Processing control statements:
   - Identifying the base checkpoint from the CHKPT statement, if you specify the simple CHKPT control statement format without the LAST option. Identifying the logs, dynamically allocating the logs, and identifying the base checkpoint from the CHKPT statement, if you specify the CHKPT control statement with the LAST option
   - Determining log record selection criteria for date and time stamps, log sequence number, and record type from the SELECT statement
   - Determining log record selection criteria for message sources and destinations from the INCLUDE and EXCLUDE statements
2. Locating the base checkpoint in the input SLDS
3. Passing control to the correct program function logic to select the correct record types
4. Processing system log records using type-specific record processing logic
5. Writing records that fail to meet the selection criteria of SELECT and INCLUDE statements to the SCRAPSEL data set
6. Writing records that do meet the selection criteria of EXCLUDE statements to the SCRAPSEL data set
7. Writing successfully processed records to the LOGOUT data set to be further processed by subsequent IMS Queue Control Facility job steps
8. Terminating at end-of-file on the input SLDS

Selection parameters

IMS Queue Control Facility select logic selects messages from the log data sets based on the following two parameters that you specify with control statements:

Base checkpoint
Selection criteria

Base checkpoint

Any checkpoint can serve as the base checkpoint for REPROCESS processing. The checkpoint is the starting point for log record selection. Select logic selects only log records that occur in or follow the base checkpoint.

Selection criteria

You specify selection criteria to the select logic with SELECT, INCLUDE, and EXCLUDE statements. If none of these statements is present, all records that can be selected for REPROCESS are selected.

Message records can be selected based on the following criteria:

Specified in SELECT statement:

Beginning date and time
Ending date and time
Beginning log record sequence number
Ending log record sequence number
Log record type
System messages or no system messages
Conversational messages or no conversational messages

Specified in INCLUDE or EXCLUDE statement:

Destination with optional system ID

Logical unit name
Logical unit name destination
Logical unit sidename
Logical unit tpname
OTMA tmember name
OTMA tmember name destination
OTMA tmember name source
OTMA tpipe name
Source with optional system ID
VTAM name

The SELECT statement specifies global criteria, the range of times or log sequence numbers, and whether input, output, or both types of log records should be selected by the select logic. Only one SELECT statement is allowed.

INCLUDE and EXCLUDE statements specify selection criteria specific to message sources and destinations. You can have any number of INCLUDE and EXCLUDE statements.

If any INCLUDE statements are present, a record must meet all criteria for at least one INCLUDE statement to be included in the select logic selections. Unspecified criteria are considered to be met.

A record that meets all criteria specified by an EXCLUDE statement is excluded from the select logic selections, even if the record meets all criteria specified by the SELECT statement and an INCLUDE statement. Unspecified criteria are considered to be met.

Selection processing

After processing the control statements, the select logic searches the SLDS for the base checkpoint. The base checkpoint is located by finding a particular record identifying the checkpoint that is specified by the CHKPT control statement. The location of that particular record signifies the beginning of a checkpoint.

Control is passed to the REPROCESS logic to select the proper log record types.

If the record does not meet any of the selection criteria, the record is rejected and written to the SCRAPSEL data set. If the record meets the criteria for an EXCLUDE statement, the record is rejected and written to the SCRAPSEL data set. Otherwise, the record is written to the LOGOUT data set.

When end-of-file is reached on the input SLDS, the select logic writes a complete report of log record processing statistics to the QCFPRINT data set and terminates normally.

The SCRAPSEL data set that is produced by this step contains the messages that were rejected because they met the selection parameters specified for this step.

If a route code has been entered for WTO messages, select logic gives control to the IMS Queue Control Facility PARMS logic for processing.

Control statements for REPROCESS function

The control statements that are used by REPROCESS are described in the following topics.

The processing logic for the SELECT job step performs the following tasks:

- Interpret each statement
- Validate the data
- Build control blocks
- Issue error messages
- Run the appropriate logic

Control statements for SELECT job step

Use the SELECT job step to specify the messages that you want read into the nonshared message queue.

To specify the messages that you want read into the message queue in a nonshared-queues environment, provide control statements to the select logic in the SELECT job step. These control statements specify the base checkpoint, the program function (which is also the name of the procedure to be followed), and the selection criteria to be used to select the messages.

The following control statements are recognized by the SELECT job step:

```
* comment
CHKPT
FUNCTION
INCLUDE
EXCLUDE
NOWTOMSG
SELECT
TITLE
WTOMSG
```

CHKPT control statement syntax

The CHKPT control statement specifies the base checkpoint. The base checkpoint is the checkpoint that the SELECT step uses as a starting point for selecting messages from the input log data set.

The syntax of the CHKPT statement is:

►►─ CHKPT ── D= *yyyy.ddd,* T= *hh:mm:ss.thmiju,*   Z= *shh:mm* ─►◄

or

►►─ CHKPT ── D= *yyyy.ddd,* T= *hh:mm:ss* ─►◄

In the first example:

> *yyyy.ddd* is the date
> *hh:mm:ss.thmiju* is the time
> *shh:mm* is the zone

Together, these three comprise the Universal Time Stamp. This represents the IMS checkpoint that is to be used as the base checkpoint.

- *yyyy* = year
- *ddd* = day
- *hh* = hour
- *mm* = minute
- *ss* = second
- *t* = tenths of a second
- *h* = hundredths of a second
- *m* = milliseconds
- *i* = tenths of a millisecond
- *j* = hundredths of a millisecond
- *u* = microseconds
- *s* = sign + or - from Greenwich
- *hh* = hour offset from Greenwich

- *mm*= minutes offset from Greenwich

Because most installations are in an hourly time zone, the zone can be represented simply as the sign (if negative or in other words, if West of GMT (Greenwich mean time.)) and the offset.

For example, San Jose, CA is Z=-7 or Z=-8; Uithoorn in the Netherlands is Z=1 or Z=2; GMT is Z=0. If minutes are required, such as for Gander, Newfoundland, the leading zero can be omitted (for example, Z=-3:30).

All date and time stamps are in UTC (coordinated universal time) format. The zone is added to the UTC to produce local time.

In the second example, *yyyy.ddd* is the date, and *hh:mm:ss* is the local time as the local time appears in the IMS checkpoint timestamp message DFS994I.

Another option for the CHKPT control statement is to specify a value of 0 or 00000000000, starting at column 10. This allows SELECT to select the first valid checkpoint that SELECT encounters for that function type as the base checkpoint according to the following rules.

- For RECOVERAB, the first DUMPQ or SNAPQ checkpoint is selected.
- For RECOVERDM, the first DUMPQ checkpoint is selected.
- For REPROCESS, the first checkpoint is selected.

CHKPT control statement syntax, with zero values as input:

The alternate syntax of the CHKPT control statement lets you run SELECT with automatic checkpoint location, automatic log selection, and dynamic allocation.

To use this alternate syntax of CHKPT, add DD statements for automatic checkpoint location, automatic log selection, and dynamic allocation to the SELECT job step in the RECOVERAB, RECOVERDM, and REPROCESS procedures.

These DD statements include:

- RECON data sets
- IMSDALIB
- Sample log LOGIN

The alternate syntax of this version of the CHKPT control statement is:
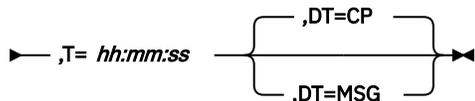
The fields are defined as follows:

LAST
Indicates that the last valid checkpoint for the procedure type should be used.

imsid
The 4-byte IMSID.

function
One of the following procedure types:

AB
RECOVERAB procedure

DM
>   RECOVERDM procedure

RE
>   REPROCESS procedure

D=*yyyy.ddd*
>   For RECOVERDM and RECOVERAB, this parameter is optional and D=*yyyy.ddd* is the start date for the LIST.LOG RECON command. If a value for D= is omitted, D=*yyyy.ddd* defaults to the current date minus 7 days.
>
>   For REPROCESS, this parameter is required and D=*yyyy.ddd* is the time of the selected checkpoint.

T=*hh:mm:ss*
>   For RECOVERDM and RECOVERAB, this parameter is optional and T=*hh:mm:ss* is the start time for the LIST.LOG RECON command. If a value for T= is omitted, T=*hh:mm:ss* defaults to 00:00:00.
>
>   For REPROCESS, this parameter is required and T=*hh:mm:ss* is the time of the selected checkpoint.

DT=[CP | MSG]
>   Parameter DT=CP will start with the checkpoint, with timestamp equal to D=*yyyy.ddd*,T=*hh:mm:ss* local time. All messages that have been on the queue (and not processed) at the time of the checkpoint, and all messages that have been inserted after the checkpoint, will be selected.
>
>   Message selection can be restricted with the following SELECT statement:

```
SELECT STARTTIME=D=yyyy.ddd,T=hh:mm:ss
```

>   Parameter DT=MSG will select all messages that have been inserted at and after D=*yyyy.ddd*,T=*hh:mm:ss* local time. With this parameter setting, an earlier checkpoint will be used as the starting point. All messages that have been inserted after the specified date/time will be selected.
>
>   **Note:** Messages loaded with the LOAD function preserve the date and time of their original creation, and will not be selected if DT=MSG is specified, even if the messages are reloaded after the specified time.
>
>   The two DT settings can produce significantly different results. DT=CP usually selects more messages. This setting selects messages that have been created at an earlier point and have not been processed until the checkpoint (these messages will be in the checkpoint records). DT=MSG selects only messages that have been created at and after the selected D=*yyyy.ddd*,T=*hh:mm:ss*.
>
>   However, the following two sets of statements will create the same output:

```
CHKPT     LAST,SYS3,RE,D=2020.290,T=13:38:11,DT=CP
FUNCTION REPROCESS
SELECT   STARTTIME=D=2020.290,T=11:38:12.000000
```

>   and

```
CHKPT     LAST,SYS3,RE,D=2020.290,T=13:38:12,DT=MSG
FUNCTION REPROCESS
```

FUNCTION control statement syntax

The FUNCTION control statement must follow certain syntactic specifications.

FUNCTION control statement syntax:
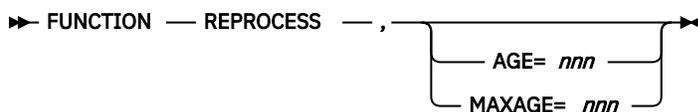
INCLUDE control statement syntax

The INCLUDE control statement must follow certain syntactic specifications.

INCLUDE control statement syntax:
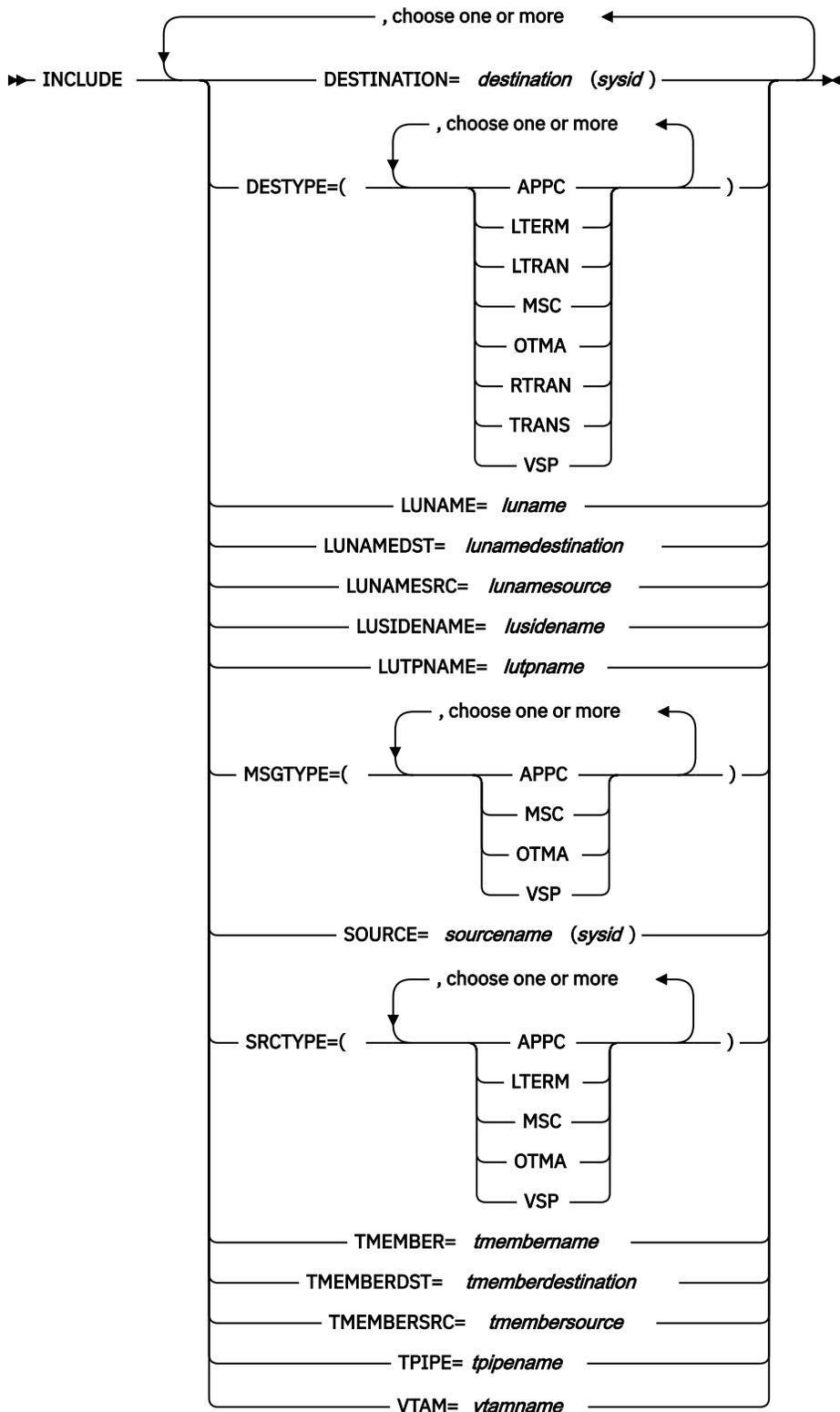
SELECT control statement syntax

The SELECT control statement must follow certain syntactic specifications.

SELECT control statement syntax:
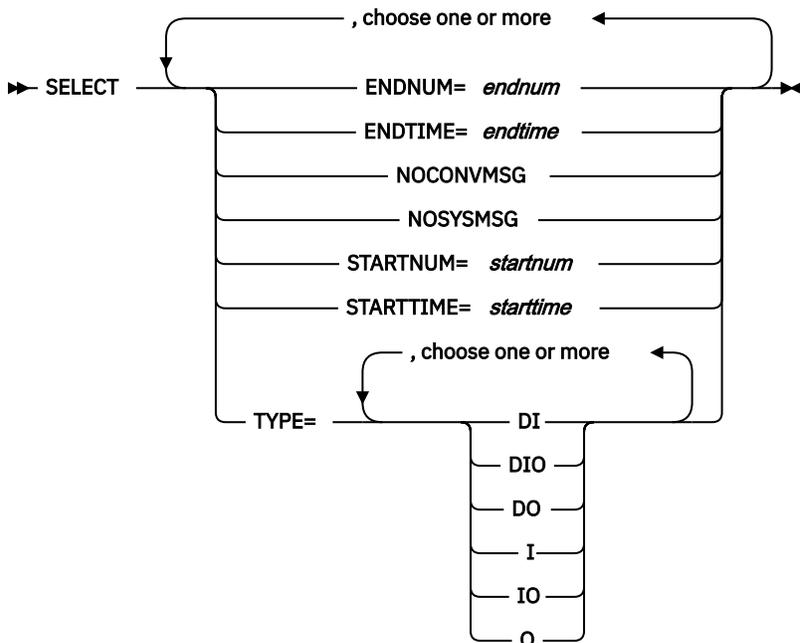
Job steps for REPROCESS

The job steps that are required for the REPROCESS function are as follow, in this order: SELECT, SORTB, LOAD.

Load function processing for rebuilding the queues

Use the load function to requeue messages to the message queues and produces a data report.

For reprocessing, the SELECT step is performed by the IMS Queue Control Facility select logic and LOAD step is performed by the load function.

The load function requeues messages to the message queues and produces a data report.

The load function inserts messages to the message queues from the data set that is specified in the LOAD DD statement. This load data set was produced by one of the following functions:

• BROWSE

• UNLOAD

• RECOVER

• SORT output in the rebuilding procedure

The load function selects records to be inserted from the load data set to the message queues and performs the following tasks:

1. Processes control blocks that were built by IMS Queue Control Facility control processing logic from input control statements:

    • Determines queue selection from the QUEUETYPE= keyword of the FUNCTION statement

    • Determines record selection criteria for date and time stamps, and record type from the SELECT statement

- Determines record selection criteria for message sources and destinations from the INCLUDE and EXCLUDE statements
2. Discards and ignores records that fail to meet the selection criteria of the SELECT, INCLUDE, and EXCLUDE statements
3. Loads records that do meet selection criteria of the SELECT and INCLUDE statements to the message queues
4. Produces reports on the number and types of records the load function loaded to the message queues
5. Produces a report of the messages that IMS failed to load, with the AIBREASN code identifying the error, and writes those messages to the SCRAPLOG data set.
6. Returns control to IMS Queue Control Facility control processing logic

Selection parameters

Based on your selection criteria, the load function selects messages from the LOAD data set to load into the message queues. You specify the selection criteria in control statements that are read by the IMS Queue Control Facility control processing logic.

Selection criteria

You specify selection criteria to the load function with FUNCTION, SELECT, INCLUDE, and EXCLUDE statements. Use these statements to define the subset of message records from the LOAD data set to be inserted to the message queues.

The SELECT, INCLUDE, and EXCLUDE statements are optional.

If none of these three statements is present, all of the records in the LOAD data set will be inserted.

You can select message records based on the following criteria:

- As specified in the FUNCTION statement with the following keyword and parameters:
  - Queue type
    - ALL
    - APPC
    - DEADQ
    - LTERM
    - OTMA
    - Remote MSNAME queue
    - Serial transaction queue
    - Suspend transaction queue
    - Transaction ready queue (local and remote)
- As specified in the SELECT statement with the following parameters:

  Beginning date and time
  Conversational messages or no conversational messages
  Ending date and time
  Log record type
  System messages or no system messages
- As specified in the INCLUDE or EXCLUDE statement with the following parameters:

  Destination with optional system ID
  Logical unit name
  Logical unit name destination
  Logical unit sidename
  Logical unit tpname

OTMA tmember name

OTMA tmember name destination

OTMA tmember name source

OTMA tpipe name

Source with optional system ID

Specific IMS ID

Specific IMS ID destination

Specific IMS ID source

Either the originating unit-of-work, or the processing IMS ID, or both

Either the originating unit-of-work, or the processing token, or both

VTAM name

The FUNCTION statement QUEUETYPE= keyword specifies one or more message queues to be processed.

The SELECT statement specifies global criteria, a range of times, and the types of records to be selected by the load function for insertion. Only one SELECT statement is allowed.

INCLUDE and EXCLUDE statements specify selection criteria specific to message sources and destinations. INCLUDE and EXCLUDE statements are unrestricted in number.

If any INCLUDE statements are present, a record must meet all criteria for at least one INCLUDE statement to be included in the load function selections. Unspecified criteria are considered to be met.

A record that meets all criteria specified by an EXCLUDE statement will be excluded from the load function selections even if the record meets all criteria specified by the SELECT statement and an INCLUDE statement. Unspecified criteria are considered to be met.

Selection processing

When given control by IMS QCF control processing logic, the load function determines if a load data set is defined. If the load data set is not present, the load function terminates with a message.

The load function issues calls to IMS requesting that a Common Queue Services (CQS) LOAD function be performed.

The load function determines if the record meets the selection criteria that is specified by any SELECT and INCLUDE statements. If the record does not meet these selection criteria, the record is not inserted. If the record meets the criteria for an EXCLUDE statement, the record is not inserted.

When end-of-file is reached on the data set pointed to by the LOAD DD statement, the load function produces reports on the number and type of messages the load function inserted to the message queues, and statistics on the number and types of records that were processed.

The load function then returns control to IMS QCF control processing logic for clean up and normal termination.

JCL for the REPROCESS function

The REPROCESS procedure reinserts previously processed messages into the message queues.

You can run the REPROCESS procedure in batch mode by submitting a JCL job stream.

The following list shows the system operation flow for the REPROCESS procedure:

- IMS log data sets, either OLDS or SLDS are used as input.
- The SELECT function performs the following tasks:
  - Locates the base checkpoint on LOGIN data sets.
  - Selects messages based on your selection criteria.
  - Writes selected messages to the LOGOUT data set.
  - Writes discarded messages to the SCRAPSEL data set.

- – Prints the Select report,
- The SORTB step performs the following tasks:
  - – Reads the SORTIN data set that was the output of SELECT.
  - – Sorts messages into chronological order.
  - – Writes messages to the SORTOUT data set.
- The LOAD function performs the following tasks:
  - – Reads the SORTOUT/LOAD data set that was output from SORTB.
  - – Reinserts messages to IMS.
  - – Writes discarded messages to the SCRAPLOG data set.
  - – Prints the Load report.
- Now the IMS message queue data set is populated.

Example JCL for REPROCESS procedure

The following example JCL runs the REPROCESS procedure:

```
//****************************************************************/
//* JOB NAME = IQCARPRS                                        */
//*                                                            */
//* DESCRIPTIVE NAME = SAMPLE EXECUTION JCL FOR                */
//*                    THE QUEUE CONTROL FACILITY              */
//*                                                            */
//* STATUS = VERSION 03  RELEASE 02  MODIFICATION LEVEL 00     */
//*                                                            */
//* FUNCTION = EXECUTION A FUNCTION REPROCESS JOB              */
//*                                                            */
//****************************************************************/
//*      Licensed Materials - Property of IBM                  */
//*                                                            */
//*      IMS QCF for z/OS Version 3 Release 2                  */
//*                                                            */
//*      5697-N50                                              */
//*                                                            */
//*      Copyright IBM Corp. 2000, 2014                        */
//*                                                            */
//*      All Rights Reserved.                                  */
//*      US Government Users Restricted Rights -               */
//*      Use, duplication or disclosure restricted             */
//*      by GSA ADP Schedule Contract with IBM Corp.           */
//*                                                            */
//****************************************************************/
//* NOTES =                                                    */
//* 1) REVIEW ALL STATEMENTS BEFORE SUBMITTING THIS JOB.       */
//* 2) CHANGE hlqual    FIELDS TO VALID HIGH LEVEL QUALIFIERS  */
//*    THAT COMPLY WITH YOUR SITES NAMING STANDARDS.           */
//*    CHANGE imshlq    FIELDS TO VALID HIGH LEVEL QUALIFIERS  */
//*    FOR IMS LOAD LIBRARY.                                   */
//*    CHANGE qcfhlq    FIELDS TO VALID HIGH LEVEL QUALIFIERS  */
//*    FOR QCF LOAD LIBRARY.                                   */
//* 3) CHANGE THE sysda FIELD TO A VALID LOCATION UNIT         */
//*    IDENTIFICATION THAT COMPLIES WITH YOUR SITES NAMING     */
//*    STANDARDS.                                              */
//*    CHANGE THE volser FIELD TO A VALID VOLSER NAME          */
//*    CHANGE &sysin &sysprint &imsdalib TO VALID DSNAMES      */
//*    CHANGE plexn TO THE REAL PLEX NAME                      */
//*    CHANGE imsid TO THE IMSID THAT IS RECOVERED             */
//* 4) DEFINE FOR VSAM DATASET SHOULD BE MODIFIED TO MEET YOUR */
//*    SYSTEM REQUIREMENTS. RECSZ PARMETER, THE AVERAGE RECORD */
//*    SIZE CAN BE SET TO THE LRECL OF THE IMS/ESA SHORT       */
//*    MESSAGE QUEUE AND THE MAX VALUE MUST BE EQUAL TO OR     */
//*    GREATER THAN IMS/ESA LONG MESSAGE QUEUE.                */
//* 5) WITH CHKPT LAST-  SET UP SYSPRINT,SYSIN,                */
//*    IMSDALIB, RECON DATA SETS.                              */
//*    DO NOT SPECIFY LOGIN DD.                                */
//*    CHECKPOINT AND LOGS ARE OBTAINED FROM RECON DATA SETS.  */
//*                                                            */
//*    CHKPT LAST,IMSID,FUNCTION,D=YYYY.DDD,T=HH:MM:SS         */
//*    CHKPT LAST,IMSID,FUNCTION,D=YYYY.DDD,T=HH:MM:SS,DT=CP   */
//*    CHKPT LAST,IMSID,FUNCTION,D=YYYY.DDD,T=HH:MM:SS,DT=MSG  */
//*         -IMSID - SET TO ID OF THE IMS SYSTEM               */
```

```
//*         -FUNCTION - SET TO RE FOR REPROCESS               */
//*         -D=YYYY.NNN - IS THE START DATE TIME FOR REPROCESS*/
//*         -T=HH:MM:SS - IS THE START TIME  FOR REPROCESS    */
//*         -DT=CP/MSG  - SELECT MESSAGES STARTING FROM A      */
//*                       CHECKPOINT (CP) OR BASED OM MSGS     */
//*                       CREATION TIME (MSG)                  */
//*                       DEFAULT IS DT=CP                     */
//*                                                           */


//* 6) WITHOUT CHKPT LAST-USE LOGIN DD TO SPECIFY THE LOGS TO  */
//*    BE USED. IN THIS CASE SYSPRINT, SYSIN, IMSDALIB        */
//*    AND RECON DATA SETS ARE NOT USED.                      */
//*                                                           */
//*                                                           */
//*  APAR    DATE    ID    DESCRIPTION                        */
//****************************************************************/
//IQCARPRS  JOB (&SYSUID),MSGCLASS=H,MSGLEVEL=0,CLASS=A,
//          TIME=1440,REGION=0M,
//          USER=&SYSUID,NOTIFY=&SYSUID
//********************************************************************
//*                                                                  *
//*          DELETE SCRAPPED RECORDS DATA SETS                       *
//*                                                                  *
//********************************************************************
//DELETE EXEC PGM=IEFBR14
//DD1    DD  DSN=hlqual.SCRAPSEL,
//           UNIT=SYSDA,DISP=(MOD,DELETE),SPACE=(TRK,(1))
//DD3    DD  DSN=hlqual.SCRAPLOG,
//           UNIT=SYSDA,DISP=(MOD,DELETE),SPACE=(TRK,(1))
//********************************************************************
//*                                                                  *
//*          DEFINE FOR TEMPORARY VSAM DATA SET TO RECREATE THE       *
//*          MESSAGE QUEUE - FOR RECOVERDM AND RECOVERAB              *
//*                                                                  *
//********************************************************************
//DFNDS    EXEC PGM=IDCAMS
//SYSPRINT DD    SYSOUT=*
//SYSIN    DD  *
    DEFINE CLUSTER (NAME(hlqual) -
                STORAGECLASS(PSTANDRD) -
                MANAGEMENTCLASS(PNOBACK) -
                IXD KEYS(4 0) -
                RECSZ(100 22524) -
                SPEED IMBED REPLICATE -
                REUSE SHAREOPTIONS(3 3) ) -
        DATA    (NAME(hlqual.DATA) -
                CONTROLINTERVALSIZE(24576) -
                CYL(100,20)) -
        INDEX   (NAME(hlqual.INDEX) -
                CONTROLINTERVALSIZE(512) -
                TRACKS(1,1))


//****************************************************************/
//*                                                           */
//*          IQCSELCT STEP                                    */
//*                                                           */
//****************************************************************/
//SELECT  EXEC PGM=IQCSELCT
//STEPLIB  DD  DISP=SHR,DSN=qcfhlq.SIQCLINK
//         DD  DISP=SHR,DSN=imshlq.RESLIB
//QCFIN    DD  *
*
* NEXT STATEMENTS WILL GET THE FIRST CHECKPOINT FROM LOG DATA SETS
* LOG DATA SETS ARE REQUIRED AND SET IN LOGIN DD
* - FOR REPROCESS, THE FIRST CHECKPOINT IS SELECTED.
*
CHKPT    00000000000
FUNCTION REPROCESS
*
*-* OR *-*
*
* NEXT STATEMENTS WILL GET THE DEFINED CHECKPOINT FROM LOG DATA SETS
* LOG DATA SETS ARE REQUIRED AND SET IN LOGIN DD
*
CHKPT D=YYYY.DDD,T=HH:MM:SS
FUNCTION REPROCESS
*
*-* OR *-*
* NEXT STATEMENTS WILL ALLOCATE RECON DATA SETS AND FIND THE
* LOG DATA SETS FROM RECON. THEN THE LOG DATA SETS WILL BE ALLOCATED AND
```

```
* THE LAST VALID CHECKPOINT WILL BE USED FOR RECOVER
* LOG DATA SETS ARE NOT REQUIRED (LOGIN DD..)
*
*-* OR *-*
CHKPT LAST,IMSID,RE,D=YYYY.DDD,T=HH:MM:SS
FUNCTION REPROCESS
*-* OR *-*
CHKPT LAST,IMSID,RE,D=YYYY.DDD,T=HH:MM:SS,DT=CP
FUNCTION REPROCESS
*-* OR *-*
CHKPT LAST,IMSID,RE,D=YYYY.DDD,T=HH:MM:SS,DT=MSG
```

```
FUNCTION REPROCESS
//*
//* -LOGIN DD IS OPTIONAL
//* -SPECIFY LOGIN WHEN EXECUTING THE PROCEDURE FOR CHECKPOINT OTHER
//*  THAN THE LAST CHECKPOINT.
//* -DO NOT SPECIFY LOGIN DD WHEN RUNNING WITH CHKPT LAST AND RECONS
//* THIS IS A SAMPLE OF MULTIPLE LOG TAPES WHERE EACH TAPE IS
//* ON A SEPERATE DD STATEMENT. EACH MUST START WITH FILE 2
//* AND SPECIFY BLP. THIS IS FOR LOG TAPES THAT ARE STANDARD
//* LABELED TAPES.
//*
//LOGIN    DD  DISP=OLD,UNIT=TAPE,LABEL=(2,BLP),DSN=LOG1,
//             DCB=(LRECL=22524,BLKSIZE=22528,RECFM=VB),
//             VOL=SER=IMSLG1
//         DD  DISP=OLD,UNIT=TAPE,LABEL=(2,BLP),DSN=LOG2,
//             DCB=(LRECL=22524,BLKSIZE=22528,RECFM=VB),
//             VOL=SER=IMSLG2
//*
//* SYSPRINT AND SYSIN USED FOR DBRC COMMAND INPUT AND OUTPUT
//* IF NOT PROVIDED THEY ARE DYNAMICCALY ALLOCATED.
//*
//SYSPRINT DD DSN=&sysprint,
//         UNIT=sysda,VOL=SER=volser,
//         DISP=(NEW,KEEP,CATLG),
//         SPACE=(CYL,(1,1))
//SYSIN    DD DSN=&sysin,UNIT=sysda,
//         DISP=SHR,
//         SPACE=(TRK,(1,1))
//*
//* IMSDALIB DD - USED FOR DYNAMICALLY ALLOCATION OF RECON DATA SETS
//*
//IMSDALIB DD  DISP=SHR,DSN=&imsdalib
//*
//* IF NOT PRPOVIDED ,RECON1 AND RECON2 ARE DYNAMICALLY ALLOCATED
//*
//RECON1   DD  DISP=SHR,DSN=hlqual.RECON1
//RECON2   DD  DISP=SHR,DSN=hlqual.RECON2
//RECON3   DD  DISP=SHR,DSN=hlqual.RECON3
//*
//MSGQ     DD DSN=hlqual,
//           AMP=('BUFND=20,BUFNI=10'),
//           DISP=(OLD,KEEP,KEEP)
//*
//SCRAPSEL DD  DSN=hlqual.SCRAPSEL,UNIT=sysda,DISP=(NEW,CATLG),
//             DCB=(LRECL=22844,BLKSIZE=22848,RECFM=VB),
//             SPACE=(CYL,(5,5))
//*
//*   THE LRECL OF THE LOGOUT DATA SET MUST BE AT LEAST 320
//*   BYTES LONGER THEN THE LRECL OF THE LOGIN DATA SETS,
//*   TO ACCOMMODATE MRPREFIX.
//*
//LOGOUT   DD  DSN=&&LOGOUT,UNIT=sysda,SPACE=(CYL,(5,5)),
//             DCB=(LRECL=22844,BLKSIZE=22848,RECFM=VB),
//             DISP=(NEW,PASS)
//SORTINB  DD  DSN=&&SORTINB,UNIT=sysda,SPACE=(TRK,(1,1)),
//             DCB=(LRECL=80,BLKSIZE=800,RECFM=FB),
//             DISP=(NEW,PASS)
//QCFPRINT DD  SYSOUT=*
//SNAPDUMP DD  SYSOUT=*
//SYSOUT   DD  SYSOUT=*
//*
```

```
//*************************************************************/
//*                                                           */
//*            DELETE OF VSAM DATASET                         */
//*                                                           */
//*************************************************************/
//DLTDS    EXEC PGM=IDCAMS
```

```
//SYSPRINT DD    SYSOUT=*
//SYSIN    DD  *
    DELETE (hlqual)
//****************************************************************/
//*                                                              */
//*            SORTB JOB STEP                                    */
//*                                                              */
//****************************************************************/
//SORTB   EXEC PGM=SORT,PARM='CORE=MAX',
//             COND=((0,NE,SELECT))
//SORTLIB  DD  DISP=SHR,DSN=SYS1.SORTLIB
//SORTIN   DD  DISP=(OLD,DELETE),DSN=*.SELECT.LOGOUT
//SORTOUT  DD  DSN=hlquql.SORTOUTB,UNIT=sysda,SPACE=(CYL,(5,5)),
//             DCB=(LRECL=22844,BLKSIZE=22848,RECFM=VB),
//             DISP=(NEW,CATLG)
//SYSOUT   DD  SYSOUT=*
//SORTWK01 DD  UNIT=sysda,SPACE=(CYL,(20,10),,CONTIG)
//SORTWK02 DD  UNIT=sysda,SPACE=(CYL,(20,10),,CONTIG)
//SORTWK03 DD  UNIT=sysda,SPACE=(CYL,(20,10),,CONTIG)
//SORTWK04 DD  UNIT=sysda,SPACE=(CYL,(20,10),,CONTIG)
//SYSIN    DD  DSN=&&SORTINB,DISP=(OLD,PASS)
//*
//****************************************************************/
//*                                                              */
//*            IQCLOAD  JOB STEP                                 */
//*                                                              */
//****************************************************************/
//IQCLOAD  EXEC PGM=IQCINI0$,
//    PARM='BPEINIT=IQCBINI0,BPECFG=IQCBACNF,IMSPLEX=plexn,QCFIMS=imsid'
//STEPLIB  DD  DISP=SHR,DSN=qcfhlq.SIQCLINK
//         DD  DISP=SHR,DSN=imshlq.RESLIB
//PROCLIB  DD  DISP=SHR,DSN=hlqual.PROCLIB
//LOAD     DD  DISP=SHR,DSN=hlqual.SORTOUTB
//QCFPRINT DD  SYSOUT=*
//SNAPDUMP DD  SYSOUT=*
//* SCRAPLOG IS A REQUIRED DD STATEMENT. IT WILL CONTAIN THE
//* REJECTED MESSAGE FROM THE IQCLOAD  RUN. SPACE AND DCB
//* PARAMETERS SHOULD BE THE SAME AS LOGOUT'S.
//*
//SCRAPLOG DD  DSN=hlqual.SCRAPLOG,UNIT=sysda,DISP=(NEW,CATLG),
//             DCB=(LRECL=22844,BLKSIZE=22848,RECFM=VB),
//             SPACE=(CYL,(5,5))
//QCFIN    DD  *
TITLE='QCF LOAD'
FU LO
END
/*
//
```

Batch processing with skeletal JCL

You can use the batch logic to create REPROCESS jobs from skeletal JCL.

Store your skeletal JCL in the QCFJCLIN DD library using member names IMSID*xx*RE. The REPROCESS jobs that are created are stored in the QCFJCLOT DD library using the same names as those that were used for the input skeletal JCL.

IMS Queue Control Facility batch logic performs the following processing:

- Analyzes run parameters, allocates all necessary data sets, and prepares a LIST.LOG run command for DBRC.
- Attaches the RECON utility program and analyzes the output to find the last valid checkpoint for the function type, and to find all PRISLDs with log records for the function type.

  For REPROCESS, the checkpoint is selected using date and time (D= and T=). The log data sets consist of all PRISLDs that contain the selected checkpoints, plus all of the following PRISLDs.

- Reads the skeletal JCL for the function type and the IMSID from the skeletal JCL library, replaces, or creates the LOGIN DD concatenation statement if that statement is missing, with all of the selected PRISLDs data sets in the SELECT step.

  And replaces, or creates the CHKPT command statement if the CHKPT command statement is missing for the last valid checkpoint in the SELECT step.

- Stores the new JCL in the output JCL library using the name of the input skeletal JCL.

Skeleton to create REPROCESS JCL

You can create a REPROCESS procedure by running the IQCABAT0 skeletal JCL.

A sample skeleton to create QCF procedures running batch processing logic is delivered as part of the IMS Queue Control Facility package in *hlq*.SIQCSAMP.

IQCA##RE
  Sample skeleton for the REPROCESS procedure

Use the sample skeleton to create a JCL skeleton for every IMS system that requires REPROCESS.

Store your skeletal JCL in the QCFJCLIN library using member names IMSID*xxfunction_type*. The REPROCESS job that is created during the batch run are stored in the QCFJCLOT library using the same names as those that were used for the input skeletal JCL.

The batch processing logic performs the following actions:

- Analyzes run parameters, allocates all necessary data sets, and prepares a LIST.LOG run command for DBRC.
- Attaches the RECON utility program (DSPURX00) and analyzes the output to locate the last valid checkpoint for the function type and to find all PRISLDs with log records for the function type.
- Reads the skeletal JCL for the function type and the IMSID from the skeletal JCL library, replaces or creates (if missing) the LOGIN DD concatenation with all selected PRISLDs data sets in the SELECT step, and replaces or creates (if missing) the CHKPT command statement for the last valid checkpoint in the SELECT step.
- Stores the new JCL in the output JCL library using the name of the input skeletal JCL.

Prepare REPROCESS JCL from skeleton

Run the batch processing logic IQCABAT0 to create REPROCESS jobs from skeletal JCL.

You can use the batch processing logic to create REPROCESS jobs from skeletal JCL. Store your skeletal JCL in the QCFJCLIN library using member names IMSID*xxfunction_type*.

The REPROCESS job that is created is stored in the QCFJCLOT library using the same name as that for the input skeletal JCL.

The batch processing logic performs the following actions:

- Analyzes run parameters, allocates all necessary data sets, and prepares a LIST.LOG run command for DBRC.
- Attaches the RECON utility program (DSPURX00) and analyzes the output to locate the last valid checkpoint for the function type and to find all PRISLDs with log records for the function type:

  - For REPROCESS, the checkpoint is selected using date and time (D= and T=). The log data sets consist of all PRISLDs that contain the selected checkpoints, plus all following PRISLDs.

- Reads the skeletal JCL for the function type and the IMSID from the skeletal JCL library, replaces or creates (if missing) the LOGIN DD concatenation with all selected PRISLDs data sets in the SELECT step, and replaces or creates (if missing) the CHKPT command statement for the last valid checkpoint in the SELECT step.
- Stores the new JCL in the output JCL library using the name of the input skeletal JCL. Member IQCARUN in hlqual.SIQCSAMP contains the sample JCL for executing batch processing logic.

Example JCL for REPROCESS

The following example shows the JCL for running batch processing logic:

```
//****************************************************************/
//* JOB NAME = IQCARUN                                          */
```

```
//*                                                        */
//* DESCRIPTIVE NAME = SAMPLE EXECUTION JCL FOR            */
//*                    THE IMS/ESA QUEUE CONTROL FACILITY  */
//*                                                        */
//* STATUS = VERSION 03  RELEASE 02  MODIFICATION LEVEL 00 */
//*                                                        */
//* FUNCTION = EXECUTE A JOB TO PREPARE RECOVERAB/RECOVERDM */
//* PROCEDURES USING RECON LISTING AND JCL SKELETONS        */
//*                                                        */
//* SAMPLES FOR JCL SKELETONS ARE IN hlqual.SIQCSAMP:       */
//*                          IQCA##AB                        */
//*                          IQCA##DM                        */
//*                          IQCA##RE                        */
//* SAMPLES SHOULD BE CUSTOMIZED FOR IMSs AND STORED IN PDS  */
//* (QCFJCLIN DD) WITH NAMES:                                */
//*                          imsid##AB                        */
//*                          imsid##DM                        */
//*                          imsid##RE                        */
//* JOBS WILL BE CREATED IN PDS (QCFJCLOT DD)                */
//*                                                         */
//****************************************************************/
//*        Licensed Materials - Property of IBM              */
//*                                                         */
//*        IMS QCF for z/OS Version 3 Release 2             */
//*                                                         */
//*        5697-N50                                         */
//*                                                         */
//*        Copyright IBM Corp. 2000, 2014                   */
//*                                                         */
//*        All Rights Reserved.                             */
//*        US Government Users Restricted Rights -          */
//*        Use, duplication or disclosure restricted        */
//*        by GSA ADP Schedule Contract with IBM Corp.      */
//*                                                         */
//****************************************************************/
//*                                                         */
//IQCARUN   JOB (ACCOUNTINFO),NORMAL JOBCARD INFO
//***************************************************************
//*     EXEC IQCABAT0                                        *
//*    &prm1 - IMSID -  character, up to 4bytes              *
//*    &prm2 - FUNCTION - character ,2 bytes, AB/DM/RE       *
//*    &prm3 - START DATE/TIME D=YYYY.DDD,T=HH:MM:SS         *
//*    &prm4 - TYPE OF START DT=CP/DT=MSG                    *
//* for FUNCTIONS AB and DM-&prm3 is optional               *
//* for FUNCTIONS AB and DM-&prm4 is ignored                *
//* it can be used to reduce the LIST.LOG output            *
//* &prm3 should be earlier than the last PRISLD start time *
//*   if present -RECON LIST.LOG  works with FROMDATE=&prm3  *
//*   if missing -RECON LIST.LOG  works without FROMTIME     *
//* for FUNCTION RE -&prm3 is required                       *
//*   it is the start checkpoint date time for reprocess     *
//*   date and time are the local date and time of the       *
//*     required check point or insert time of messages      *
//* for FUNCTION RE -&prm4 is optional - defaults to DT=CP   *
//*   it is the type of starting date/time:                  *
//*      DT=CP - start date / time is used to find checlpoint *
//*      DT=MSG- start date / time is used to find first messg*
//*    RECON LIST.LOGS works without FROMTIME                *
//***************************************************************

//STEP1    EXEC PGM=IQCABAT0,PARM='&prm1,&prm2,&prm3,&prm4'
//STEPLIB  DD DISP=SHR,DSN=hlqual.SIQCLINK
//         DD DISP=SHR,DSN=IMSREL.RESLIB
//         DD DISP=SHR,DSN=SYS1.LINKLIB
//***************************************************************
//* IMSDALIB DD is optional-used for RECON1 and RECON2       *
//* allocation.                                              *
//* RECON1 and RECON2 are allocated using information        *
//* from  sources in next order:                             *
//*    - DD information in JCL (RECON1 and RECON2 DD)         *
//*    - IMSDALIB (RECON1 and RECON1 members)                *
//*    - STEPLIB  (RECON1 and RECON2 members)                *
//***************************************************************
//IMSDALIB DD DISP=SHR,DSN=imsdalib
//*
//SYSUDUMP DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//*
//***************************************************************
//* SYSIN    DD is optional                                  *
```

```
//* if not provided, SYSIN    is dynamically allocated        *
//* SYSIN is the input control cards file for DSPURX00         *
//******************************************************************
//SYSIN    DD DSN=sysin,UNIT=SYSDA,
//         DISP=SHR,
//         SPACE=(TRK,(1,1))
//*
//******************************************************************
//* SYSPRINT DD is optional                                     *
//* if not provided, SYSPRINT is dynamically allocated          *
//* SYSPRINT is the output file from LIST.LOG command           *
//******************************************************************
//SYSPRINT DD DSN=sysprint,UNIT=SYSDA,
//         DISP=SHR,
//         SPACE=(CYL,(10,10))
//*
//******************************************************************
//* RECON1 and RECON2 DD are optional                           *
//* if not provided, they are dynamically allocated             *
//* using information in IMSDALIB and STEPLIB                    *
//******************************************************************
//RECON1   DD DISP=SHR,DSN=recon1
//RECON2   DD DISP=SHR,DSN=recon2
//*
//******************************************************************
//* QCFJCLIN is PO data set with JCL skeletons                   *
//* for IMSIDs and FUNCTIONS (AB/DM)                             *
//******************************************************************
//QCFJCLIN DD DSN=qcfjclin,DISP=SHR
//*
//******************************************************************
//* QCFJCLOT is PO data set to store output JCL                  *
//* for IMSIDs and FUNCTIONS (AB/DM)                             *
//******************************************************************
//QCFJCLOT DD DSN=qcfjclot,DISP=SHR
//*
//
```

EXEC statement for IQCABAT0

The EXEC statement for IQCABAT0 tells the system which job to run, the IMSID, function type, and start date and time.

The EXEC statement must specify PGM=IQCABAT0.

The parameters in the EXEC statement specify the following parameters:

- IMSID
- Function type (RE)
- Start date and time, based on the function type:

  For REPROCESS, the start date and time is required and indicates the starting checkpoint date and time. The LIST.LOG command starts with date and time, minus 2 days.

DD statements for IQCABAT0

The DD statements in IQCABAT0 for REPROCESS specify the data sets that are required for IQCABAT0 for REPROCESS JCL.

The STEPLIB DD statement needs to contain the following parameters:

- *hlq*.SIQCLINK
- IMS RESLIB
- SYS1.LINKLIB

The IMSDALIB DD statement is used to dynamically allocate RECON data sets.

The SYSIN DD statement is the standard input data set for DBRC commands. If this statement is not provided, SYSIN is dynamically allocated.

The SYSPRINT DD statement is the standard output data set for DBRC commands. If this statement is not provided, SYSPRINT is dynamically allocated.

The RECON1 and RECON2 DD statements need to be provided. If these statements are not provided, RECON1 and RECON2 are dynamically allocated.

The QCFJCLIN DD library statement contains the skeletal JCL for REPROCESS function for each IMSID.

The QCFJCLOT DD library statement is used to store the new REPROCESS JCL.

Using batch processing logic to prepare REPROCESS JCL for IMSID IMRE

- Skeletal JCL is created in the QCFJCLIN data set with the name IMA8*xx*RE. Run the IQCARUN routine and the parameters in the EXEC statement are PARM='IMA8,RE' (D=and T= are optional.)
- Member IMA8*xx*RE in the QCFJCLOT data set contain the prepared LOGIN DD and CHKPT statements.

IQCAxxRE
    Member IQCAxxRE in *hlq*.SIQCSAMP contains the sample skeletal JCL for function REPROCESS.

Chapter 17. Unloading the message queues

The UNLOAD function unloads messages from the IMS message queues in both shared- and nonshared-queues environments.

Topics:

UNLOAD function description

The UNLOAD function unloads (or deletes) messages from the IMS message queues, or removes messages for requeuing, according to your specifications.

You can use the UNLOAD function to perform the following tasks:

- To clean up the queues by removing unnecessary messages
- To remove messages for requeuing
- To perform stress testing

When you use the UNLOAD function as a migration or fallback aid, the unload data set is used to requeue messages across supported IMS releases. Messages that are created on one supported release of IMS can be inserted into another supported release of IMS if the involved source and destination resources, such as LTERMs, transactions, MSC names, and other destination resources are defined on both systems.

UNLOAD produces reports on the messages that its logic unloads. You can specify an age limit that these messages must meet in order to be included in the unload reports.

UNLOAD logic selects records from the message queues for possible later insertion by the load function logic or for purely informational purposes. UNLOAD logic performs the following tasks:

1. Processes control blocks that were built by IMS Queue Control Facility control processing logic from input control statements to make the following determinations:

   - Determines queue selection from the QUEUETYPE= keyword of the FUNCTION statement
   - Determines record selection criteria for date and time stamps, and record type from the SELECT statement
   - Determines record selection criteria for message sources and destinations from the INCLUDE and EXCLUDE statements

2. Discards records that fail to meet the selection criteria of the SELECT, INCLUDE, and EXCLUDE statements

3. Writes records that do meet the selection criteria of the FUNCTION, SELECT, and INCLUDE statements to the unload data set

4. Produces reports about the number and types of records that UNLOAD unloaded from the message queues

5. Returns control to IMS Queue Control Facility control processing logic

The job steps that are required for the UNLOAD function are as follows: UNLOAD

UNLOAD function restrictions and limitations

The UNLOAD function does not apply to certain message types and under certain conditions.

- AOI response type messages cannot be unloaded from the AOI queue.

  The AOI response type messages are removed from IMS queue when the AOI application program completes processing the response (either reaches a SYNC point, or is terminated).

- Messages cannot be unloaded from an IMS queue (such as DEST1) when a queue message is currently being read and processed by another IMS function.

  For example, if IMS OTMA has read a message from one of the OTMA queues (such as DEST1) and is processing that message, then the IMS Queue Control Facility UNLOAD function receives a status code that indicates that the current queue is being processed by another IMS function.

  Because IMS is currently removing messages from this queue (DEST1), IMS Queue Control Facility bypasses the queue and does not attempt to unload any of the messages for this queue.

  If a second unload attempt is made and the OTMA queue (DEST1) is not being processed, the IMS Queue Control Facility UNLOAD function performs the unload of this queue.

- Messages cannot be unloaded from an IMS shared queue for the following two conditions:

  - When a shared queue message is being read and processed by another function in the current IMS system that is performing the IMS Queue Control Facility UNLOAD

    For example, if IMS Queue Control Facility (running on an IMS1 system) is attempting to read a message from the IMS shared queue, and another function within this IMS1 system is already processing this shared queue, then the IMS Queue Control Facility UNLOAD function receives a status code that indicates that another function within IMS1 is currently processing this shared queue.

    Because another function in IMS1 is currently removing messages from this queue, IMS Queue Control Facility bypasses the processing of this queue.

  - When a shared queue message is being read and processed by another IMS system.

    For example, if IMS Queue Control Facility (running on an IMS1 system) is attempting to read a message from the IMS shared queue, and IMS2 is currently removing messages from this same shared queue, then the IMS Queue Control Facility UNLOAD function receives a status code that indicates that another IMS (in this example IMS2) is currently processing this shared queue.

    Because IMS2 is currently removing messages from this queue, IMS Queue Control Facility bypasses the processing of this queue.

  If a second unload attempt is made and the shared queue is not being processed by another IMS system, the IMS Queue Control Facility UNLOAD function performs the unload of this queue.

Unload selection criteria and processing

The UNLOAD function selects and removes messages from the message queues based on your selection criteria.

Use the UNLOAD function to create an UNLOAD data set that contains the message records that were selected from the message queues. This UNLOAD data set provides the messages that can eventually be read into the message queues by the load function logic.

Selection parameters

Use the UNLOAD function to select and remove messages from the message queues based on your selection criteria. You specify the selection criteria in control statements that are read by the IMS Queue Control Facility control processing logic.

Selection criteria

Specify selection criteria to the UNLOAD function with FUNCTION, SELECT, INCLUDE, and EXCLUDE statements. Use these statements to define a subset of the message records to be selected from the message queues.

The SELECT, INCLUDE, and EXCLUDE statements are optional. If you do not specify any of these three statements, all of the records from all of the message queues are selected.

You can have message records selected based on the following criteria:

As specified in FUNCTION statement:

- Message age
- Queue type
    - ALL
    - APPC
    - DEADQ
    - LONG
    - LTERM
    - OTMA
    - REMOTE (remote MSNAME queue)
    - SHORT
    - SERIAL (serial transaction queue)
    - SUSPEND (suspend transaction queue)
    - TRANSACTION (transaction ready queue (local and remote))
- SBMP*xx*

As specified in SELECT statement:

Beginning date and time
Conversational messages or no conversational messages
Ending date and time
Log record type
System messages or no system messages

As specified in INCLUDE or EXCLUDE statement:

Destination or source type (or both) of a message
Destination with optional system ID
Logical unit name
Logical unit name destination
Logical unit sidename
Logical unit tpname
OTMA tmember name
OTMA tmember name destination
OTMA tmember name source
OTMA tpipe name
Source with optional system ID
Specific IMS ID
Specific IMS ID destination
Specific IMS ID source
Either the originating unit-of-work, or the processing IMS ID, or both
Either the originating unit-of-work, or the processing token, or both
VTAM name

Use the FUNCTION statement to specify one or more message queues that are to be processed and, optionally, the age of messages to be processed.

Use the TOIPARMS statement to support communication between IMS Queue Control Facility and IMS system. TOIPARMS identifies the XCF group that QCF is to join and whether to issue IMS **/ST0** commands against active destinations. PROCLIB member, FOI*imsid*P is required for the IMS system to identify the group that the IMS system is to join.

Within the group, which should be the same for IMS Queue Control Facility and IMS, IMS Queue Control Facility can send commands to IMS to stop destinations.

Use the SELECT statement to specify global criteria, a range of times, and the types of records that are to be selected by the UNLOAD logic. You can also use the SELECT statement to specify whether the UNLOAD data set will be created using the READ or REPORTONLY options of the ACTION= keyword. However, unlike using BROWSE and RECOVER, if you use READ or REPORTONLY with the UNLOAD job, messages will be unloaded off the queues and there will be no backup of the unloaded messages. You can specify only one SELECT statement.

Use the INCLUDE and EXCLUDE statements to specify selection criteria that is specific to message sources and destinations. You can specify as many INCLUDE and EXCLUDE statements as are necessary for your purposes.

If any INCLUDE statements are present, a record must meet all of the criteria for at least one INCLUDE statement to be included in the unload logic selections. Unspecified criteria are considered to be met.

A record that meets all of the criteria that is specified by an EXCLUDE statement will be excluded from the unload logic selections even if the record meets all criteria that is specified by the SELECT statement and an INCLUDE statement. Unspecified criteria are considered to be met.

Selection processing

When given control by IMS Queue Control Facility control processing logic, the UNLOAD function determines whether an UNLOAD data set should be created by determining if `ACTION=(READ)` was specified, or defaulted to, on the FUNCTION or SELECT statement. If the UNLOAD data set was to be created but is not present, the UNLOAD function issues a message and continues as if `ACTION=(REPORTONLY)` were specified. However, unlike using BROWSE and RECOVER, if you use READ or REPORTONLY with the UNLOAD job, messages will be unloaded off the queues and there will be no backup of the unloaded messages.

The UNLOAD function issues a call to IMS requesting that a GETUNIQUE function be performed on the specific message queues that you specified.

The UNLOAD function determines if the record meets the selection criteria that is specified by any FUNCTION, SELECT, and INCLUDE statements. If the record does not meet these selection criteria, the record is discarded. If a record does meet these selection criteria, the output logic determines if the record meets the criteria for any EXCLUDE statement. If the record meets the criteria for an EXCLUDE statement, the record is discarded. The UNLOAD function logic appends a prefix to each message record before writing each message record to the UNLOAD data set.

When end-of-data is signaled by IMS, the UNLOAD function produces reports about the number and type of messages that the UNLOAD function found on the message queues, and about the statistics on the number and types of records that were processed.

The UNLOAD function then returns control to IMS Queue Control Facility control processing logic for clean up and normal termination.

Control statements for UNLOAD function

Use the UNLOAD function to remove selected messages from the queues and to make a copy of them in the IMS Queue Control Facility UNLOAD data set.

The UNLOAD function lets you remove selected messages from the queues and make a copy of them in the IMS Queue Control Facility UNLOAD data set.

The control statements that are recognized by the UNLOAD function are:

    * comment
    EXCLUDE
    FUNCTION
    INCLUDE
    SELECT
    TITLE
    TOIPARMS

UNLOAD control statement abbreviations

The following table shows a list of UNLOAD keyword abbreviations that are allowed by the IMS Queue Control Facility control processing interpreter:

Table 18. IMS QCF control processing UNLOAD keyword abbreviations

Keyword	Abbreviation
EXCLUDE	EX
FUNCTION	FU
INCLUDE	IN
SELECT	SE

FUNCTION UNLOAD statement syntax

The FUNCTION UNLOAD statement must follow certain syntactic specifications.

The syntax of the UNLOAD FUNCTION statement is:

```
►►─ FUNCTION ── UNLOAD ──────────────────────────────────────────────►
                        └─ ACTION=( ──┬── READ ────── ),──┘
                                      └── REPORTONLY ──┘

    ►──────────────────────────────────────────────────────►
       └─ AGE= nnnnnnnn , ──┬── AGEUNITS=DAYS, ──┬──┘
                            └── AGEUNITS=SECONDS, ──┘

    ►──────────────────────────────────────────────────────►
                           ┌─ , choose one or more ◄─┐
       └─ AREA=( ──┬─────── ALL ───────┬── ) ──┘
                   ├── GLOBAL ──┤
                   ├── LOCAL ───┤
                   ├── OVERFLOW ─┤
                   └── QSN ─────┘

    ►──────────────────────────────────────────────────────►◄
                           ┌─ , choose one or more ◄─┐        └─ SBMPxx, ¹ ─┘
       └─ QUEUETYPE=( ──┬── ALL ───┬── ) ──┘
                        ├── APPC ──┤
                        ├── LONG ──┤
                        ├── LTERM ─┤
                        ├── DEADQ ─┤
                        ├── OTMA ──┤
                        ├── REMOTE ┤
                        ├── SHORT ─┤
                        ├── SERIAL ┤
                        ├── SUSPEND ┤
                        └── TRANSACTION ┘
```
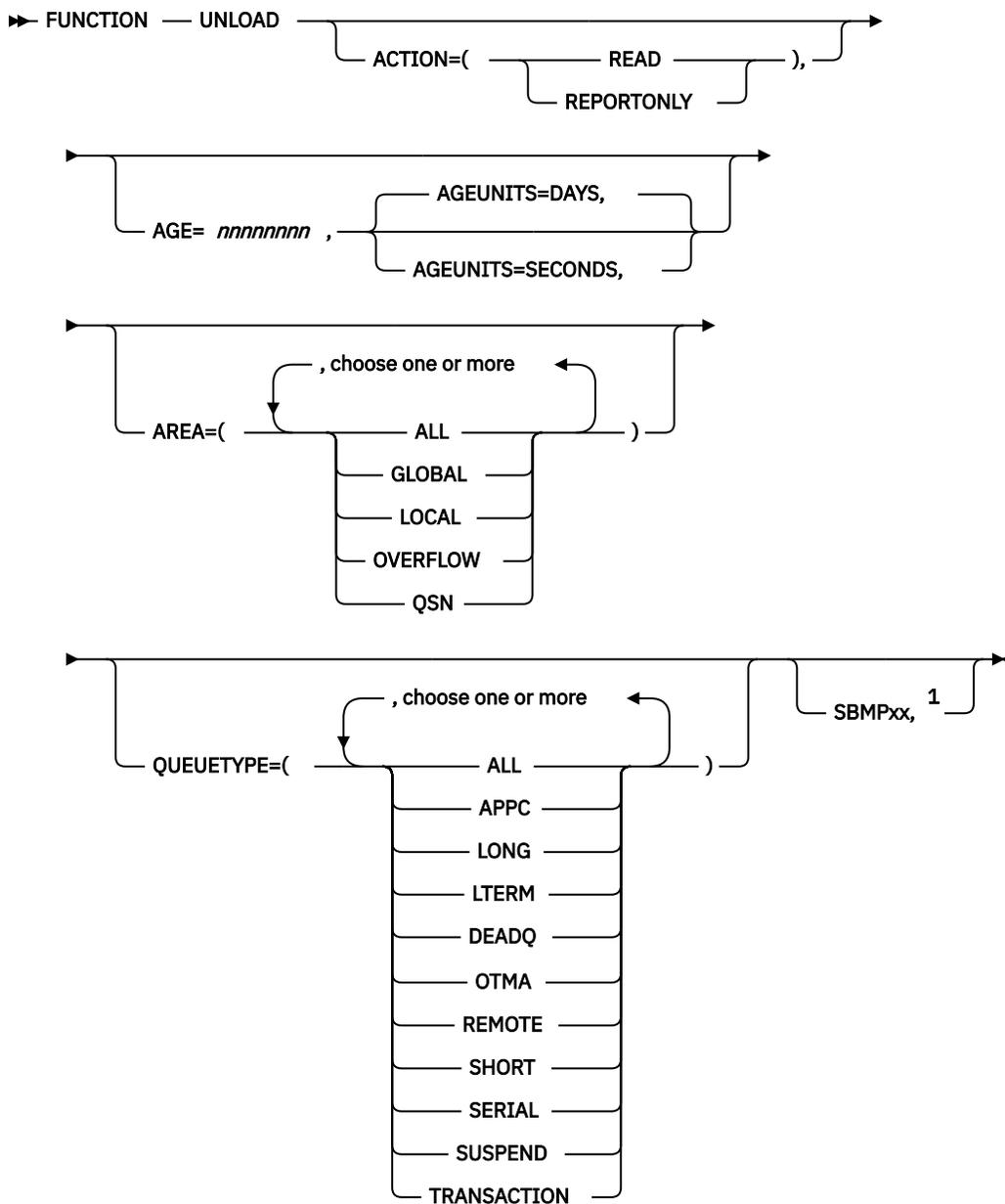
Notes:

[1] Required for all jobs that are started because a threshold was crossed.

FUNCTION UNLOAD INCLUDE statement syntax

The FUNCTION UNLOAD INCLUDE statement must follow certain syntactic specifications.

In the JCL, the INCLUDE control statement is written on a separate line after the FUNCTION UNLOAD control statement.

JCL example 1

```
FUNCTION UNLOAD QUEUETYPE=(ALL)
INCLUDE DESTINATION=A*
END
```

JCL example 2

```
FUNCTION UNLOAD AREA=(ALL),QUEUETYPE=(ALL)
INCLUDE DESTINATION=A*,
IMSIDS=IMS1
END
```

Syntax diagram

Other options

```
                         ◄─── , choose one or more ───◄
►►─┬─────────────────── DESTINATION= destination (sysid) ──────────────┬─►◄
   │                    ┌─── , choose one or more ───◄                   │
   ├─ DESTYPE=( ─┬───┬─ APPC ──┬───┬─ ) ──────────────────────────────┤
   │             │   ├─ LTERM ─┤                                        │
   │             │   ├─ LTRAN ─┤                                        │
   │             │   ├─ MSC ───┤                                        │
   │             │   ├─ OTMA ──┤                                        │
   │             │   ├─ RTRAN ─┤                                        │
   │             │   ├─ TRANS ─┤                                        │
   │             │   └─ VSP ───┘                                        │
   ├────────── IMSDSTID= destinationimsid ──────────────────────────────┤
   ├────────── IMSIDS= imsid ────────────────────────────────────────────┤
   ├────────── IMSSRCID= sourceimsid ───────────────────────────────────┤
   ├────────── LUNAME= luname ──────────────────────────────────────────┤
   ├────────── LUNAMEDST= lunamedestination ────────────────────────────┤
   ├────────── LUNAMESRC= lunamesource ─────────────────────────────────┤
   ├────────── LUSIDENAME= lusidename ──────────────────────────────────┤
   ├────────── LUTPNAME= lutpname ──────────────────────────────────────┤
   │                    ┌─── , choose one or more ───◄                   │
   ├─ MSGTYPE=( ─┬───┬─ APPC ──┬───┬─ ) ─────────────────────────────────┤
   │             │   ├─ MSC ───┤                                        │
   │             │   ├─ OTMA ──┤                                        │
   │             │   └─ VSP ───┘                                        │
   ├────────── PERCENTDST= percentdestination ──────────────────────────┤
   ├────────── RECCOUNTDST= reccountdestination ────────────────────────┤
   ├────────── SOURCE= sourcename (sysid) ──────────────────────────────┤
   │                    ┌─── , choose one or more ───◄                   │
   ├─ SRCTYPE=( ─┬───┬─ APPC ──┬───┬─ ) ─────────────────────────────────┤
   │             │   ├─ LTERM ─┤                                        │
   │             │   ├─ MSC ───┤                                        │
   │             │   ├─ OTMA ──┤                                        │
   │             │   └─ VSP ───┘                                        │
   ├────────── TMEMBER= tmembername ────────────────────────────────────┤
   ├────────── TMEMBERDST= tmemberdestination ──────────────────────────┤
   ├────────── TMEMBERSRC= tmembersource ───────────────────────────────┤
   ├────────── TPIPE= tpipename ────────────────────────────────────────┤
   ├────────── UOWORGID= originatingimsid ──────────────────────────────┤
   ├────────── UOWORGTK= originatingtoken ──────────────────────────────┤
   ├────────── UOWPROID= processingimsid ───────────────────────────────┤
   ├────────── UOWPROTK= processingtoken ───────────────────────────────┤
   └────────── VTAM= vtamname ──────────────────────────────────────────┘
```

FUNCTION UNLOAD SELECT statement syntax

The FUNCTION UNLOAD SELECT statement must follow certain syntactic specifications.

In the JCL, the SELECT control statement is written on a separate line after the FUNCTION UNLOAD control statement.

JCL example 1

```
FUNCTION UNLOAD QUEUETYPE=(ALL)
SELECT NOCONVMSG
END
```

JCL example 2

```
FUNCTION UNLOAD AREA=(ALL),QUEUETYPE=(ALL)
SELECT NOCONVMSG,
NOSYSMSG
END
```

The syntax of the FUNCTION UNLOAD SELECT statement is:

Syntax diagram
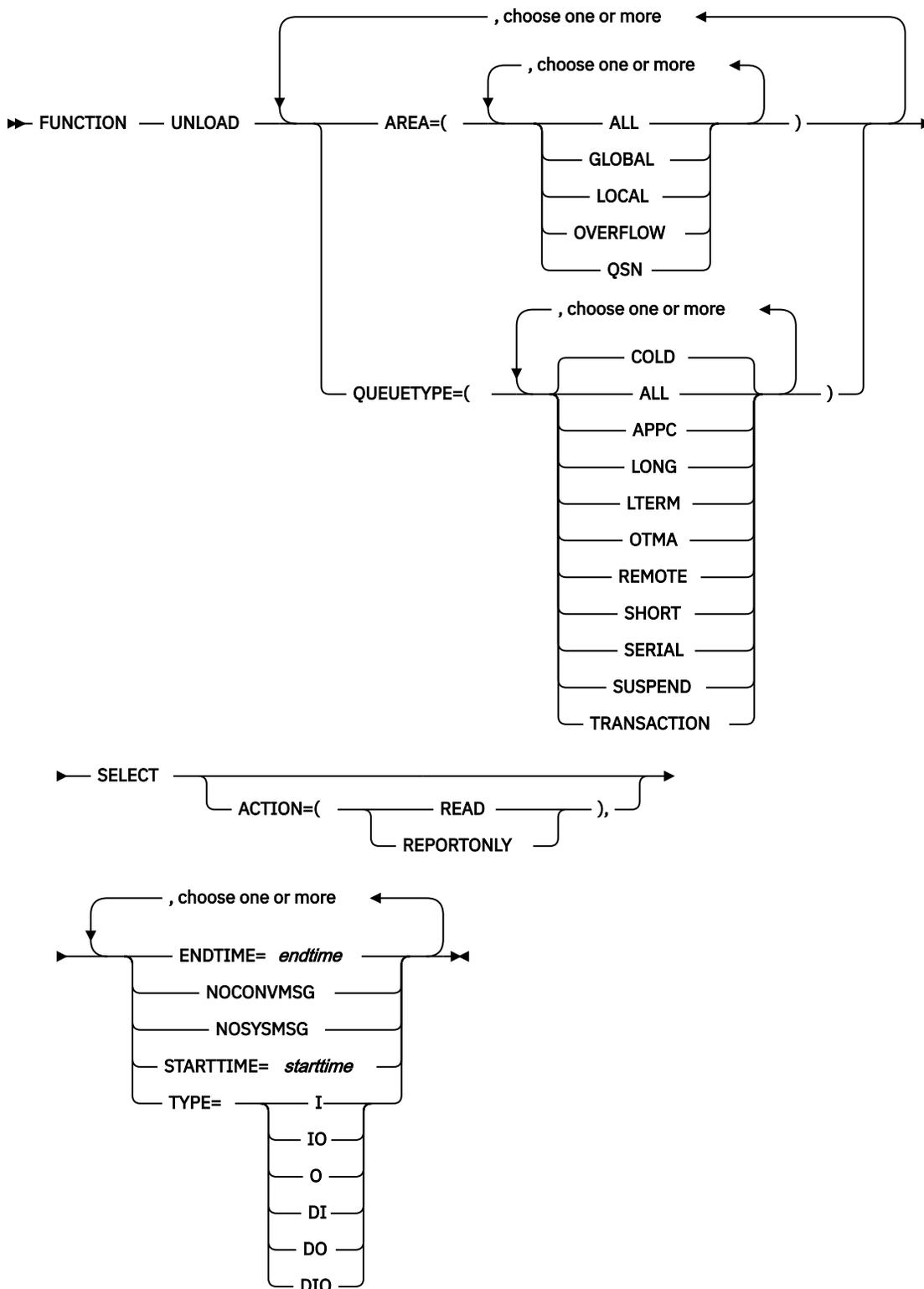
TOIPARMS statement for UNLOAD

Use the TOIPARMS to cause QCF to handle the A068 return code in a special way.

The TOIPARMS statement is optional and must follow certain syntactic specifications.

The syntax of the TOIPARMS statement is:

```
►►─ TOIPARMS ── XCFGROUP=TOI  xxxxx ─┬─────────────────┬─►◄
                                     ├─ ,ENABLESTOP  ──┤
                                     └─ ,DISABLESTOP ──┘
```
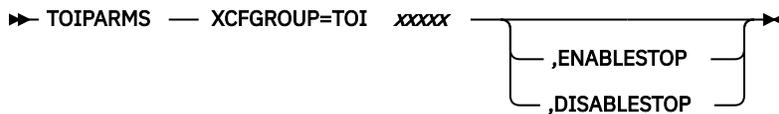
where:

xxxxx
> Specifies the XCF group name that IMS Tools Online System Interface joins to interface with client programs. This name is also specified in the XCFGROUP= keyword parameter in the IMS Tools Online System Interface PROCLIB member. For details, see the topic "PROCLIB member keyword parameters" in *Tools Base for z/OS Configuration Guide for IMS*.

One of the main problems that queue overflow protection encountered was the AIBREASN=A068 (UNLOAD LOCAL QUEUES, REQUESTED DESTINATION IS BEING READ BY ANOTHER TASK) errors that were returned on the UNLOAD that prevented QCF from successfully unloading a build-up of committed messages.

This build-up of committed messages were typically inserted from an application program, usually a BMP. The destination could be another transaction, LTERM or LU62 client. In all cases the UNLOAD function was activated successfully to locate the destination and unload the messages. But, because the destination's ITASK was also trying to process the messages, error A068 was returned to IMS Queue Control Facility. IMS Queue Control Facility skipped the busy destination and unloaded the next destinations. In most cases this skipped destination was the destination that was causing the message build-up.

IMS Queue Control Facility now communicates with IMS and issues commands to stop the destination from processing, so that IMS Queue Control Facility can unload the destination. The communication between IMS and IMS Queue Control Facility is on the XCF level. IMS Tools Online System Interface, which is part of IMS Queue Control Facility package, supports this communication.

Full format of the TOIPARMS statement for UNLOAD

Use the TOIPARMS statement to specify the XCF group name to join when an A068 error is encountered on an UNLOAD function, and to request that a stop command is issued to IMS system for the current destination.

Additionally, specify ENABLESTOP to enable QCF to issue IMS **/ST0** commands, or specify DISABLESTOP to disable QCF from issuing IMS **/ST0** commands.

The full format of the TOIPARMS statement is shown as follows:

```
TOIPARMS XCFGROUP=TOIxxxxxx
```

or

```
TOIPARMS XCFGROUP=TOIxxxxxx,ENABLESTOP
```

or

```
TOIPARMS XCFGROUP=TOIxxxxxx,DISABLESTOP
```

When an A068 error is encountered on an UNLOAD function, IMS Queue Control Facility uses the XCF group name, which is specified in the statement to join the group and to request that a stop command is issued to IMS system for the current destination.

Unload continues within the same destination. If the full format of the TOIPARMS statement is not specified, no stop command is issued and the unload skips the current destination.

JCL for the UNLOAD function

The UNLOAD step is performed by IMS Queue Control Facility to remove messages from the shared queues for reinsertion later or for test purposes, similar to the REPROCESS procedure.

Use the UNLOAD step to remove messages from the shared queues for reinsertion later.

You can run the UNLOAD function in batch mode by submitting a JCL job stream.

Sample JCL to run the UNLOAD function

The following example shows the JCL that is required to start a UNLOAD step. Replace the items in italics with values that are appropriate for your environment:

```
//****************************************************************
//* SAMPLE JCL TO EXECUTE THE UNLOAD FUNCTION                    *
//****************************************************************
//RECOVER  EXEC PGM=IQCINI0$,
// PARM='BPEINIT=IQCBINI0,BPECFG=IQCIVPCF,IMSPLEX=IPLEX,QCFIMS=IMS1'
//STEPLIB  DD  DISP=SHR,DSN=hlqual.SIQCLINK
//         DD  DISP=SHR,DSN=IMSREL.RESLIB
//*
//QCFPRINT DD SYSOUT=A
//*
//UNLOAD   DD DSN=QCF.UNLOAD,UNIT=SYSDA,DISP=(NEW,CATLG),
//         DCB=(LRECL=32756,BLKSIZE=32760,RECFM=VB),
//         SPACE=(CYL,(5,5))
//*
//SNAPDUMP DD SYSOUT=A
//*
//QCFIN    DD DSN=USER.PRIVATE.PROCLIB(unload),UNIT=SYSDA
```

DD statements for UNLOAD

STEPLIB DD
> Defines the library containing the IMS Queue Control Facility program load modules and IMS libraries that are required for the UNLOAD function to run.
>
> This DD statement is required.

QCFPRINT DD
> Defines the output report data set.
>
> This DD statement is required.

UNLOAD DD
> Defines the output data set. This can be used as input to the LOAD function.
>
> This DD statement is optional.
>
> If ACTION=(READ) or ACTION=(REPORTONLY) is specified or defaulted to on the FUNCTION or SELECT statement, messages will be unloaded off of the queues and there will be no backup of the unloaded messages.

QCFIN DD
> Defines the input control statement data set.
>
> This DD statement is required.
>
> Specify DD DUMMY if you do not want to allocate the UNLOAD data set. However, the UNLOAD job will continue as if ACTION=REPORTONLY was specified, so the messages will be unloaded off of the queues and there will be no backup of the unloaded messages.

Example of the PROC USER.PRIVATE.PROCLIB(*unload*)

Following is an example of the PROC that was specified above in the QCFIN DD statement as, USER.PRIVATE.PROCLIB(*unload*):

```
***********************************************************
* unload the appc and lterm queues
***********************************************************
function unload queuetype(appc,lterm)
end
```

Chapter 18. Invoking the server commands

You can invoke most of the IMS Queue Control Facility server commands with the z/OS **MODIFY** command.

The exception is the command to stop the IMS Queue Control Facility server that can be entered either through the z/OS **MODIFY** command or on an operator console.

The following diagram illustrates the general syntax for entering the IMS Queue Control Facility commands that are available through the z/OS **MODIFY** command.

►►— F —— *jobname* —— , —— *command* —►◄

Command parameter reference:

F
> The z/OS **MODIFY** command.

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

command
> The command that you are issuing.

Topics:

Using wildcard characters in command expressions

You can use wildcard characters for pattern matching for some command parameters.

For these parameters, you can use the following wildcard character schemes:

*
> Matches zero or more characters.

%
> Matches exactly one character.

The following examples illustrate some possible uses of wildcard characters.

BE*
> Matches any string that begins with BE, of any length. For example, BE, BEE, or BEEBLEBROX.

%%S
> Matches any three-character string that ends with an S. For example, IMS or CQS.

R*S*T%R
> Matches any string that meets the following requirements:
>
> - Begins and ends with an R
> - Contains an S after the R
> - The S is followed by a string that contains a T
> - Has any number of intervening characters between the first R, the S, and the T
> - Has exactly one character between the T and the final R

For example, ROASTER, ROSTER, RESORTER, RECEPTOR, RSTZR.

*

Matches any string.

Querying server destinations with highest message queue use

Use the **QUERY TOPAM** command to display the top number of IMS Queue Control Facility server destinations with the highest message queue use.

The **QUERY TOPAM** command displays the top *nn* number of IMS Queue Control Facility server destinations that are using the message queues the most.

You specify the IMS ID and the *nn* number of destinations that you want to see usage information about.

QUERY TOPAM command syntax

The following diagram shows the syntax of the **QUERY TOPAM** command:

▶▶─┬─ QUERY ─┬─── QCFIMS ── NAME(imsid) ── INCLUDE ── (TOPAM(number)) ──▶◀
   └─ QRY ───┘
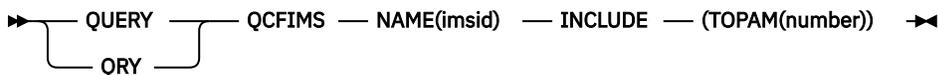
QUERY TOPAM command parameters

The **QUERY TOPAM** command uses the NAME and TOPAM parameters:

NAME(*imsid*)
This required parameter specifies a 1- to 4-character IMS name for the IMS Queue Control Facility extensions.

TOPAM(*number*)
This optional parameter specifies the number of the top destinations that are using the message queues to display.

Valid integers are 1 to 99.

Example display of top use destinations

The following example shows how to display the ten destinations that are the top users of the message queues on an IMS Queue Control Facility server:

```
F QCF31,QRY QCFIMS NAME(IMS1) INCLUDE(TOPAM(10))
```

The following output shows an example command response from the previous command. The destination names are listed under `Destinations`. The queue type of LTERM is listed under QT LT.

```
IQC3519I Top       10        All   Queue      Destinations     IMS1
IQC3520I Dest      QT      LongM   ShrtM  LongP/ShrtP    LastActive
IQC3521I TSUED01   LT      18        0   1.5%/ 0.0%  01/10/30-07:58:02:6
IQC3521I DESRZA70.A7CICHBT
IQC3521I -DFSASYNC AP      16        0   1.3%/ 0.0%  01/10/31-03:29:40:6
IQC3521I CLIENT1-
IQC3521I -T3270LC  OT      12        0   1.0%/ 0.0%  01/11/02-11:19:04:1
IQC3521I T0910122  LT      12        0   1.0%/ 0.0%  01/10/31-03:33:36:9
IQC3521I IMSNET.L62MVS1
IQC3521I -DFSASYNC AP       8        0   0.6%/ 0.0%  00/05/20-13:57:54:3
IQC3521I VTAGB588  LT       8        0   0.6%/ 0.0%  01/10/30-08:00:08:3
IQC3521I T1LRPT01  LT       8        0   0.6%/ 0.0%  01/10/30-07:25:27:4
IQC3521I T0912056  LT       8        0   0.6%/ 0.0%  01/10/30-08:16:58:8
IQC3521I L63SP2T1  LT       8        0   0.6%/ 0.0%  99/11/04-18:46:21:6
IQC3521I T2958327  LT       6        0   0.5%/ 0.0%  01/10/30-01:03:17:8
0090  IQC4998I QRYQIMS  RC=0000 AIBRC=0000 AIBRS=0000
```

Querying destinations exceeding specified time limit

Use the **QUERY OVERLM** command to display the destinations, over the last number of minutes that you specify, that have used the message queues.

The **QUERY OVERLM** command allows you to display the destinations over the last *nn* number of minutes that have used the message queues.

QUERY OVERLM command syntax

The following diagram shows the syntax of the **QUERY OVERLM** command.

```
►►─┬─ QUERY ─┬─── QCFIMS ── NAME ─( ── imsid ── ) ── INCLUDE(OVERLM( minutes )) ─►◄
   └─ QRY ───┘
```
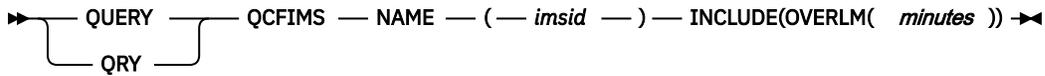
QUERY OVERLM command parameters

The **QUERY OVERLM** command uses the NAME and OVERLM parameters:

NAME(*imsid*)
> This required parameter specifies a 1- to 4-character IMS name for the IMS Queue Control Facility extensions.

OVERLM(*minutes*)
> This optional parameter specifies the destinations over the last *nnnn* number of minutes that have used the message queues.
>
> Enter a minutes value that is an integer from 1 to 1440.

Example destinations display

The following example shows how to display all of the destinations that have used the message queues during the last 10 minutes:

```
F QCF31,QRY QCFIMS NAME(IMS1) INCLUDE(OVERLM(10))
```

The following example shows the command response from the previous command. The destination names are listed under the **Dest** column. The queue type is listed under **QT**.

```
IQC3528I Over Last    10       Minute       Destinations   IMS1
IQC3529I Dest       QT     LastActive         LongM   ShrtM  LongP/ShrtP
IQC3530I BOXUE      LT  07/07/26-11:29:40:9       0       1    0.0%/ 0.0%
IQC3530I LIJI       LT  07/07/26-11:29:41:9       0       1    0.0%/ 0.0%
IQC4998I QRYQIMS  RC=0000 AIBRC=0000 AIBRS=0000
```

Starting a requeue or offload command

Use the **START TASK** command to start a requeue or offload.

START TASK command syntax

The following diagram shows the syntax of the **START TASK** command:

```
►►─┬─ START ─┬─── TASK ── NAME(member) ─►◄
   └─ STA ───┘
```
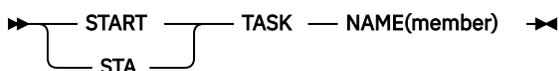
START TASK command parameter

The **START TASK** command uses the name parameter:

NAME
>This required parameter specifies a 1- to 8-character PROCLIB member name.
>
>The IMS Queue Control Facility server opens this member to find the control statements and processes the requeue or offload task accordingly.

Example command to start a task from a PROCLIB member

You can use the **START TASK** command to start a task from a PROCLIB member.

The following example shows the z/OS **MODIFY** command used to start a task from PROCLIB member *deadq*:

```
F QCFJOB,STA TASK NAME(deadq)
```

The following example shows the command response from the previous command. The command response in this case is a BPE message.

```
BPE0032I STA TASK COMMAND COMPLETED
```

Syntax for the server PROCLIB member

The PROCLIB member (IQCSCMDS) must be available to the IMS Queue Control Facility server, which uses it to initiate the specified function. Use the **START TASK** command to access the member.

The **START TASK** command uses the following syntax:

where:

NAME()
>specifies a 1- to 8-character PROCLIB member name.
>
>The server opens this member, finds the control statement, and processes a re-queue or offload task.

Examples of using the **START TASK** command can be found in *hlq*.SIQSAMP, member IQCSCMDS.

Parameter reference for the server PROCLIB member

The following parameters can be coded in the server PROCLIB member:

BROWSE(*user_data_set*)
>This parameter defines 1- to 44-character name of the data set that is used for BROWSE function output data.
>
>This parameter is optional.

IMS(*imsid*)
>This parameter defines a 1- to 4-character IMS ID name.

LOAD(*user_data_set*)
>This parameter defines 1- to 44-character name that is used for the IMS Queue Control Facility LOAD input data.
>
>This parameter is optional.

QCFIN(DSN(*user_proclib*),MEM(*iqcscmds*)
>This parameter defines a 1- to 44-character data set name (DSN) for the partitioned data set (PDS), and a 1- to 8-character name for the member that contains the IMS Queue Control Facility input command stream.
>
>This parameter is required.

QCFPRINT(*user_data_set*)
This parameter defines 1- to 44-character data set name that is used for the IMS Queue Control Facility print output data.

This parameter is required.

QUERY(*user_data_set*)
This parameter defines a 1- to 44-character name for the data set that is used for QUERY function output data.

This parameter is optional.

RECOVER(*user_data_set*)
This parameter defines a 1- to 44-character name for the data set that is used for RECOVERY function output data.

This parameter is optional.

SCRAPLOG(*user_data_set*)
This parameter defines 1- to 44-character name for the output data set that contains the rejected messages from the LOAD steps.

This parameter is required.

UNLOAD(*user_data_set*)
This parameter defines a 1- to 44-character name for the simple data set that is used for UNLOAD function output data.

This parameter is optional.

UNLOADGDG(DSN,UNIT,VOLSER,SPACE,SPACEUNIT,BLKSIZE,DATACLAS)
This parameter defines 1- to 44-character names for the generation data sets that are used for UNLOAD function output data (DSN, UNIT, VOLSER, SPACEUNIT, BLKSIZE, DATACLAS).

This parameter is optional.

Stopping the IMS Queue Control Facility server

To stop the IMS Queue Control Facility server, you can issue a z/OS **Purge** command from an operator console.

The following example shows the command that you can enter to stop, or actually purge, the IMS Queue Control Facility server from an operator console:

```
P qcfserver
```

Or, you can issue a z/OS **MODIFY SHUTDOWN** command.

The following example shows the z/OS **MODIFY** command that you would enter to stop the IMS Queue Control Facility server from an operator console:

```
F qcfserver,SHUTDOWN
```

Chapter 19. Using the TSO client (ISPF user interface)

Use the IMS Queue Control Facility TSO client (ISPF user interface) to select transaction queue interaction functions and queue overflow protection functions.

These topics describe the IMS Queue Control Facility TSO client (ISPF) panels in both nonshared- and shared-queues environments.

Use these IMS Queue Control Facility functions and selection criteria to perform the following tasks:

- Display the IMS queue environment and status
- Query the messages on all queue types
- Load previously copied or deleted (unloaded) messages
- View previously copied or deleted (unloaded) messages
- Find IMS processes waiting for resources
- Update queue overflow control files for committed and uncommitted message processing

After you start the ISPF feature and gain access to the IMS Queue Control Facility panels, you can use the **Help** pull-down menu for general panel help, or press the Help key (typically F1 or PF1) with your cursor on the field in question to access help information for each field.

The panel functions use temporary data sets, which are allocated while starting IMS Queue Control Facility.

Target authorized high-level qualifiers are required for creating and accessing the data sets. The target high-level qualifiers are set in the starting REXX EXEC (named IQC or whatever name you specified during customization of the starting REXX EXEC, concatenated with a time stamp string).

The high-level qualifiers are different for every new start of IMS Queue Control Facility.

If no value is set, your TSO ID is used as the high-level qualifier.

The list of temporary data sets is:

- *usrhlq*.BROWSE
- *usrhlq*.BROWSES
- *usrhlq*.QCFIN
- *usrhlq*.QCFPRINT
- *usrhlq*.QUERY
- *usrhlq*.UNLOAD

These data sets are deleted and reallocated while starting a IMS Queue Control Facility session.

Do not rely on these data sets to preserve information between IMS Queue Control Facility starts and even between different IMS Queue Control Facility functions. Use permanent data sets to copy or unload messages from the queues.

Note: The IMS Queue Control Facility panels do not operate as a real-time interface. When messages are displayed on the panels, they are from a snapshot of the message queue at the point in time that the TSO client obtained access to them.

How the panels are presented

The IMS Queue Control Facility panels are presented in these topics in the following ways:

- If no differences exist for shared and nonshared queues, only the nonshared version of the panel is shown.
- If differences do exist for shared and nonshared queues, one of the following methods is used:

- – If the differences are slight, the nonshared version of the panel is shown with the differences are highlighted in bold text and described.
- – If the differences are greater than can be easily described, both versions of the panel are shown; nonshared first, followed by the shared queues version, if one exists. Otherwise, the shared queues version is shown.
- The PF or function key settings are not shown in these topics, but they might be displayed on your live panels.

IMS Queue Control Facility main menu panel

Use this panel to select an IMS Queue Control Facility function and its related tasks.

Panel IQCP00 shows the Main Menu panel for IMS Queue Control Facility.

This panel allows you to select a function and its related tasks.

⚠️ **Attention:** If you press the Attn key while waiting for a response from IMS Queue Control Facility, IMS Queue Control Facility might become inoperable or the ISPF panels might become inaccessible. To resolve this issue, refresh IMS Queue Control Facility.

⚠️ **Attention:** The options under Queue Overflow Protection Functions on the Main Menu panel that appear in **bold** text apply to messages in nonshared queues only. In a shared-queues environment, the Queue Overflow Protection Tasks 4, 5, and 6 will not be activated and you will be unable to choose them.

The options under Queue Overflow Protection Functions on the Main Menu panel apply to messages with an in-process status that is incomplete. In process, uncommitted tasks can continue to request more space in the message queue.

The queue space overflow protection logic monitors the queues and determines the actions to be performed on the in-process, uncommitted tasks and on the complete, committed messages. The actions are performed after area thresholds and the thresholds that you set for queue usage are crossed.

Function 4 displays waited messages only after the queue space overflow protection logic determines that a task must wait.

```
   Preferences  Help
 IQCP00                       QCF Main Menu
 Option ===>

 Select an option or press END to exit.
                                            APAR . . : PH19814 2020/01/
 Server . . :                               JDTE . . : 2020.016
 IMS ID . . :                               TIME . . : 09:56:59
                                            DATE . . : 2020/01/16
                                               Server and IMS selection
  0   Select - PLEX, Server and IMS to be used

      Transaction Queue Interactive Functions
  1   Status - IMS environment and queue statistics
  2   Query  - List destinations with queued messages
  3   Load   - Re-insert removed queued messages
  3a  View   - View unloaded messages data set

      Queue Overflow Protection Functions
  4   Wait   - List and operate on waited tasks
  5   Tables - View, modify, and load overflow parameters
  6   Notify - Modify queue space utilization notification parameters

 Option 6 is active, so mutually exclusive option 5 is unavailable
```

Figure 20. Main Menu panel (IQCP00)

For more information about how to use the IMS Queue Control Facility Main Menu panel, use the **Help** pull-down menu. For detailed field-level help, press the Help key with your cursor on the field in question.

The following figure illustrates these differences between Transaction Queue Interactive Functions and Queue Overflow Protection Functions:

TRANSACTION QUEUE
INTERACTIVE FUNCTIONS

QUEUE OVERFLOW
PROTECTION FUNCTION

Messages enqueued and
ready for processing

Tasks in-process but
not complete

Transactions
Terminals
MSC links
APPC output
OTMA output
Dead queue output

Messages

Messages

Programs
Terminals
MSC links
APPC sessions
OTMA sessions

Local queue buffers

Queue data sets

QBLK   SMSGQ   LMSGQ

QSN waited list
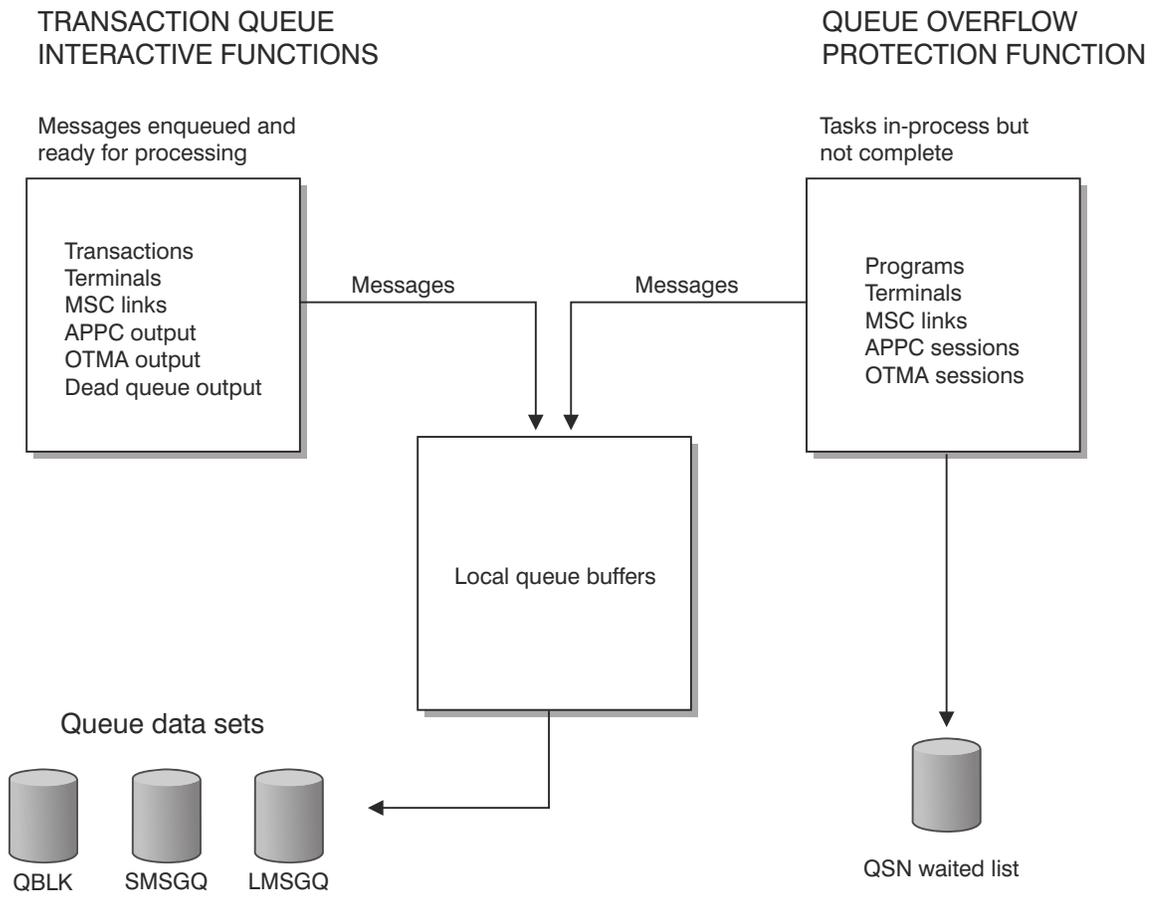
Transaction queue interactive functions let you
view information about queue entries that are
ready for processing. These messages can be:

- Copied
- Listed
- Viewed
- Unloaded
- Deleted
- Unlocked

Queue overflow protection function lets you
view processes on the wait queue. These tasks
can be:

- Listed
- Viewed
- Released
- Aborted
- Suspended
- Stopped

Figure 21. QSUN message queue processing

From the Main Menu panel, if you choose the **Preferences** pull-down menu, a pull-down list is displayed,
giving you the option to change your session preferences or to restore your session default preferences:

```
  Preferences   Help

  _____   _____
 |    1. Change QCF session preferences     |
 |    2. Restore QCF session default preferences |
 |_____|
 Select an option or press END to exit.
                                           APAR . . : PH19814 2020/01/
 Server . . :                              JDTE . . : 2020.016
 IMS ID . . :                              TIME . . : 17:00:18
                                           DATE . . : 2020/01/16
                                                           More:

     Server and IMS selection
  0  Select - PLEX, Server and IMS to be used

     Transaction Queue Interactive Functions...
```

Figure 22. IMS Queue Control Facility Main Menu with Preferences pull-down list displayed

If you choose option **1 - Change QCF session preferences**, the following Update Defaults dialog box is displayed:

```
 IQCP34V                         Update Defaults
 Command ===>                                           Scroll ===> PAGE

 Press END to save updates
                                        APAR . . : PH19814 2020/01/
  QCF Product Default Preferences

 Edit QCF           Display GMT as     Include MVSID in    Include IMSID
 generated stmts    local time         DSNames             in DSNames
 Y  (Default = Y)   N  (Default = N)   Y  (Default = Y)    Y  (Default =
                                                              Y)

 Maximum Number of messages to be added to the display at one time:
 03000  (Numeric value 00005 to 99999, default is 03000)

  Permanent msg UNLOAD work data set high lvl qualifier or blank

  Permanent SCRAPLOG work data set high level qualifier or blank


 Plex Names    PLEX1
 ----------
```

Figure 23. Update Defaults (IQCP34V) dialog box

A single TSO client can connect to a maximum of 16 IMSplexes. A new IMSplex can be added or an existing IMSplex can be deleted by using option **1**. Up to 16 IMSplex names can be entered in panel IQCP34V and they are saved in the TSO user profile. If there are no existing IMSplexes in the profile, the IMSplex name in the starting script is added to the list and the list is saved.

Note: GMT means Greenwich mean time.

For more information about how to use the Update Defaults dialog, use the **Help** pull-down menu. For detailed field-level help, press the Help key with your cursor on the field in question.

When you are finished with the Update Defaults dialog, press your Exit key and the Main Menu panel is displayed again.

If you choose option **2 - Restore QCF session default preferences** from the **Preferences** pull-down menu and answer Yes to the prompt, your session default preferences are restored.

The QCF Main Menu panel is displayed again with messages stating that your session defaults have been restored and to what your session defaults have been restored:

```
   Preferences  Help
 ------------------------------------------------------------------------------
 IQCP00                            QCF Main Menu
 Option ===>

 Select an option or press END to exit.
                                                      APAR . . : PH19814 2020/01/
 Server . . : _____                                JDTE . . : 2020.016
 IMS ID . . : _____                                TIME . . : 17:15:08
                                                      DATE . . : 2020/01/16
                                                                  More:     +
       Server and IMS selection
  0    Select - PLEX, Server and IMS to be used

       Transaction Queue Interactive Functions
 .----------------------------------------------------------------------------.
 ! Default settings restored to:                                              !
 ! -Edit QCF generated control statements.                                    !
 ! -Do not display times in GMT on QCF panels.                                !
 ! -03000 for number of messages to be added to the current display.         !
 ! -Permanent messages work data set name high level qualifier not provided.  !
 ! -Permanent SCRAPLOG work data set name high level qualifier not provided.  !
 '----------------------------------------------------------------------------'
  6    Notify - Modify queue space utilization notification parameters
```

Figure 24. QCF Main Menu

IMS Queue Control Facility IMSplex, server, and IMS selection panels

Use the QCF Server selection panel to select the IMS Queue Control Facility IMSplex and server that you want to use. Use the QCF IMS selection panel to select an IMS system to which you want to connect.

Choose the required option zero (0) on the Main Menu panel to display the QCF Server selection panel (IQCP002X). Select the IMSplex and server to use from the list on this panel by entering S in the **Action** column on that server's row, and press Enter.

```
   Help

 ─────────────────────────────────────────────────────────────────────────────
 IQCP002X                   QCF Server selection panel          Row 1 to 1 of 1
 Command ===>                                                   Scroll ===> PAGE

 Enter S to select a server or press END to exit
                                                      APAR . . : PH19814 2020/01/
                                                      JDTE . . : 2020.039
                                                      TIME . . : 15:15:40
                                                      DATE . . : 2020/02/08


         Server        Server Job   Server MVS
 Action  Member name   Name         ID             Plex name    Status
 S       IQCSERV1      IQCSERV1     EC01593        PLEX1        NO IMS
 ****************************** Bottom of data ********************************
```

Figure 25. QCF Server selection panel (IQCP002X)

For more information about how to use this panel, use the **Help** pull-down menu. For detailed field-level help, press the Help key (typically F1 or PF1) with your cursor on the field in question.

When you press Enter, the QCF IMS selection panel (IQCP001X), as shown in the following figure, is displayed. Use this panel to select an IMS system to which you want to connect.

To choose the IMS system that you want to process, enter S in the **Action** column on that IMS system's row and press Enter.

```
   Help
 _____
 IQCP001X                  QCF IMS selection panel              Row 1 to 1 of 1
 Command ===>                                                   Scroll ===> PAGE

 Enter S to select an IMS or press END to exit
                                               APAR . . : PH19814 2020/01/
 Server . . : QCFSERVE                         JDTE . . : 2020.039
                                               TIME . . : 15:16:11
                                               DATE . . : 2020/02/08
                  ----------------- IMS ------------------
 Action    IMSID  Member name  Job name    MVS ID
 S         IMS1   IQCFIMS1     IMS1        ECDV255
           IMSA   IQCFIMSA     IMSA        ECDV256
 ***************************** Bottom of data *******************************
```

Figure 26. QCF IMS selection panel (IQCP001X)

For more information about how to use this panel, use the **Help** pull-down menu. For detailed field-level help, press the Help key (typically F1 or PF1) with your cursor on the field in question.

When you press Enter, the Main Menu panel is displayed again with the server and IMS ID fields populated.

Status sub-menu or Shared Queues Status sub-menu

Use the Status Sub-Menu panel or the Shared Queues Status Sub-Menu panel to display the IMS environment, local queue status, and for shared queues, the structure status.

To display the Status Sub-Menu panel or the Shared Queues Status Sub-Menu panel, choose **1 - Status** from the Main Menu panel. The Status Sub-Menu panel or the Shared Queues Status Sub-Menu panel is used to display the IMS environment, local queue status, and for shared queues, the structure status.

From the Status Sub-Menu panel or the Shared Queues Status Sub-Menu panel, you can perform the following tasks:

- Choose Task 1 to show the IMS environment's system job, version, start time, restart, and feature information (see Figure 28 on page 275).
- Choose Task 2 to show the local queue status' IMS message queue summary status, counts, and capacity information (see Figure 29 on page 276).
- For shared queues only, choose Task 3 (shown in bold text) to show shared queue structure status, primary and overflow statistics information (see Figure 30 on page 277.

```
   Help
 _____
 IQCP00A | IQCP00B         Shared Queues Status Sub-Menu
 Option ===>

 Select an option or press END to exit.
                                               APAR . . : PH19814 2020/01/
 Server . . : QCFSERVE                         JDTE . . : 2020.039
 IMS ID . . : IMS1                             TIME . . : 15:18:10
                                               DATE . . : 2020/02/08

  1   Environment - Show IMS environment
  2   Local Queue - Show local queue status
  3   Structure Status - Show shared queue structure status
```

Figure 27. Status Menu panel (IQCP00A or IQCP00B) for nonshared or shared queues

For more information about how to use the Status Sub-Menu panel or the Shared Queues Status Sub-Menu panel, use the **Help** pull-down menu. For detailed field-level help, press the Help key with your cursor on the field in question.

When you are finished with the Status Sub-Menu panel or the Shared Queues Status Sub-Menu panel, press the End key to return to the Main Menu.

IMS Environment panel

Use the IMS Queue Control Facility IMS Environment panel to view the IMS environment information.

To display the Environment panel, select option **1- Environment** on the Status Sub-Menu panel or the Shared Queues Status Sub-Menu panel.

Use the Environment panel to view the IMS environment information. The entire panel is shown here for simplicity; you must actually press your Forward key (for example, PF8) to see the last few lines of the panel.

The items that appear in **bold text** apply to, and can be viewed only in a shared-queues environment.

The items that appear in *italics* apply to, and can be viewed only in a nonshared-queues environment.

```
  Help

 ────────────────────────────────────────────────────────────────────────
 IQCP010L | IQCP010S          QCF IMS Environment
 Command ===>

 Press ENTER to refresh status or press END to exit.
                                             APAR . . : PH19814 2020/01/
 Server . . : QCFSERV1                       JDTE . . : 2020.039
 IMS ID . . : IMS1                           TIME . . : 19:13:43
                                             DATE . . : 2020/02/08
                                                        More:
 IMS Environment
 IMS job name . . . . . . . . . . . : IMS1
 JES ID . . . . . . . . . . . . . . : JOB00085
 IMSID/RSENAME  . . . . . . . . . . : IMS1
 Version  . . . . . . . . . . . . . : 1510
 GMT start time . . . . . . . . . . : 2020.036/18.09.20.23
 Restart type . . . . . . . . . . . : COLD
  Shared Queue Environment
 CQS job name . . . . . . . . . . . : CQSEI1
 CQS group name . . . . . . . . . . : DFSGRUP1
  Structure Data
 Type . . . . . . . . . . . . . . . : MSG QUEUE
 Attributes . . . . . . . . . . . . : WAIT FOR REBUILD
 Status . . . . . . . . . . . . . . : AVAILABLE
  Components/Optional Features
                               MSC
                               SHARED QUEUES
                               FAST PATH
                               OTMA

                               _____
                               IRLM(NAME=IRLM)
  QCF Options
   Overflow protection installed-QSUN option 6 is active
```

Figure 28. IMS Queue Control Facility IMS Environment panel (IQCP010L)

For more information about how to use the Environment panel, use the **Help** pull-down menu. For detailed field-level help, press the Help key with your cursor on the field in question.

When you are finished with the Environment panel, press the End key to return to a display of the Status Sub-Menu panel or the Shared Queues Status Sub-Menu panel.

Local Queue Status panel

Use the Local Queue Status panel to view the status of messages on the queues.

To display the Local Queue Status panel, select option **2 - Local Queue** on the Status Sub-Menu panel or the Shared Queues Status Sub-Menu panel.

The Local Queue Status panel is used to view the status of messages on the queues. The entire panel is shown here for simplicity, you must actually press your FORWARD key to see the last few lines of the panel.

The items on this panel that appear in bold text are available (and are displayed) only for nonshared queues.

```
  Help
_____
IQCP010 | IQCP011            QCF Local Queue Status
Command ===>

Press ENTER to refresh status or press END to exit.
                                         APAR . . : PH19814 2020/01/
Server . . : QCFSERV1                    JDTE . . : 2020.039
IMS ID . . : IMS1                        TIME . . : 19:20:42
                                         DATE . . : 2020/02/08
                                                       More:
 Local Message Queue Status
 In-core buffers  . . . . . . . . . :        20
 Buffer blocksize . . . . . . . . . :      2520
 Locate calls . . . . . . . . . . . :         6
 Release calls  . . . . . . . . . . :        37
 Purge calls  . . . . . . . . . . . :         1
 Reads  . . . . . . . . . . . . . :        18
 Writes . . . . . . . . . . . . . :        19
 Local Message Queue Data Set Capacity
 QBLK records . . . . . . . . . . . :      1420
  Inuse count/percentage  . . . . . :         4      <1  %
  Available count/percentage  . . . :      1416     100  %
  Highwater count/percentage  . . . :         4      <1  %
 Short message queue records  . . . :     53800
  Inuse count/percentage  . . . . . :        14      <1  %
  Available count/percentage  . . . :     53786     100  %
  Highwater count/percentage  . . . :        14      <1  %
 Long message queue records . . . . :     10600
  Inuse count/percentage  . . . . . :         1      <1  %
  Available count/percentage  . . . :     10599     100  %
  Highwater count/percentage  . . . :         2      <1  %
```

Figure 29. Local Queue Status panel (IQCP010 | IQCP011)

For more information about how to use the Local Queue Status panel, use the **Help** pull-down menu. For detailed field-level help, press the Help key with your cursor on the field in question.

When you are finished with the panel, press the End key to return to a display of the Status Sub-Menu panel or the Shared Queues Status Sub-Menu panel.

Queue Structure Status panel for shared queues only

Use the Queue Structure Status panel to view the structure status and primary and overflow structure statistics.

To display the Queue Structure Status panel, select option **3 - Structure Status** on the Shared Queues Status Sub-Menu panel. The Queue Structure Status panel is used to view the structure status and primary and overflow structure statistics. The entire panel is shown here for simplicity; you must actually press your FORWARD key to see the last few lines of the panel.

```
  Help
 _____
IQCP012                      Queue Structure Status
Command ===>                                                 Scroll ===> PAGE

Press ENTER to refresh status or press END to exit.
                                                 APAR . . : PH19814 2020/01/
Server . . : QCFSERVE                            JDTE . . : 2020.039
IMS ID . . : IMSA                                TIME . . : 18:51:18
                                                 DATE . . : 2020/02/08
                                                              More:
 Structure Status
 In-core buffers . . . . . . . . . :       200
 Buffer blocksize  . . . . . . . . :      2520
 Locate calls  . . . . . . . . . . :         7
 Release calls . . . . . . . . . . :        37
 Purge calls . . . . . . . . . . . :         1
 CQSREAD requests  . . . . . . . . :         0
 CQSPUT requests . . . . . . . . . :         0
 Structure Statistics           Primary           Overflow
 Structure Name  . . . . . . . . . : IMSMSGQ01         IMSMSGQ01OFLW
 List Entries - Allocated  . . . . :    18084               0
 List Entries - Inuse  . . . . . . :        3               0
 inuse/entries Alloc Ratio:  . . . :        1               0
 Elements - Allocated  . . . . . . :    18084               0
 Elements - Inuse  . . . . . . . . :        3               0
 inuse/Element Alloc Ratio:  . . . :        1               0
```

Figure 30. Queue Structure Status panel (IQCP012)

For more information about how to use the Queue Structure Status panel, use the **Help** pull-down menu.
For detailed field-level help, press the Help key with your cursor on the field in question.

When you are finished with the panel, press End to return to a display of the Shared Queues Status
Sub-Menu panel.

Query panels and dialog boxes

Use the Query Menu panel to process queries that are based your on selection criteria and to list
messages using minimum summary criteria to list messages using additional filter statements.

To display the Query Menu panel, select option **2 - Query** on the Main Menu panel. Use the Query Menu
panel to process queries that are based your on selection criteria. Also, use the Query Menu to list
messages using minimum summary criteria to list messages using additional filter statements (INCLUDE
and EXCLUDE).

The Query panel is shown as one screen here for simplicity; you must actually press your FORWARD key
to see the bottom several lines of the Query panel.

```
   Help

IQCP91A                              Query
Command ===>                                                    Scroll ===> PAGE

Press ENTER to continue or END to exit.
                                                   APAR . . : PH42117 2021/12/
Server . . : IFD7QCFS                              JDTE . . : 2022.017
IMS ID . . : IFD7                                  TIME . . : 02:25:47
                                                   DATE . . : 2022/01/17
 Message Queue include/exclude and filtering criteria

Get summary messages information . . . . Y  Y/N
Get detailed messages information  . . . N  Y/N
 Queue Type Filters within Local Message Queue:
Enter "/" to select Q type filters, if ALL no other filter can be used
 /  ALL                      APPC                      Dead Queue
    LTERM                    OTMA                      Remote
    Serial                   Suspend                   Transaction
 Options
 Get old msgs  . . 00000000   Number of time periods prior to current time
 Unit of time  . . D          D/S  D - Days(24-hour periods)  S - Seconds
/  Long msg queue       /  Short msg queue
 Record count  . . N       Y/N
 Generate parameters for:
 INCLUDE (Y) . . .                    EXCLUDE (Y) . . .

QCF user built control statement data set, if required
(e.g. Fully qualified data set name up to 44 characters in quotes):
 Data set name . .
 Member  . . . . .
```

Figure 31. Query Menu panel (IQCP91A)

For more information about how to use the Query panel, use the **Help** pull-down menu. For detailed field-level help, press the Help key with your cursor on the field in question.

Restriction: Specifying **Y** for **Record count** on the Query panel causes the display of only the primary count on the panel report. Primary and secondary (**Primry** and **Secdry**) counts are displayed in the printed report.

When you are finished with the Query panel, press the End key to return to a display of the Main Menu panel.

Query for Shared Queues panel

Use the Query for Shared Queues panel to process queries that are based your on selection criteria and to list messages using minimum summary criteria using additional filter statements.

To display the Query for Shared Queues panel, select option **2 - Query** on the Main Menu panel. Use the Query for Shared Queues panel to process queries that are based your on selection criteria. Also, use the Query Menu to list messages using minimum summary criteria using additional filter statements (INCLUDE and EXCLUDE).

The Query for Shared Queues panel is shown as one screen here for simplicity; you must actually press your FORWARD key to see the bottom several lines of the panel.

```
   Help

IQCP91S1                    Query for Shared Queues
Command ===>                                                     Scroll ===> PAGE

Press ENTER to continue or END to exit.
                                                       APAR . . : PH42117 2021/12/
Server . . : IFD8QCFS                                  JDTE . . : 2022.017
IMS ID . . : IFD8                                      TIME . . : 02:23:28
                                                       DATE . . : 2022/01/17
 Structure/message queue include/exclude and filtering criteria

Get summary messages information . . . . Y  Y/N
Get detailed messages information  . . .    Y/N
 CQS Structure/Message Queue Filters, enter "/" to select
/  Primary              / Overflow            / Local
 Queue Type Filters within Local Message Queue:
Enter "/" to select Q type filters, if All no other filter can be used
 /  ALL                      APPC                    Cold Queue
    LTERM                     OTMA                    Remote
    Serial                    Suspend                 Transaction
 Options
 Get old msgs  . . 00000000   Number of time periods prior to current time
 Unit of time  . . D          D/S  D - Days(24-hour periods)  S - Seconds
 Record count  . . N      Y/N
Generate parameters for:
 INCLUDE (Y) . . .                   EXCLUDE (Y) . . .
QCF user built control statement data set, if required
(e.g. Fully qualified data set name up to 44 characters in quotes):
 Data set name . .
 Member  . . . . .
```

Figure 32. Query for Shared Queues Menu panel (IQCP91S1)

For more information about how to use the Query for Shared Queues panel, use the **Help** pull-down menu. For detailed field-level help, press the Help key with your cursor on the field in question.

When you are finished with the panel, press the End key to return to a display of the Main Menu panel.

Edit parameters generated by the Query (shared and nonshared queues)

Use the following panel to edit the parameters generated by the Query (for shared and nonshared queues) to edit and check the function and parameters for correctness.

The edit session for the Query generated parameters panel is displayed only if you specified Y for the **Edit QCF generated stmts** field on the Update Defaults (IQCP34V) dialog box.

When you generate parameters on either of the Query panels, an ISPF edit session similar to the following is displayed. Check the function and parameters for correctness and press the End key.

```
   File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
 ------------------------------------------------------------------------------
ISREDDE2   IQC.X0012345.QCFIN                            Columns 00001 00072
Command ===>                                                  Scroll ===> PAGE
****** **************************** Top of Data ******************************
000001 FUNCTION BROWSE action=read,
000002 queuetype=(all),
000003 age=0,
000004 area=(global,local,overflow)
000005 END
****** ************************** Bottom of Data ****************************
```

Figure 33. Edit session for Query generated parameters

Execute Confirmation dialog

Use the Execute Confirmation dialog box to validate the fact that you want to run your Query generated parameters.

When you press the End key from the edit session to validate Query generated parameters, this dialog box is displayed.

```
   Help
 --------------------------------------------------------------------------
| IQCP96                       Execute Confirmation                        |
| Command ===>                                            Scroll ===> PAGE  |
|                                                                          |
| Press ENTER to continue or END to exit.                                  |
|                                                                          |
|                                            APAR . . : PH19814 2020/01/    |
| Server . . : QCFSERV1                                                    |
| IMS ID . . : IMSA                                                        |
| Execute: . . Y  Execute the specified control statements.               |
|                                                                          |
|                                                                          |
 --------------------------------------------------------------------------
```

Figure 34. Execute Confirmation dialog box

Messages Destinations (Summary) panel (shared and nonshared queues)

Use the Messages Destinations (Summary) panel to display a summary of your message destinations for copy, delete, copy and delete, list, or unlock actions, for shared and nonshared queues.

To display the Messages Destinations (Summary) panel, Enter Y (for yes) in the **Get summary messages information** field on the Query panel, exit Figure 33 on page 279, and then answer Y on Figure 34 on page 280.

```
   View  Table_Actions  Help
 _____
 IQCP94C               Messages Destinations (Summary)       Row 1 to 9 of 39
 Command ===>                                            Scroll ===> PAGE

 Select a row action or press END to exit
                                                 APAR . . : PH19814 2020
 Server . . : QCFSERVE                           JDTE . . : 2020.039
 IMS ID . . : IMS1                               TIME . . : 15:31:48
 QCF Func . : SUMMARY                            DATE . . : 2020/02/08
 MSGQs  . . : ALL
 Row actions: C - Copy  D -Delete  X - Copy/Delete  L - List  U - Unlock

                                  Structure Queue Primary
 Act  Destination name           Queue     Type  Msgcnt   Status
      TSUED01                     LOC       LT       54    _____
      DESRZA70.A7CICHBT DFSASYNC  LOC       AP       48    _____
      T0910122                    LOC       LT       36    _____
      T0912056                    LOC       LT       24    _____
      L63SP2T1                    LOC       LT       24    _____
      T1LRPT01                    LOC       LT       24    _____
      IMSNET.L62MVS1   DFSASYNC   LOC       AP       24    _____
      T0910582                    LOC       LT       18    _____
      T0913000                    LOC       LT       18    _____
```

Figure 35. Messages Destinations (Summary) panel (IQCP94C)

For more information about how to use the Messages Destinations (Summary) panel, use the Help pull-down menu. For detailed field-level help press the Help key with your cursor on the field in question.

If you choose the **Table_Actions** pull-down menu, the following panel is displayed:

```
   View  Table_Actions  Help
 _____ _____
IQCP9 |   1. Copy all messages displayed in table            |  1 to 9 of 39
Comma |   2. Delete all messages displayed in table          |  ll ===> PAGE
      |   3. Copy then delete all messages displayed in table |
Selec |_____|
                                                        APAR . . : PH19814 202
Server . . : QCFSERVE                                   JDTE . . : 2020.039
IMS ID . . : IMS1                                       TIME . . : 15:31:48
QCF Func . : SUMMARY                                    DATE . . : 2020/02/08
MSGQs  . . : ALL
Row actions: C - Copy  D -Delete  X - Copy/Delete  L - List  U - Unlock

                                Structure Queue Primary
Act   Destination name          Queue    Type  Msgcnt   Status
      TSUED01                    LOC      LT        54   _____
      DESRZA70.A7CICHBT DFSASYNC LOC      AP        48   _____
      T0910122                   LOC      LT        36   _____
      T0912056                   LOC      LT        24   _____
      L63SP2T1                   LOC      LT        24   _____
      T1LRPT01                   LOC      LT        24   _____
      IMSNET.L62MVS1   DFSASYNC  LOC      AP        24   _____
      T0910582                   LOC      LT        18   _____
      T0913000                   LOC      LT        18   _____
```

*Figure 36. The **Table_Actions** pull-down list on the Query panel (IQCP9AC)*

If you choose option **1- Copy all messages displayed in table**, the Copy DSN dialog box (panel IQCP94B) is displayed where you can specify a data set into which the copied message will be placed. If the data set you specify does not exist, then the DCB Parameters to Allocate a Data Set panel (IQCP98) is displayed.

```
 _____
| IQCP94B                          Copy DSN                                |
| Command ===>                                          Scroll ===> PAGE   |
|                                                                          |
| Press ENTER to continue or END to exit.                                  |
|                                                       APAR . . : PH19814 2020/ |
| Server . . : QCFSERVE                                                    |
| IMS ID . . : IMS1                                                        |
| MSGQs  . . : ALL                                                         |
|                                                                          |
| Enter the data set name where the messages                               |
| will be copied into:                                                     |
|                                                                          |
|                                                                          |
|                                                                          |
|                                                                          |
|                                                                          |
|_____|
```

Figure 37. Copy DSN dialog box (IQCP94B)

Specify a data set in which to place the sorted message information. Ensure that you specify a data set large enough to hold the messages, otherwise an S000 abend might be triggered.

For detailed field-level help on the Copy DSN dialog box, press the Help key with your cursor on the field in question.

For more information about how to use the DCB Parameters to Allocate a Data Set panel, use the **Help** pull-down menu. For detailed field-level help press the Help key with your cursor on the field in question.

```
   Help
 _____
IQCP98                  DCB Parameters to Allocate ABC
Command ===>                                            Scroll ===> PAGE

Press ENTER to continue.
                                            APAR . . : PH19814 2020/01/
Server . . : QCFSERVE                       JDTE . . : 2020.039
IMS ID . . : IMSA                           TIME . . : 15:59:36
QCF task . : SUMMARY                        DATE . . : 2020/02/08

Data set name
ABC

 If data set is to be allocated enter the following DCB parameters:

Volume serial  . . . DUMVOL  Space  . . . . . . .  ____  Primary
Cyl/Trk  . . . . . .  ____                          ____  Secondary
```

Figure 38. DCB Parameters to Allocate a Data Set panel (IQCP98)

If you choose option **2 - Delete all messages displayed in table**, the Copy DSN dialog box (IQCP94B) is displayed where you specify a data set into which to put the copied messages before they are deleted from the queue. Ensure that you specify a data set large enough to hold the messages; otherwise, an S000 abend might be triggered.

If you choose option **3 - Copy then delete all messages displayed in table**, the Copy DSN dialog box (IQCP94B) is displayed where you specify a data set into which to put the copied messages before they are deleted from the queue. When you finish entering the DSN on the Copy DSN dialog box, the Confirm Delete dialog box is displayed.

```
 _____
| IQCP97                        Confirm Delete                         |
| Command ===>                                           Scroll ===> PAGE |
|                                                                      |
| Press ENTER to continue or END to exit.                              |
|                                            APAR . . : PH19814 202    |
| Server . . : QCFSERVE                                                |
| IMS ID . . : IMS1                                                    |
| Destination(s) will be deleted starting with:                        |
| MESSAGES FOR DESTINATION  TSUED01  WILL ONLY BE DELETED              |
|                                                                      |
| Warning: . . N  This panel is displayed if option D (DELETE) is      |
|                 specified on any QCF table.                           |
|                 Enter Y to proceed with deletion of                   |
|                 destination(s) or message(s).                        |
|                 Enter N to stop deletion of destination(s)/message(s). |
|                                                                      |
|_____|
```

Figure 39. Confirm Delete dialog box (IQCP97)

For detailed field-level help on the Confirm Delete dialog box, press the Help key with your cursor on the field in question.

When you are finished with the Messages Destinations (Summary) panel, press the End key to return to a display of the Query panel.

Messages Destinations (Detail) panel (shared and nonshared queues)

Use the Messages Destinations (Detail) panel to display a detailed view of your message destinations for copy, delete, copy/delete, list, or unlock actions for shared and nonshared queues.

To display the Messages Destinations (Detail) panel, enter Y (for yes) in the **Get detailed messages information** field on the Query panel, exit Figure 33 on page 279, and then answer Y on Figure 34 on page 280.

```
 View  Table_Actions  Help
 ─────────────────────────────────────────────────────────────────────
IQCP94                  Messages Destinations (Detail)        Row 1 to 9 of 39
 Command ===>                                            Scroll ===> PAGE

 Select a row action or press END to exit
                                                  APAR . . : PH19814 202
 Server . . : QCFSERVE                            JDTE . . : 2020.039
 IMS ID . . : IMS1                                TIME . . : 15:41:05
 QCF Func . : DETAIL                              DATE . . : 2020/02/08
 MSGQs  . . : ALL
 Row actions: C - Copy  D - Delete  X - Copy/Delete  L - List  U - Unlock

                            Str Q  - Pct - ------ Msgcnt -------
 A Destination name         Q   Tp  Lmg Smg Primry Secdry Total   Status
   TSUED01                  LOC LT   0   0     54      0     54 _____
   DESRZA70.A7CICHBT DFSASYNC LOC AP   0   0     48      0     48 _____
   T0910122                 LOC LT   0   0     36      0     36 _____
   T0912056                 LOC LT   0   0     24      0     24 _____
   L63SP2T1                 LOC LT   0   0     24      0     24 _____
   T1LRPT01                 LOC LT   0   0     24      0     24 _____
   VTAGB588                 LOC LT   0   0     12     12     24 _____
   IMSNET.L62MVS1  DFSASYNC LOC AP   0   0     24      0     24 _____
   T2958327                 LOC LT   0   0      6     12     18 _____
```

Figure 40. Messages Destinations (Detail) panel (IQCP94)

For more information about how to use the Messages Destinations (Detail) panel, use the Help pull-down menu. For detailed field-level help, press the Help key with your cursor on the field in question.

Selecting **List** (**L**) will display panel IQCP95, as shown in the following figure:

```
  View  Table_Actions  Help
 ─────────────────────────  ─────────────────────────────────────────
|   1. Sort table columns |  ted Destination Messages        Row 1 to 6 of 432
|_____|                            Scroll ===> PAGE

 Select a row action or press END to exit
                                                  APAR . . : PH19814 2020/01/
 Server . . : QCFSERVE                            JDTE . . : 2020.039
 IMS ID . . : IMS1                                TIME . . : 15:37:24
 IMS Job  . : IMS1                                DATE . . : 2020/02/08
 QCF Func . : SUMMARY
 Msgs for . : Q TYPE=LTERM, DEST=TSUED01
 View formatted IMS prefix  . . Y  View IMS prefix  . . N
 View user data . . . . . . . . N  View QCF prefix  . . N
 Row actions: C - Copy  D - Delete  X - Copy/Delete  V - View  U - Unlock

                                     Secdry
 A Source Name      IMS ID   Format   SYS  Msgcnt Date    Time        Status
 _ LUNAME = L62MVS SYS3     _____  10      0 2020038 20575432925  _____
 _ LUNAME = L62MVS SYS3     _____  10      0 2020038 20575432925  _____
 _ LUNAME = L62MVS SYS3     _____  10      0 2020038 20575432925  _____
 _ LUNAME = L62MVS SYS3     _____  10      0 2020038 20575432925  _____
 _ LUNAME = L62MVS SYS3     _____  10      0 2020038 20575432925  _____
 _ LUNAME = L62MVS SYS3     _____  10      0 2020038 20575432925  _____
```

Figure 41. The View pull-down on the Query panel (IQCP95)

If you choose the **View** pull-down menu, a pull-down list is displayed, giving you the option to sort table columns. If you specify option **1 - Sort table columns**, the following Sort Columns dialog box is displayed.

```
| IQCPS020            Sort Columns                          |
| Command ===>                                              |
|                                                           |
| Press ENTER to continue or END to exit.                   |
|                                       APAR : PH19814 2     |
|                                                           |
|                        A/D    Seq 1-6                     |
| Source Name . . . . . _        _                          |
| Source IMSID  . . . . _        _                          |
| Source System ID  . . _        _                          |
| Source Format . . . . _        _                          |
| Secondary Msgcnt  . . _        _                          |
| Date/Time . . . . . . _        _                          |
|                                                           |
|                                                           |
|                                                           |
|                                                           |
|_____|
```

Figure 42. Sort Columns dialog box (IQCPS020)

You can sort all columns in ascending (A) or descending (D) order and further place those columns in order from column 1 to 6.

When you are finished with the Sort Columns dialog box, press Enter to cause the sort to be performed, or press the End key to return to a display of the Query panel.

Reload DSN dialog

Use the Reload DSN dialog to reinsert messages that were removed from the message queues.

To display the Reload DSN dialog, choose option **3**, Load on the IMS Queue Control Facility Main Menu panel. Use the Reload DSN dialog to reinsert removed queued messages.

Enter the name of the data set that contains the messages that will be reloaded.

```
| IQCP34U                  Reload DSN                                 |
| Command ===>                                  Scroll ===> PAGE      |
|                                                                     |
| Press ENTER to continue or END to exit.                             |
|                                            APAR . . : PH19814 20     |
| Server . . : QCFSERVE                                               |
| IMS ID . . : IMS1                                                   |
| QCF Func . : LOAD                                                   |
|                                                                     |
| Enter the data set name containing the messages.                    |
|                                                                     |
| _____                         |
|                                                                     |
|_____|
```

Figure 43. Reload DSN dialog box (IQCP34U)

For detailed field-level help, press the Help key with your cursor on the field in question.

When you finish entering the DSN on the dialog, press Enter to reload messages that were removed from the queue, panel IQCP34 is displayed.

```
--------------------------------------------------------------------------------
IQCP34                        Message Destinations              Row 1 to 5 of 5
Command ===>                                                    Scroll ===> PAGE

                                                 APAR . . : PH19814 2020/01/
Server . . : QCFSERV1                            JDTE . . : 2020.039
IMS ID . . : IMSA                                TIME . . : 17:36:12
QCF Func . : LOAD                                DATE . . : 2020/02/08
Select Row actions: A - Load  C - Copy  L - List  R - Reset
Then press ENTER to save them and END to start execution or to Exit.
                                 Struc/ Q   ------ Msgcnt -------
Act Destination name             Q     Typ Primry Secdry Total   Status
    TSUED01                      PRI   LT      1      0     1    _____
    DESRZA70.A7CICHBT DFSASYNC   LOC   AP                 48    _____
    IMSNET.L82MVS1  DFSASYNC     LOC   AP                 24    _____
    T0910582                     LOC   LT                 18    _____
    T0913000                     LOC   LT                 18    _____
```

Figure 44. Messages Destinations panel (IQCP34)

If you want to display the content of a message file, select option **3a** on the Main Menu.

```
--------------------------------------------------------------------------------
IPCP34W                       Message Destinations             Row 1 to 19 of 42
Command ===>                                                   Scroll ===> PAGE

                                                 APAR . . : PH19814 2020/01/
Server . . : QCFSERV                             JDTE . . : 2020.039
IMS ID . . : IMS1                                TIME . . : 17:39:03
QCF Func . : VIEW                                DATE . . : 2020/02/08
Select Row actions: L - List  then press ENTER to execute.
                                 Struc/ Q   ------ Msgcnt -------
Act Destination name             Q     Typ Primry Secdry Total
    TRAN31C0                     LOC   TR      8      0     8
    TRAN31V0                     LOC   TR      8      0     8
    T0910026                     LOC   LT      8      0     8
    TSUED01                      LOC   LT     36      0    36
    IMSUS06                      LOC   LT      8      0     8
    VTAG3138                     LOC   LT      4      4     8
    SEGUNDO                      LOC   LT      4      0     4
    VTKK4838                     LOC   LT      4      0     4
    L62TERM1                     LOC   LT      8      0     8
    T0913029                     LOC   LT      8      0     8
    VTAG2708                     LOC   LT      4      4     8
    TPX80002                     LOC   LT      4      0     4
    T0915604                     LOC   LT      4      4     8
    T0912054                     LOC   LT      8      0     8
    T3270LA                      LOC   LT      8      0     8
    IMSUS03                      LOC   LT      8      0     8
    T2167012                     LOC   LT      4      4     8
    VTAGB588                     LOC   LT      8      8    16
    T0914004                     LOC   LT      4      0     4
```

Figure 45. View Content of a Message File panel (IPCP34W)

For more information about how to use the View Content of a Message File panel, use the Help pull-down menu. For detailed field-level help, press the Help key with your cursor on the field in question.

Display Waited Tasks panel

Use the Display Waited Tasks panel to display a list of all tasks that were waited by the Queue Space Notification logic.

To display the Display Waited Tasks panel, select option 4- Wait on the Main Menu panel. On the Display Waited Tasks panel, you can take various actions (release, abort, stop, suspend, and list) on the displayed table or individual rows in the table.

The Display Waited Tasks panel shows a list of all tasks that were waited by the Queue Space Notification logic.

```
   View  Table_Actions  Help
───────────────────────────────────────────────────────────────────
IQCP94A                      Display Waited Tasks
Command ===>                                            Scroll ===> PAGE

Select a row action or press END to exit
                                          APAR . . : PH19814 2020/01/
Server . . : QCFSERVE                     JDTE . . : 2020.039
IMS ID . . : IMS1                         TIME . . : 15:47:42
QCF Func . : WAIT                         DATE . . : 2020/02/08
 Waited Tasks
Row actions: R-Release  A-Abort  X-Stop  S-Suspend

   -------- Task --------
A Type Identifier        Msg cnt DestType Insert Time    PCB     Status
***************************** Bottom of data ****************************
```

Figure 46. Display Waited Tasks panel (IQCP94A)

For more information about how to use the Display Waited Tasks panel, use the **Help** pull-down menu. For detailed field-level help, press the Help key with your cursor on the field in question.

When you are finished with the Display Waited Tasks panel, press the End key to return to the Main Menu.

You can choose the **View** pull-down menu to sort table columns.

```
   View  Table_Actions  Help
┌──────────────────────────┬────────────────────────────────────────
│   1. Sort table columns │ Display Waited Tasks
└──────────────────────────                     Scroll ===> PAGE
─────────────────────────────
Select a row action or press END to exit
                                          APAR . . : PH19814 2020/01/
Server . . : QCFSERVE                     JDTE . . : 2020.039
IMS ID . . : IMS1                         TIME . . : 15:50:40
QCF Func . : WAIT                         DATE . . : 2020/02/08
 Waited Tasks
Row actions: R-Release  A-Abort  X-Stop  S-Suspend

   -------- Task --------
A Type Identifier        Msg cnt DestType Insert Time    PCB     Status
***************************** Bottom of data ****************************
```

Figure 47. View pull-down menu on the Display Waited Tasks panel

If you choose option **1 - Sort table columns**, the following Sort Columns dialog box is displayed:

```
.----------------------------------------------------.
| IQCPS050           Sort Columns                    |
| Command ===>                                       |
|                                                    |
| Press ENTER to continue or END to exit.            |
|                    APAR : PH19814 2020/01/          |
|                                                    |
|                  A/D   Seq 1-6                      |
| Task Type . . . . . . _      _                     |
| Task ID . . . . . . . _      _                     |
| Secondary Msgcnt  . . _      _                     |
| Destination . . . . . _      _                     |
| Insert Time . . . . . _      _                     |
| PCB . . . . . . . . . _      _                     |
|                                                    |
|                                                    |
|                                                    |
|                                                    |
'----------------------------------------------------'
```

Figure 48. Sort Columns dialog box (IQCPS050)

For detailed field-level help on the Sort Columns dialog box, press the Help key with your cursor on the field in question.

After you make your specifications and press Enter, your sort is performed and the Display Waited Tasks panel is displayed again.

If you choose the **Table_Actions** pull-down menu from the Display Waited Tasks panel, a pull-down list is displayed as follows:

```
   View   Table_Actions   Help
 _____ _____  _____
 IQCP9 |    1. Release all displayed tasks | Tasks
 Comma |    2. Stop all displayed tasks    |                 Scroll ===> PAGE
       |_____|
 Select a row action or press END to exit
                                              APAR . . : PH19814 2020/01/
 Server . . : QCFSERVE                        JDTE . . : 2020.039
 IMS ID . . : IMS1                            TIME . . : 15:56:26
 QCF Func . : WAIT                            DATE . . : 2020/02/08
  Waited Tasks
 Row actions: R-Release  A-Abort  X-Stop  S-Suspend  L-List  U-Unlock

    -------- Task --------
 A Type Identifier        Msg cnt DestType Insert Time     PCB      Status
   BMP  QCFBATCH                0 T0910582 2020038/230223 ALTPCB   _____
 ***************************** Bottom of data *****************************
```

Figure 49. **Table_Actions** *pull-down menu on the Display Waited Tasks panel*

If you choose option **1 - Release all displayed tasks**, the following dialog box is displayed.

```
   View   Table_Actions   Help
 _____ _____  _____
 IQCP9 |    1. Release all displayed tasks | Tasks
 Comma |    2. Stop all displayed tasks    |                 Scroll ===> PAGE
       |_____|
 Select a row action or press END to exit
                                              APAR . . : PH19814 2020/01/
 Server . . : QCFSERVE                        JDTE . . : 2020.039
 IMS ID . . : IMS1                            TIME . . : 15:59:22
 QCF Func . : WAIT                            DATE . . : 2020/02/08
  Waited Tasks
 Row actions: R-Release  A-Abort  X-Stop  S-Suspend  L-List  U-Unlock

    -------- Task --------
 A Type Identifier        Msg cnt DestType Insert Time     PCB      Status
   BMP  QCFBATCH                0 T0910582 2020038/230223 ALTPCB   **Rlse**
 ***************************** Bottom of data *****************************
```

Figure 50. Release all displayed tasks dialog box

For detailed field-level help on the Release all displayed tasks result, press the Help key with your cursor on the field in question.

When you finish entering your choices on the Release all displayed tasks panel, press Enter to cause the task to be performed. When the task is completed, the Display Waited Tasks panel is displayed.

Note: Notice the changed status of **Rlse** on the Display Waited Tasks panel.

If you choose option **2 - Stop all displayed tasks**, the following dialog box is displayed.

```
   View  Table_Actions  Help
_____ _____ _____
IQCP9 |    1. Release all displayed tasks | Tasks
Comma |    2. Stop all displayed tasks    |             Scroll ===> PAGE
      |_____|
Select a row action or press END to exit
                                              APAR . . : PH19814 2020/01/
Server . . : QCFSERVE                         JDTE . . : 2020.039
IMS ID . . : IMS1                             TIME . . : 16:06:06
QCF Func . : WAIT                             DATE . . : 2020/02/08
 Waited Tasks
Row actions: R-Release  A-Abort  X-Stop  S-Suspend  L-List  U-Unlock

   -------- Task --------
A Type Identifier        Msg cnt DestType Insert Time    PCB    Status
  BMP  QCFBATCH               0 T0910582 2020038/230223 ALTPCB  STOP
****************************** Bottom of data ******************************
```

Figure 51. Stop all displayed tasks dialog box

For detailed field-level help on this dialog box, press the Help key with your cursor on the field in question.

When you finish entering your choices on the Stop all displayed tasks dialog, press Enter to cause the task to be performed. When the task is completed, the Display Waited Tasks panel is displayed.

Note: Notice the changed status of **STOP** on the Display Waited Tasks panel.

Queue Overflow Parameters sub-menu

Use the Queue Overflow Parameters sub-menu to edit the IQCQSNUN table and the queue overflow protection parameters for maintenance tasks.

To display the Queue Overflow Parameters Sub-Menu panel, select option **5 - Tables** on the Main Menu panel. You can use the Queue Overflow Parameters sub-menu panel to edit the IQCQSNUN table and the queue overflow protection parameters.

On the Queue Overflow Parameter Maintenance panel, you can select either of the following maintenance tasks:

- Maintenance of the main IQCQSNUN table
- Maintenance of the control statements used by the jobs that are started after thresholds are passed

Related reading: For more information about the tasks described on these panels, see Chapter 9, "Message queue overflow protection," on page 85. These topics describe the following tasks:

- Dividing the queue space into areas
- Understanding the flow of the queue space overflow protection logic (driven by the IQCQSNUN table)
- Understanding the content of the IQCQSNUN table

For an example of how to use this panel, see the example provided for option **6** under "Queue Space utilization notification parameters panel" on page 296. The information applies to option **5**, as well.

```
  Help
_____
IQCP70                  Queue Overflow Parameters Sub-Menu
Option ===>

SELECT an option or press END to exit.
                                            APAR . . : PH19814 2020/01/
Server . . : QCFSERVE                       JDTE . . : 2020.036
IMS ID . . : IMS2                           TIME . . : 12:12:26
                                            DATE . . : 2020/02/05
 Queue Overflow Protection Parameter Maintenance

     Committed and Uncommitted Messages Parameters
  1   Select queue space notification table

     Committed Messages Parameters
  2   Select AREA A-B invoked command processing
  3   Select AREA B-C invoked command processing
  4   Select AREA C-D invoked command processing
  5   User-set Threshold

 Tools for queue overflow parameters

 6 Space utilization notification testing tool
```

Figure 52. Queue Overflow Protection Parameters Sub-Menu panel (IQCP70)

For more information about how to use the Queue Overflow Parameters Sub-Menu panel, use the Help pull-down menu. For detailed field-level help, press the Help key with your cursor on the field in question.

When you are finished with the Queue Overflow Parameters Sub-Menu panel, press the End key to return to the Main Menu panel.

Queue Space Notification Tables panel

Use Queue Space Notification Tables panel to edit queue space notification tables and to view the current QSN table that controls the queue space overflow protection logic.

To display the Queue Space Notification Tables panel, select option **1 - Select queue space notification table** on the Queue Overflow Parameters Sub-Menu panel. Use the Queue Space Notification Tables panel to edit queue space notification tables and to view the current QSN table that controls the queue space overflow protection logic.

```
  Help
_____
IQCP71A                 Queue Space Notification Tables
Option ===> _____

Select an option or press END to exit.
                                            APAR . . : PH19814 2020/01/
Server . . : QCFSERVE                       JDTE . . : 2020.039
IMS ID . . : IMS3                           TIME . . : 10:51:29
                                            DATE . . : 2020/02/08
 QSN Table Processing for committed and uncommitted messages

     Committed and Uncommitted Messages Processing
  1   Edit queue space notification tables
  2   View the current QSN table used by IMS
```

Figure 53. Queue Space Notification Tables panel (IQCP71A)

For more information about how to use the Queue Space Notification Tables panel, use the Help pull-down menu. For detailed field-level help, press the Help key with your cursor on the field in question.

QSN Table Names panel

Use the QSN Table Names panel display a list of all tables that you created and saved for copy, edit, or delete actions.

To display the QSN Table Names panel, select option **1 - EDIT** on the Queue Space Notification Tables panel. The Member List of QSN Table Names panel shows a list of all tables that you created and saved (panel IQCP71M). On this panel, you can take actions on the tables that are displayed in the panel.

These tables can have different parameters and can be used at different times. The current table name used by IMS, IQCQSNUN, is shown with the description, **CURRENT OVERFLOW PARAMETERS TABLE**, next to it.

You can copy, edit, or delete each of these tables. Copy prompts you for a new unique name in the **New Member** field.

Each time you edit and save the current table, a panel prompts you to reinsert the table to IMS. If you want to send one of the other tables to IMS, you must copy the other table to the default name.

If you do not have a table with the IQCQSNUN name, a message warns you of the missing default name (see), and IQCQSNUN will be shown in the **New Member** field and, after pressing Enter, IQCQSNUN will be restored as the current table (see).

```
  Help
_____
IQCP71M                        QSN Table Names              Row 1 to 1 of 1
Command ===> _____         Scroll ===> PAGE

SELECT a row action and press ENTER to process or END to exit.

                                            APAR . . : PH19814 2020/01/
Server . . : QCFSERVE                       JDTE . . : 2020.036
IMS ID . . : IMS3                           TIME . . : 12:10:15
                                            DATE . . : 2020/02/05


Members in . . . . : IQC.ECDV255.IMS3.UNCOMMIT
New member name  . . _____

Row options: Copy(C), Edit(E), Delete(D)

Action  Member Name  New Member  PDS description
e _     IQCQSNUN     _____    CURRENT OVERFLOW PARAMETERS TABLE
***************************** Bottom of data ********************************
```

Figure 54. QSN Table Names panel (IQCP71M)

For more information about how to use the QSN Table Names panel, use the **Help** pull-down menu. For detailed field-level help, press the Help key with your cursor on the field in question.

Queue Threshold parameters panel

Use the Queue Threshold parameters panel to display your committed and uncommitted message thresholds and your jobs or procedures to be called at threshold crossover.

To display the Queue Threshold parameters panel, select row action **E** on the QSN Table Names panel (IQCP71M).

```
   Help
  _____
IQCP71BA                    Queue Threshold parameters
Command ===> _____

Press ENTER to continue or END to exit.          APAR . . : PH19814 2020/01/
  Server . . : QCFSERVE                           Jdate  . : 2020.036
  IMS ID . . : IMS3                               Time . . : 12:32:19
  Member . . : IQCQSNUN                           Date . . : 2020/02/05

 Committed and uncommitted messages: Thresholds.
 Queue  upper  threshold  . . . 075  Queue  lower  threshold . . . . 060
 Queue IQC6101 threshold  . . . 085

 Committed messages: Job/Procedure names to be called at threshold crossover.
 Ignore application calls inserting messages to express PCBs . . . . Y

 Jobnames  - A-B:  . . . . NOUNLOAD  B-C: . . . NOUNLOAD  C-D: . . . NOUNLOAD
 Procnames - A-B:  . . . .           B-C: . . .           C-D: . . .

 Jobnames  - IQC6101  . . NOUNLOAD
 Procname  - IQC6101  . . _____
```

Figure 55. Queue Threshold parameters panel (IQCP71BA)

When you are finished editing this panel, press Enter, and the next panel (IQCP71B) will be displayed.

For more information about how to use the Queue Threshold parameters panel, use the Help pull-down menu. For detailed field-level help, press the Help key with your cursor on the field in question.

QSN Table Maintenance panel for uncommitted messages

Use the QSN Table Maintenance panel for uncommitted messages to display each AREA status started or stopped for which you can enter a set of percentages or actions.

To display the QSN Table Maintenance panel, when you have finished entering information on the Queue Threshold parameters panel, press Enter.

```
   Help
  _____
IQCP71B                        QSN Table Maintenance
Command ===>

Press ENTER to continue or END to exit.          APAR . . : PH19814 2020/01/
  Server . . : QCFSERVE                           Jdate  . : 2020.026
  IMS ID . . : IMS1                               Time . . : 16:03:41
                                                  Date . . : 2020/01/26

 Uncommitted messages: Modify QSN entries using the AREAs "ALL" Callers TYPE.
For each AREA status Started/Stopped enter a set of Percent/Action(N, S, W).

A: Start . . 100 N  Stop . . 100 N     B: Start . . 050 W  Stop . . 050 W
C: Start . . 008 W  Stop . . 005 W     D: Start . . 008 S  Stop . . 005 S
                                                      More:
 Uncommitted messages: Modify individual entries in the QSN table.
For each Caller status Started/Stopped enter a set of Percent/Action(N, S, W).

      ___TYP0____ ___APPL____ ___APPC____ ____DC_____ ___OTMA____ ___MSC_____
AREA Start  Stop Start  Stop Start  Stop Start  Stop Start  Stop Start  Stop
A    100 N 100 N 100 N 100 N 100 N 100 N 100 N 100 N 100 N 100 N 100 N 100 N
B    050 W 050 W 050 W 050 W 050 W 050 W 050 W 050 W 050 W 050 W 050 W 050 W
C    008 W 005 W 008 W 005 W 008 W 005 W 008 W 005 W 008 W 005 W 008 W 005 W
D    008 S 005 S 008 S 005 S 008 S 005 S 008 S 005 S 008 S 005 S 008 S 005 S
```

Figure 56. QSN Table Maintenance panel (IQCP71B)

For more information about how to use the QSN Table Maintenance panel, use the **Help** pull-down menu. For detailed field-level help, press the Help key with your cursor on the field in question.

Update Confirmation dialog

Use the Update Confirmation dialog to confirm that you really want to update the IQCQSNUN default table.

When you are finished entering information on the QSN Table Maintenance panel and you press Enter, the Update Confirmation dialog box is displayed.

```
 ┌──────────────────────────────────────────────────────────────┐
 │ IQCP72UW          Update Confirmation                        │
 │ Command ===> _____ Scroll ===> PAGE      │
 │                                                              │
 │ Press ENTER to continue or END to exit.                     │
 │                      APAR : PH19814 2020/01/                │
 │ Server . . . . . . : QCFSERVE                               │
 │ IMS ID . . . . . . : IMS3                                   │
 │ Default table . . : IQCQSNUN                                │
 │                                                              │
 │ Warning:  . . . N   Do you really want to update the        │
 │                     Member containing the values            │
 │                     currently in use: IQCQSNUN ?            │
 │                     Enter Y to confirm update.              │
 │                                                              │
 │                                                              │
 │                                                              │
 │                                                              │
 │ ─────────────────────────────────────────────────────────── │
 └──────────────────────────────────────────────────────────────┘
```

Figure 57. Update Confirmation dialog box (IQCP72UW)

For detailed field-level help on the Update Confirmation dialog box, press the Help key with your cursor on the field in question.

QSN Table Maintenance for committed and uncommitted messages

To display the QSN Table Maintenance panel for committed and uncommitted messages, select option **2 - View the current QSN table used by IMS** from the Queue Space Notification Tables panel (IQCP71A).

```
   Help
 ──────────────────────────────────────────────────────────────────────────
 IQCP71V                    QSN Table Maintenance
 Command ===>

 Press ENTER to continue or END to exit.          APAR . . : PH19814 2020/01/
   Server . . : QCFSERVE                          Jdate  . : 2020.039
   IMS ID . . : IMS1                              Time . . : 16:05:26
   Member . . : IQCQSNUN                          Date . . : 2020/02/08

  Committed and uncommitted messages: Thresholds.
  075  Upper       060  Lower        085  IQC6101     Y    Express PCBs
                                                             More:
  Uncommitted messages:  QSN table.
 For each Caller statuses: Started/Stopped a set Percent/Action is shown.

       ___TYP0_____ ___APPL_____ ___APPC_____ ____DC_____ ___OTMA_____ ___MSC_____
 Area Start  Stop Start  Stop Start  Stop Start  Stop Start  Stop Start  Stop
 A    100 N 100 N 100 N 100 N 100 N 100 N 100 N 100 N 100 N 100 N 100 N 100 N
 B    050 W 050 W 050 W 050 W 050 W 050 W 050 W 050 W 050 W 050 W 050 W 050 W
 C    008 W 005 W 008 W 005 W 008 W 005 W 008 W 005 W 008 W 005 W 008 W 005 W
 D    008 S 005 S 008 S 005 S 008 S 005 S 008 S 005 S 008 S 005 S 008 S 005 S

  Committed messages: Job/Procedure names to be called at threshold crossover.
  IQCPRCAB  A-B  IQCPRCBC  B-C  IQCPRCCD  C-D  IQCPRCUT  IQC6101 - Jobnames
            A-B            B-C            C-D            IQC6101 - Procedures
```

Figure 58. QSN Table Maintenance (IQCP71V)

For more information about how to use this QSN Table Maintenance panel, use the **Help** pull-down menu. For detailed field-level help press the Help key with your cursor on the field in question.

When you are finished with this QSN Table Maintenance panel, press the End key.

The Queue Overflow Parameters Sub-Menu panel is displayed, as follows:

```
   Help
 _____
 IQCP70                   Queue Overflow Parameters Sub-Menu
 Option ===>

 Press ENTER to continue or END to exit.
                                                 APAR . . : PH19814 2020/01/
 Server . . : QCFSERVE                           JDTE . . : 2020.039
 IMS ID . . : IMS1                               TIME . . : 16:09:29
                                                 DATE . . : 2020/02/08
  Queue Overflow Protection Parameter Maintenance

     Uncommitted Messages Parameters
  1   Select queue space notification table

     Committed Messages Parameters
  2   Select AREA A-B invoked command processing
  3   Select AREA B-C invoked command processing
  4   Select AREA C-D invoked command processing
  5   User-set Threshold
```

Figure 59. Queue Overflow Parameters Sub-Menu panel (IQCP70)

For more information about how to use the Queue Overflow Parameters Sub-Menu panel, use the **Help**
pull-down menu. For detailed field-level help press the Help key with your cursor on the field in question.

When you choose option **2 - Select AREA A-B invoked command processing** from the Queue Overflow
Parameters Sub-Menu panel, the QSN Table Names panel is displayed:

```
   Help
 _____
 IQCP72M                       QSN Table Names                 Row 1 to 1 of 1
 Command ===>                                              Scroll ===> PAGE

 Select a row action or press END to exit
                                                 APAR . . : PH19814 2020/01/
 Server . . : QCFSERVE                           JDTE . . : 2020.030
 IMS ID . . : IMS1                               TIME . . : 16:10:14
                                                 DATE . . : 2020/02/08
 Members in . . . . : IQC.ECDV255.IMS1.COMMIT
 New member name  . .          Area Threshold . : AB
 Row actions: C - Copy  E - Edit  D - Delete

 Action  Member Name  New Member  Description
        IQCQSNAB                  DEFAULT CONTROL CARDS MEMBER
 ***************************** Bottom of data *****************************
```

Figure 60. QSN Table Names panel (IQCP72M)

For more information about how to use the QSN Table Names panel, use the **Help** pull-down menu. For
detailed field-level help, press the Help key with your cursor on the field in question.

As an example, if you choose option **E - Edit** on the QSN Table Names panel, the Edit Control Statements
panel is displayed:

```
   Help
_____
IQCP72A                    Edit Control Statements              Row 1 to 3 of 7
Command ===>                                              Scroll ===> PAGE

Select a row action or press END to exit
                                                  APAR . . : PH19814 2020/01/
Server . . : QCFSERVE                             JDTE . . : 2020.039
IMS ID . . : IMS1                                 TIME . . : 16:11:06
QCF Func . : TABLES                               DATE . . : 2020/02/08
 Create INCLUDE/EXCLUDE parms for automated committed messages
Crossed Threshold  . : AB        Member . . : IQCQSNAB
TOIPARMS XCFGROUP   . : TOINAME1  Msgs for . : T0913000
DSN  . . . : IQC.T0054370.BROWSE

Confirm update?  . . _  Y

Row actions: E - Edit  C - Copy  D - Delete I - Insert

   QSN      Tp IN - Queue -   -- Destination --- - Queue - ---- Source ----
A Action    Ds EX Type   Age QType Name        %   Cnt   QType Name
   BROWSE   10 IN ALL    __  _____ _____ __  _____ _____ _____
   UNLOAD   __ IN ALL    __  MSC   _____ 10  _____ _____ _____
   UNLOAD   __ IN ALL    __  VSP   _____ 10  _____ _____ _____
```

Figure 61. Edit Control Statements panel (IQCP72A)

For more information about how to use the Edit Control Statements panel, use the **Help** pull-down menu. For detailed field-level help, press the Help key with your cursor on the field in question.

As an example, if you choose option **E - Edit** from the Edit Control Statements panel, the Edit Requirements for a Set of Messages panel is displayed. The entire panel is shown here for simplicity; you must actually press your FORWARD key to see the last few lines of this panel.

```
   Help
_____
IQCP72B            Edit Requirements for a Set of Messages
Command ===>

Press ENTER to continue or END to exit.
                                                  APAR . . : PH19814 2020/01/
Server . . : QCFSERVE                             JDTE . . : 2020.039
IMS ID . . : IMS1                                 TIME . . : 16:15:45
                                                  DATE . . : 2020/02/08
 Create threshold parms
Action . . . BROWSE      Q Type . . ALL          Filter . . IN
Top Dest . . 10          Age  . . .
                                                             More:
 With Filter=IN
Queue percent  . .                   Queue Rcount . . . .
 With Filter=IN and Queue Percent/Queue Rcount
 With Filter=IN/EX and no queue percent/Queue Rcount
Dest Type  . . . .                   Sidename . . . . . .
DEST NAME  . . . .                   Dest TMname  . . . .
Dest LUname  . . .
 With Filter=EX and no Queue Percent/Queue Rcount
Dest SYSID . . . .                   LU 6.2 name  . . . .
Srce Type  . . . .                   Network ID  . . .
Source name  . . .                   Dest NetID  . . .
 Source SYSID  . .                   Source LUname . .
Tmember name . . .                   Source NetID  . .
 Source TMname . .                   TP name . . . . .
Tpipe name . . . .                VTAM LUname . . . .
 IMSID  . . . . . .
```

Figure 62. Edit Requirements for a Set of Messages panel (IQCP72B)

For more information about how to use the Edit Requirements for a Set of Messages panel, use the **Help** pull-down menu. For detailed field-level help, press the Help key with your cursor on the field in question.

Then, if you choose option **3 - Select AREA B-C invoked command processing** from the Queue Overflow Parameters Sub-Menu panel, the QSN Table Names panel is displayed:

```
   Help
_____
IQCP72M                        QSN Table Names                   Row 1 to 1 of 1
Command ===>                                              Scroll ===> PAGE

Select a row action or press END to exit
                                                  APAR . . : PH19814 2020/01/
Server . . : QCFSERVE                             JDTE . . : 2020.039
IMS ID . . : IMS1                                 TIME . . : 16:18:50
                                                  DATE . . : 2020/02/08
Members in . . . . . : IQC.ECDV255.IMS1.COMMIT
New member name  . .          Area Threshold . : BC
Row actions: C - Copy  E - Edit  D - Delete

Action  Member Name  New Member  Description
        IQCQSNBC                 ***   DEFAULT TABLE   ***
****************************** Bottom of data *******************************
```

Figure 63. QSN Table Names panel (IQCP72M)

For more information about how to use the QSN Table Names panel, use the **Help** pull-down menu. For detailed field-level help, press the Help key with your cursor on the field in question.

Then, if you choose option **E - Edit** from the QSN Table Names panel, the Edit Control Statements panel is displayed:

```
   Help
_____
IQCP72A                     Edit Control Statements            Row 1 to 3 of 7
Command ===>                                              Scroll ===> PAGE

Select a row action or press END to exit
                                                  APAR . . : PH19814 2020/01/
Server . . : QCFSERVE                             JDTE . . : 2020.039
IMS ID . . : IMS1                                 TIME . . : 16:22:50
QCF Func . : TABLES                               DATE . . : 2020/02/08
 Create INCLUDE/EXCLUDE parms for automated committed messages
Crossed Threshold  . : BC        Member . . : IQCQSNBC
TOIPARMS XCFGROUP  . : TOINAME1  Msgs for . : T0913000
DSN  . . . : IQC.T0054370.BROWSE

Confirm update?  . . _  Y

Row actions: E - Edit  C - Copy  D - Delete I - Insert

  QSN       Tp IN - Queue -  -- Destination --- - Queue - ---- Source ----
A Action   Ds EX Type   Age QType Name       %  Cnt  QType Name
  BROWSE   10 IN ALL    __  _____ _____ __ _____ _____ _____
  UNLOAD   __ IN ALL    __  MSC   _____ 5  _____ _____ _____
  UNLOAD   __ IN ALL    __  VSP   _____ 5  _____ _____ _____
```

Figure 64. Edit Control Statements panel (IQCP72A)

For more information about how to use the Edit Control Statements panel, use the **Help** pull-down menu. For detailed field-level help, press the Help key with your cursor on the field in question.

Then, if you choose option **I - Insert** from the Edit Control Statements panel, the Edit Requirements for a Set of Messages panel is displayed. The entire panel is shown here for simplicity; you must actually press your FORWARD key to see the last few lines of the panel.

```
   Help
_____
IQCP72B            Edit Requirements for a Set of Messages
Command ===>

Press ENTER to continue or END to exit.
                                                APAR . . : PH19814 2020/01/
Server . . : QCFSERVE                           JDTE . . : 2020.039
IMS ID . . : IMS1                               TIME . . : 16:26:19
                                                DATE . . : 2020/02/08
 Create threshold parms
Action . . .              Q Type . .             Filter . .
Top Dest . .              Age  . . .
                                                          More:
 With Filter=IN
Queue percent  . .                    Queue Rcount . . . .
 With Filter=IN and Queue Percent/Queue Rcount
 With Filter=IN/EX and no queue percent/Queue Rcount
Dest Type  . . . .                    Sidename . . . . . .
DEST NAME  . . . .                    Dest TMname  . . . .
Dest LUname  . . .
 With Filter=EX and no Queue Percent/Queue Rcount
Dest SYSID . . . .                 LU 6.2 name  . . . .
Srce Type  . . . .                    Network ID  . . . .
Source name  . . .                   Dest NetID  . . . .
 Source SYSID  . .                   Source LUname . .
Tmember name . . .                   Source NetID  . .
 Source TMname . .                   TP name . . . . .
Tpipe name . . . .                 VTAM LUname . . . .
IMSID  . . . . . .
```

Figure 65. Edit Requirements for a Set of Messages panel (IQCP72B)

Queue Space utilization notification parameters panel

Use the Queue Space utilization notification parameters panel to choose to either edit queue space utilization notification parameters or view the current space utilization notification parameters.

To display the Queue Space utilization notification parameters panel, choose option **6 - Notify** from the Main Menu.

```
   Help
_____
IQCP71AS        Queue Space utilization notification parameters
Option ===>

Select an option or press END to exit.
                                                APAR . . : PH19814 2020/01/
Server . . : QCFSERVE                           JDTE . . : 2020.039
IMS ID . . : IMS1                               TIME . . : 10:25:47
                                                DATE . . : 2020/02/08
 Queue space utilization notification parameters

 1   Edit queue space utilization notification parameters
 2   View the current space utilization notification parameters
```

Figure 66. Queue Space utilization notification parameters panel (IQCP71AS)

For more information about how to use the Queue Space utilization notification parameters panel, use the **Help** pull-down menu. For detailed field-level help, press the Help key with your cursor on the field in question.

Then, if you choose option **1 -Edit queue space utilization notification parameters** from the Queue Space utilization notification parameters panel, the Queue space utilization notification parameters table panel (IQCP71MS) is displayed.

```
    Help

 _____
IQCP71MS    Queue space utilization notification para    member deleted
Command ===> _____     Scroll ===> PAGE

Press ENTER to continue or END to exit.
                                                  APAR . . : PH19814 2020/01/
Server . . : QCFSERVE                             JDTE . . : 2020.039
IMS ID . . : IMS1                                 TIME . . : 11:55:15
                                                  DATE . . : 2020/02/08
Members in . . . . : IQC.ECDV255.IMS1.NOTPARMS
New member name  . .
_____
Row options: Copy(C), Edit(E), Delete(D)

Action  Member Name  New Member  PDS description
  _        IQCQSUNP    _____   CURRENT OVERFLOW PARAMETERS TABLE
****************************** Bottom of data ******************************
```

Figure 67. Queue space utilization notification parameters table panel (IQCP71MS)

For more information about how to use the Queue Space utilization notification parameters table panel, use the **Help** pull-down menu. For detailed field-level help, press the Help key with your cursor on the field in question.

Then, if you choose option **E - Edit** on the Queue space utilization notification parameters table panel, the Queue Threshold parameters panel (IQCP71BA) is displayed.

```
   Help

 _____
IQCP71BA                   Queue Threshold parameters
Command ===> _____

Press ENTER to continue or END to exit.        APAR . . : PH19814 2020/01/
  Server . . : QCFSERVE                         Jdate  . : 2020.039
  IMS ID . . : IMS1                             Time . . : 11:56:23
  Member . . : IQCQSUNP                         Date . . : 2020/02/08

 Committed messages: Thresholds.
 Queue  upper  threshold  . . . 075  Queue  lower  threshold . . . . 060
 Queue IQC6101 threshold  . . . 085

 Committed messages: Job/Procedure names to be called at threshold crossover.
 Ignore application calls inserting messages to express PCBs . . . . Y

 Jobnames  - A-B:  . . . . NOUNLOAD  B-C: . . . NOUNLOAD  C-D: . . . NOUNLOAD
 Procnames - A-B:  . . . .           B-C: . . .           C-D: . . . _____

 Jobnames   - IQC6101  . . NOUNLOAD                       ____
 Procedures - IQC6101  . . _____                        ____
```

Figure 68. Queue Threshold parameters panel (IQCP71BA)

For more information about how to use the Queue Threshold parameters panel, use the Help pull-down menu. For detailed field-level help, press the Help key with your cursor on the field in question.

When you are finished entering information on the Queue Threshold parameters panel, if you press Enter, the Queue Overflow Notification Parameters panel (IQCP91JX) is displayed. The entire panel is shown here for simplicity; you must actually press your FORWARD key to see the last few lines of the panel.

```
  Help
 ------------------------------------------------------------------------------
IQCP91JX             Queue Overflow Notification Parameters
Command ===>                                                Scroll ===> PAGE

Select a row action or press ENTER to continie     APAR . . : PH19814 2020/01/
Server . . : IQCSERV                               JDTE . . : 2020.039
IMS ID . . : SYS3                                  TIME . . : 12:28:33
                                                   DATE . . : 2020/02/08
Select Source/Destination types with a "/"
Culprit and other actions: A - ABEND, N - NONE, O - WTO, S - STOP, W - WAIT
                         J - STOP THE SOURCE
Enter S to work with Notification Jobs Lists for each AREA.
 Area      Percent  _Source/Destination Types_  _Culprit Act __Other Act_ Not
 Name      Tot Usd  ALL APPC APPL  DC MSC OTMA Strtd  Stppd Strtd  Stppd Lists
                                                                    More:     +
AREA0001  20  50    /                            N       O    N       O     S
AREA0002  25  45    /                            N       O    N       O     _
AREA0003  30  40    /                            N       O    N       O     _
AREA0004  40  30    /                            N       O    N       O     _
--------  --  --    _                            _       _    _       _     _
--------  --  --    _                            _       _    _       _     _
--------  --  --    _                            _       _    _       _     _
--------  --  --    _                            _       _    _       _     _
--------  --  --    _                            _       _    _       _     _
--------  --  --    _                            _       _    _       _     _
```

Figure 69. Queue Overflow Notification Parameters panel (IQCP91JX)

Press Enter to leave this panel.

```
  Help
 ------------------------------------------------------------------------------
IQCP91JY             Queue Overflow Notification Parameters
Command ===>                                                Scroll ===> PAGE

ENTER to continue, END to go to previous panel.    APAR . . : PH19814 2020/01/
Server . . : IQCSERV                               JDTE . . : 2020.039
IMS ID . . : SYS3                                  TIME . . : 13:00:30
                                                   DATE . . : 2020/02/08
 Select percent queue utilization and action for each FAILSAFE statement.
 Actions: A - ABEND, N - NONE, O - WTO, S - STOP, W - WAIT
 Enter S to work with notification Jobs Lists for each FAILSAFE.
                                                            More:     +
                         PERCENT
FAILSAFE ID    PERCENT   ALLOWED       ACTION    NOT LISTS
FAIL0001       65        05            O         s
FAIL0002       75        02            W             -
--------       --        --            _             -
--------       --        --            _             -
--------       --        --            _             -
--------       --        --            _             -
--------       --        --            _             -
--------       --        --            _             -
--------       --        --            _             -
--------       --        --            _             -
```

Figure 70. Queue Overflow Notification Parameters panel (IQCP91JY)

For more information about how to use panel IQCP91JY, use the **Help** pull-down menu. For detailed field-level help, press the Help key with your cursor on the field in question.

Detailed use of panel IQCP71MS

Use panel IQCP71MS to copy, add or change default overflow parameters table.

The following sequence provides more detail about how to use this panel.

You can copy each table as demonstrated in the following figure:

```
   Help
_____
IQCP71MS    Queue space utilization notification parameters ta Row 1 to 1 of 1
Command ===> _____     Scroll ===> PAGE

Press ENTER to continue or END to exit.
                                             APAR . . : PH19814 2020/01/
Server . . : QCFSERVE                        JDTE . . : 2020.036
IMS ID . . : IMS1                            TIME . . : 11:55:15
                                             DATE . . : 2020/02/05
Members in . . . . : IQC.ECDV255.IMS1.NOTPARMS
New member name  . .
_____

Row options: Copy(C), Edit(E), Delete(D)

Action  Member Name  New Member  PDS description
c       IQCQSUNP     COPY0001 _  CURRENT OVERFLOW PARAMETERS TABLE
***************************** Bottom of data ******************************
```

Figure 71. Queue space utilization notification parameters table panel (IQCP71MS)

Pressing Enter will complete the Copy as shown in IQCP1MS:

```
   Help
_____
IQCP71MS    Queue space utilization notification parameters ta Row 1 to 2 of 2
Command ===> _____     Scroll ===> PAGE

Press ENTER to continue or END to exit.
                                             APAR . . : PH19814 2020/01/
Server . . : QCFSERVE                        JDTE . . : 2020.042
IMS ID . . : IMS1                            TIME . . : 15:28:57
                                             DATE . . : 2020/02/11
Members in . . . . : IQC.ECDV255.IMS1.NOTPARMS
New member name  . . _____

Row options: Copy(C), Edit(E), Delete(D)

Action  Member Name  New Member  PDS description
        COPY0001     _____
_       IQCQSUNP     _____   CURRENT OVERFLOW PARAMETERS TABLE
***************************** Bottom of data ******************************


  _____
 |                                                                        |
 | IQC1713P member COPY0001 added in data set IQC ECDV255.IMS1.NOTPARMS |
 |_____|
```

Figure 72. Queue space utilization notification parameters table panel (IQCP71MS)

You can also create a new member by using the **New member name** field, as shown in panel IQCP71MS:

```
   Help
_____
IQCP71MS    Queue space utilizion notification parameters ta  Row 1 to 2 of 2
Command ===> _____       Scroll ===> PAGE

Press ENTER to continue or END to exit.
                                            APAR . . : PH19814 2020/01/
Server . . : QCFSERVE                       JDTE . . : 2020.042
IMS ID . . : IMS1                           TIME . . : 15:30:30
                                            DATE . . : 2020/02/11
Members in . . . . : IQC.ECDV255.IMS1.NOTPARMS
New member name  . . Newmem01

Row options: Copy(C), Edit(E), Delete(D)

Action  Member Name  New Member  PDS description
_       COPY0001     _____
_       IQCQSUNP     _____      CURRENT OVERFLOW PARAMETERS TABLE
***************************** Bottom of data ******************************
```

Figure 73. Queue space utilization notification parameters table panel (IQCP71MS)

Pressing Enter will prompt you for updates to the values currently being used by IMS, as shown in panel IQC71BC:

```
   Help
_____
IQCP71BC    Queue space utilization notification parameters
Command ===> _____       Scroll ===> PAGE

Press ENTER to continue or END to exit.
                                            APAR . . : PH19814 2020/01/
Server . . : QCFSERVE                       JDTE . . : 2020.042
IMS ID . . : IMS1                           TIME . . : 15:32:19
Member . . : NEWMEM01                       DATE . . : 2020/02/11

Committed messages   Thresholds
Queue   upper  threshold . . . 075  Queue upper threshold . . . 080
Queue location threshold . . . 085

Committed messages  Job/Procedure names to be called at threshold crossover
Ignore application calls inserting messages to express PCB . . . . . . Y

Jobnames - A-B . . . NOUNLOAD  B-C . . . NOUNLOAD  C-D . . . NOUNLOAD
Procnames - A-B . . . _____  B-C . . . _____  C-D . . . _____

  Jobname - IQC6101 . .NOUNLOAD
Procnames - IQC6101 . .NOUNLOAD
```

Figure 74. Queue Threshold Parameters panel (IQCP71BC)

The next panel to be displayed is IQCP91JX:

```
   Help
--------------------------------------------------------------------------------
IQCP91JX           Queue Overflow Notification Parameters
Command ===>                                               Scroll ===> PAGE

Select a row action or press ENTER to continie    APAR . . : PH19814 2020/01/
Server . . : IQCSERV                              JDTE . . : 2020.039
IMS ID . . : SYS3                                 TIME . . : 12:28:33
                                                  DATE . . : 2020/02/08
Select Source/Destination types with a "/"
Culprit and other actions: A - ABEND, N - NONE, O - WTO, S - STOP, W - WAIT
                      J - STOP THE SOURCE
Enter S to work with Notification Jobs Lists for each AREA.
 Area      Percent _Source/Destination Types_ _Culprit Act __Other Act_ Not
 Name      Tot Usd ALL APPC APPL  DC MSC OTMA Strtd  Stppd Strtd  Stppd Lists
                                                                    More:     +
AREA0001  20  50    /                          N      O     N      O      S
AREA0002  25  45    /                          N      O     N      O      _
AREA0003  30  40    /                          N      O     N      O      _
AREA0004  40  30    /                          N      O     N      O      _
--------  --  --    _                          _      _     _      _      _
--------  --  --    _                          _      _     _      _      _
--------  --  --    _                          _      _     _      _      _
--------  --  --    _                          _      _     _      _      _
--------  --  --    _                          _      _     _      _      _
--------  --  --    _                          _      _     _      _      _
```

Figure 75. Queue Overflow Notification Parameters panel (IQCP91JX)

The next panel to be displayed is IQCP91JY:

```
  Help
--------------------------------------------------------------------------------
IQCP91JY           Queue Overflow Notification Parameters
Command ===>                                               Scroll ===> PAGE

ENTER to continue, END to go to previous panel.   APAR . . : PH19814 2020/01/
Server . . : IQCSERV                              JDTE . . : 2020.039
IMS ID . . : SYS3                                 TIME . . : 13:00:30
                                                  DATE . . : 2020/02/08
 Select percent queue utilization and action for each FAILSAFE statement.
 Actions: A - ABEND, N - NONE, O - WTO, S - STOP, W - WAIT
 Enter S to work with notification Jobs Lists for each FAILSAFE.
                                                            More:     +
                        PERCENT
FAILSAFE ID    PERCENT  ALLOWED      ACTION    NOT LISTS
FAIL0001       65       05           O         s
FAIL0002       75       02           W           -
--------       --       --           -         -
--------       --       --           -         -
--------       --       --           -         -
--------       --       --           -         -
--------       --       --           -         -
--------       --       --           -         -
--------       --       --           -         -
--------       --       --           -         -
```

Figure 76. Queue Overflow Notification Parameters panel (IQCP91JY)

Tips:

• You can edit the notification job list entries for the selected AREA or FAILSAFE by entering S. If there are
  no notification job list entries, an empty list is shown and you have the option to create the first entry.
  Notification job list entries can be copied, deleted, edited, or inserted. Help panels are provided that
  contain more information.

• Changes made to the NOTIFY parameters by using the TSO client are valid (like all other changes
  to overflow protection parameters) only for the current IMS run. On IMS start, the control region
  customization member is applied.

Pressing Enter causes IQCP71MS to be displayed, as shown in the following figure:

```
   Help
 _____
IQCP71MS    Queue space utilizion notification parameters ta Row 1 to 3 of 3
Command ===> _____      Scroll ===> PAGE

Press ENTER to continue or END to exit.
                                               APAR . . : PH19814 2020/01/
Server . . : QCFSERVE                          JDTE . . : 2020.042
IMS ID . . : IMS1                              TIME . . : 15:37:28
                                               DATE . . : 2020/02/11
Members in . . . . . : IQC.ECDV255.IMS1.NOTPARMS
New member name  . . _____

Row options: Copy(C), Edit(E), Delete(D)

Action   Member Name   New Member   PDS description
_        COPY0001      _____
_        IQCQSUNP      _____       CURRENT OVERFLOW PARAMETERS TABLE
_        NEWMEM01      _____
***************************** Bottom of data *******************************


   _____
  |                                                                     |
  | IQC1713K member NEWMEM01 added in data set IQC ECDV255.IMS1.NOTPARMS |
  |_____|
```

Figure 77. Queue space utilization notification parameters table panel (IQCP71MS)

Any new members and be edited, copied or deleted. You can also make any or the members the current overflow parameter table used by IMS. To do this, you must move the contents of the member to IQCQSNUP; this is the only member that IMS Queue Control Facility sends to IMS.

The sequence of panels that follow shows you how to make a member the current overflow parameter table. Before you begin, it is recommended that you copy the contents of IQCQSUNP to a new member name.

The first panel in the sequence is IQCP71MS:

```
   Help
 _____
IQCP71MS    Queue space utilization notification parameters ta Row 1 to 3 of 3
Command ===> _____      Scroll ===> PAGE

Press ENTER to continue or END to exit.
                                               APAR . . : PH19814 2020/01/
Server . . : QCFSERVE                          JDTE . . : 2020.042
IMS ID . . : IMS1                              TIME . . : 16:05:30
                                               DATE . . : 2020/02/11
Members in . . . . . : IQC.ECDV255.IMS1.NOTPARMS
New member name  . . _____

Row options: Copy(C), Edit(E), Delete(D)

Action   Member Name   New Member   PDS description
_        COPY0001      _____
D        IQCQSUNP      _____       CURRENT OVERFLOW PARAMETERS TABLE
_        NEWMEM01      _____
***************************** Bottom of data *******************************
```

Figure 78. Queue space utilization notification parameters table panel (IQCP71MS)

Pressing Enter will cause panel IQCP72DW to be displayed:

```
   Help
 _____
 I │IQCP72DW    Delete Confirmation                                │ 1 to 3 of 3
 C │Command ===> _____ Scroll ====> PAGE       │l ===> PAGE
   │                                                               │
 P │Press ENTER to continue or END to exit.                        │
   │                                        more         +         │ 14 2020/01/
 S │                                 APAR    PH19814 2020/01/       │ 042
 I │Server . . . . . . : QCFSERVE                                  │ :30
   │IMS ID . . . . . . : IMS1                                      │ 02/11
 M │Default table  . . : IQCQSUNP                                  │
 N │                                                               │
   │Warning . . . . . .  u  Do you really want to delete the       │
 R │                        the default table: IQCQSUNP ?          │
   │  F1=Help   F3=End   F7=Up   F8=Down   F10=Actions             │
 A │ F12=Cancel                                                    │
   │_____│
 D │      IQCQSUNP              CURRENT OVERFLOW PARAMETERS TABLE
   │      NEWMEM01
 ****************************** Bottom of data ******************************
```

Figure 79. Delete Confirmation panel (IQCP72DW)

The system needs a CURRENT OVERFLOW PARAMETER TABLE, so it prompts you with a new member name of IQCQSUNP, as indicated in panel IQCP71MS. Copy this name to the member you want as the new current overflow parameter table and issue a **COPY** command.

Important: Ensure that you blank out the **New member name** field.

```
   Help
 _____
 IQCP71MS    Queue space utilization notification parameters ta Row 1 to 2 of 2
 Command ===> _____      Scroll ===> PAGE

 Press ENTER to continue or END to exit.
                                               APAR . . : PH19814 2020/01/
 Server . . : QCFSERVE                         JDTE . . : 2020.042
 IMS ID . . : IMS1                             TIME . . : 16:06:43
                                               DATE . . : 2020/02/11
 Members in . . . . : IQC.ECDV255.IMS1.NOTPARMS
 New member name  . . IQCQSUNP

 Row options: Copy(C), Edit(E), Delete(D)

 Action   Member Name   New Member   PDS description
 _        COPY0001       _____     _____
 _        NEWMEM01       _____     _____
 ****************************** Bottom of data ******************************

 _____
 │                                                                          │
 │IQC17130 Member containing the overflow parameters values used by IMS is  │
 │missing. Press enter if you want it rebuilt using the suggested name above.│
 │_____│
```

Figure 80. Queue space utilization notification parameters table panel (IQCP71MS)

After you type Y to the prompt and press Enter, panel IQCP71MS is displayed again. The panel indicates that IQCQSUNP has been recreated using the values in NEWMEM01.

```
   Help
 _____
IQCP71MS    Queue space utilizion notification parameters ta Row 1 to 3 of 3
Command ===> _____        Scroll ===> PAGE

 Press ENTER to continue or END to exit.
                                               APAR . . : PH19814 2020/01/
 Server . . : QCFSERVE                         JDTE . . : 2020.042
 IMS ID . . : IMS1                             TIME . . : 16:10:43
                                               DATE . . : 2020/02/11
 Members in . . . . : IQC.ECDV255.IMS1.NOTPARMS
 New member name  . . _____

 Row options: Copy(C), Edit(E), Delete(D)

 Action   Member Name  New Member  PDS description
 _        COPY0001     _____    _____
 _        IQCQSUNP     _____    CURRENT OVERFLOW PARAMETERS TABLE
 _        NEWMEM01     _____    _____
  ***************************** Bottom of data *****************************

 _____
|                                                                     |
|IQC1715E Updated overflow parameters sent to IMS.                    |
|                                                                     |
 _____
```

Figure 81. Queue space utilization notification parameters table panel (IQCP71MS)

Queue Overflow Notification Parameters panel

If you choose option **2 - View the current space utilization notification parameters** from the Queue Space utilization notification parameters panel (IQCP71AS), the Queue Threshold parameters panel (IQCP71VA) is displayed for AREA and FAILSAFE values.

```
   Help
 --------------------------------------------------------------------------
IQCP71VA                  Queue Threshold parameters
 Command ===>_____
 Press ENTER to continue or END to exit      . APAR . . : PH19814 2020/01/
   Server . . : QCFSERVE                        JDTE . . : 2020.039
   IMS ID . . : IMS1                            TIME . . : 09:00:38
                                                DATE . . : 2020/02/08
  Committed and uncommitted messages: Thresholds.
  Queue upper   threshold . . : 075    Queue lower threshold . . . : 060
  Queue IQC6101 threshold . . : 085

Committed messages: Job/Procedure names to be called at threshold crossover.
Ignore application calls inserting messages to express PCBs. . . : Y

Jobnames - A-B: . . . : NOUNLOAD B-C: . . : NOUNLOAD C-D: . . : NOUNLOAD
Procnames - A-B: . . . :          B-C: . . :          C-D: . . :

Jobname   - IQC6101  . : NOUNLOAD
Procname  - IQC6101  . :
```

Figure 82. Queue Threshold parameters (IQCP71VA)

Then, if you press Enter, panel IQCP91JS is displayed:

```
Help
-------------------------------------------------------------------------------
IQCP91JS            Queue Overflow Notification Parameters
Command ===>                                                   Scroll ===> PAGE

Select a row action or press ENTER to continie
                                                    APAR . . : PH19814 2020/01/
Server . . : IQCSERV                                JDTE . . : 2020.039
IMS ID . . : SYS3                                   TIME . . : 09:03:45
                                                    DATE . . : 2020/02/08

 "/" indicates Selected  Source/Destination Types
 Culprit and other actions: A - ABEND, N - NONE, O - WTO, S - STOP, W - WAIT
                         J - STOP THE SOURCE
 Enter V to view notification Jobs Lists for each AREA.
  Area     Percent  _Source/Destination Types_ _Culprit Act __Other Act_ Not
  Name     Tot Usd  ALL APPC APPL  DC MSC OTMA Strtd  Stppd Strtd  Stppd Lists
                                                                  More:     +
 AREA0001  20  50    /                         N      O     N      O     V
 AREA0002  25  45    /                         N      O     N      O     -
 AREA0003  30  40    /                         N      O     N      O     -
 AREA0004  40  35    /                         N      O     N      O     -
```

Figure 83. Queue Overflow Notification Parameters panel (IQCP91JS)

Press Enter to see the current FAILSAFE value (panel IQCP91JV), or the End key to return to the Main Menu panel.

```
Help
-------------------------------------------------------------------------------
IQCP91JV            Queue Overflow Notification Parameters
Command ===>                                                   Scroll ===> PAGE

Press ENTER to continue or END to exit.
                                                    APAR . . : PH19814 2020/01//
Server . . : IQCSERV                                JDTE . . : 2020.039
IMS ID . . : SYS3                                   TIME . . : 09:05:50
                                                    DATE . . : 2020/02/08
 Shows percent queue utilization and action for each FAILSAFE statement.
 Actions: A - ABEND, N - NONE, O - WTO, S - STOP, W - WAIT
 Enter V to view notification Jobs Lists for each FAILSAFE.
                                                            More:     +
                             PERCENT
 FAILSAFE ID     PERCENT     ALLOWED       ACTION    NOT LISTS
 FAIL0001        65          05            O         V
 FAIL0002        75          02            W         -
```

Figure 84. Queue Overflow Notification Parameters panel (IQCP91JV)

Tip: You can view detailed information for the notification job list entries for the selected AREA or FAILSAFE by entering V. If there are no notification job list entries, an empty list is shown.

For more information about how to use the panel IQCP91JV, use the **Help** pull-down menu. For detailed field-level help, press the Help key with your cursor on the field in question.

Pressing Enter will return you to panel IQCP71AS; pressing End will return you to the previous panel, IQCP91JS.

Part 4. Troubleshooting

The topics in this section provide you with technical references to help you troubleshoot and diagnose IMS Queue Control Facility problems.

You can diagnose errors and failures by using the diagnostics processor module to create SCRAPLOG and 6701-MRQE diagnostic records. Guidance information for interpreting diagnostic records is provided. You can also use error messages and AIB reason codes to help diagnose and solve any errors or failures.

Topics:

- Chapter 20, "Runtime messages (IQC)," on page 309
- Chapter 21, "AIB reason codes," on page 399
- Chapter 22, "Return codes," on page 437
- Chapter 23, "Recovering from AREA and FAILSAFE actions," on page 439
- Chapter 24, "Diagnosing requeuing problems," on page 447

Chapter 20. Runtime messages (IQC)

This reference section provides detailed information about IMS Queue Control Facility messages.

Messages are written to the QCFPRINT data set by the following IMS Queue Control Facility program functions:

- Automatic Checkpoint Selection
- Browse
- Cancel
- Load
- Parameter Processing
- Query
- Recover
- Select
- Unload
- Communication Server

Some of these messages are also issued as write-to-operator (WTO) messages. WTO messages will also be written to the job log or to some other destination based on the ROUTECDE execution parameter and the NOWTOMSG control statement. The default is to write all WTO messages to the JCL log.

Message format

IMS Queue Control Facility runtime messages adhere to the following format:

```
IQCnnnnx
```

where:

IQC
Indicates that the message was issued by IMS Queue Control Facility

nnnn
Indicates the message identification number

x
Indicates the severity of the message:

A
Indicates that operator intervention is required before processing can continue.

E
Indicates that an error occurred, which might or might not require operator intervention.

I
Indicates that the message is informational only.

W
Indicates that the message is a warning to alert you to a possible error condition.

T
Terminal.

C
Catastrophic.

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.

Module:
The affected module in the code.

IQC1000I **MESSAGE REQUEUER CANCEL ROUTINE**

Explanation

This message prints as soon as QCFPRINT is opened to indicate the start of processing.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCCANCL

IQC1001I **INVALID RECORD IN QCFIN, IGNORED**

Explanation

The QCFIN record appearing before this message has invalid data, which IQCCANCL could not decipher.

System action

Processing continues.

User response

Correct the invalid data and try the operation again.

Module

IQCCANCL

IQC1002T **UNKNOWN IMS RELEASE, TERMINATING**

Explanation

The IMS release level in the DFSMRQPF macro was not recognized by IQCCANCL.

System action

Processing terminates.

User response

Contact your IBM representative.

Module

IQCCANCL

IQC1003I **SORTINB DID NOT OPEN. FOLLOWING SHOULD BE USED AS THE SORT CONTROL RECORDS:**

Explanation

IQCCANCL tried to open the SORTINB data set to produce the SORT control records.

System action

Processing continues.

User response

The SORT control records are in the QCFPRINT output data set.

Module

IQCCANCL

IQC1004W **LOG CODE TYPE NOT VALID -** *hh*

Explanation

An invalid IMS log record was found. The value of *hh* is the code in hexadecimal.

System action

Processing continues.

User response

Report the problem to your IBM Software Support personnel.

Module

IQCCANCL

IQC1005E   **INCREASE LRECL FOR CANCLOUT**

Explanation

The logical record length of the CANCLOUT data set was not large enough to contain a selected IMS log record plus the DFSMRQPF macro prefix.

System action

Processing terminates.

User response

Increase the LRECL parameter on the CANCLOUT DD statement, and run the job again.

Module

IQCCANCL

IQC1006I   **MESSAGE NOT ENQUEUED; IT HAS BEEN DROPPED**

Explanation

A message record was found without an accompanying enqueue record.

System action

Processing continues.

User response

If you know that the message was properly processed, report the problem to your IBM software support personnel; otherwise, this is an information only message.

Module

IQCCANCL

IQC1007I   **TOTAL FOR THIS DESTINATION** *nnnnnnnn*

Explanation

The value of *nnnnnnnn* is the number of messages for this particular destination.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCCANCL

IQC1008I   **DEBUGGING NOW ENABLED**

Explanation

You supplied the DEBUG control statement.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCCANCL

IQC1009I   **DEBUGGING NOT ENABLED**

Explanation

You supplied the DEBUG control statement; however, the SNAPDUMP DD statement did not open.

System action

Processing continues.

User response

Supply the SNAPDUMP DD statement and run the job again.

Module

IQCCANCL

IQC1100T   *xxxxxxxx* **DD CARD FAILED TO OPEN, TERMINATING**

Explanation

Required DD statement that is shown in the value of *xxxxxxxx* failed to open.

System action

Processing terminates.

User response

Supply the missing DD statement and run the job again.

Module

IQCCANCL

IQC1101I   **MESSAGES SELECTED:** *nnnnnnnn*

Explanation

The value of *nnnnnnnn* is the number of messages IQCCANCL selected for processing.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCCANCL

IQC1102I   **END OF CANCEL**

Explanation

This message indicates a normal end of the IQCCANCL step.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCCANCL

IQC1103I   **ASSEMBLY DATE AND STATUS OF** *ssssssss*

Explanation

This informational message appears if the DEBUG option was chosen. The value of *ssssssss* is a CSECT within IQCCANCL. One message appears for each CSECT within IQCCANCL.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCCANCL

IQC1104E   **MESSAGES LOST - CHECK THE TIMESTAMPS**

Explanation

The log records have UOW entries for failed messages, but those messages cannot be found in the logs that were used to recover the queues. The timestamps for the recovered and missing messages are printed.

System action

Processing stops. The load step is not run.

User response

Compare the timestamps of the recovered messages to the missing messages. If the timestamps of the missing messages precede the timestamps of the recovered messages, select an earlier checkpoint and issue the RECOVERAS procedure.

Module

IQCCANCL

IQC2500I   **:CQS LOAD ROUTINE**

Explanation

This message is printed when control is passed to the load function by IMS QCF control processing

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCLOAD0

IQC2501I	END OF LOAD

Explanation

This message is printed when the load function passes control back to IQCXCTRL.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCLOAD0

IQC2502I	ASSEMBLY DATE AND STATUS OF *ssssssss*

Explanation

This informational message appears if the DEBUG option was chosen. The value of *ssssssss* is a CSECT within the load function. One message appears for each CSECT within the load function.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCLOAD0

IQC2503I	MESSAGES LOADED: *nnnnnnnn*

Explanation

The value of *nnnnnnnn* is the number of messages the load function loaded to the message queues.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCLOAD0

IQC2504I	UNKNOWN RELEASE OF IMS

Explanation

The release level of IMS in the IQCMRQPF macro was not recognized by the load function.

System action

Processing continues.

User response

Contact IBM software support.

Module

IQCQUERY

IQC2505I	PREMATURE EOD ON LOAD DATA SET

Explanation

Either no records were found in the LOAD data set, or an incomplete series of records was found.

System action

Processing continues.

User response

Run the UNLOAD again job and check its output.

Module

IQCLOAD0

IQC2506I	LRECL OF SCRAPLOG TOO SMALL; INCREASE IT

Explanation

The logical record length of the SCRAPLOG data set is not large enough to contain the log record plus the IQCMRQPF macro prefix.

System action

Processing continues.

User response

Increase the LRECL of the SCRAPLOG data set and run the job again. The SCRAPLOG data set should

be allocated with the following DCB specifications: DCB=(LRECL=32756,BLKSIZE=32760,RECFM=VB).

Module

IQCLOAD0

IQC2507W   **SCRAPLOG RECORDS TRUNCATED**

Explanation

The SCRAPLOG data set was defined with an LRECL that is less than the maximum record length.

System action

Processing continues.

User response

Correct the LRECL for the SCRAPLOG data set and run the job again. The SCRAPLOG data set should be allocated with the following DCB specifications: DCB=(LRECL=32756,BLKSIZE=32760,RECFM=VB).

Module

IQCLOAD0

IQC3000I   **MESSAGE REQUEUER SELECT ROUTINE**

Explanation

This message is printed as soon as QCFPRINT is opened to indicate the start of processing.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCSELCT

IQC3001E   **CHKPT CARD MISSING**

Explanation

The first control statement is not a CHKPT statement.

System action

Processing terminates.

User response

Supply the required control statement, or correct the one in error, and run the job again.

Module

IQCSELCT

IQC3002E   **INVALID CHKPT IDENTIFICATION NUMBER**

Explanation

The CHKPT statement operand contains characters that are not decimal digits.

System action

Processing terminates.

User response

Correct the CHKPT statement and run the job again.

Module

IQCSELCT

IQC3003E   **FUNCTION CARD MISSING**

Explanation

The second control statement is not a FUNCTION statement.

System action

Processing terminates.

User response

Supply the required control statement, or correct the one in error, and run the job again.

Module

IQCSELCT

IQC3004E   **INVALID FUNCTION**

Explanation

The FUNCTION statement operand is not RECOVERDM, RECOVERAB, or REPROCESS.

System action

Processing terminates.

User response

Correct the FUNCTION statement and run the job again.

Module

IQCSELCT

IQC3005E   **SELECT CARD INVALID**

Explanation

This control statement should be a SELECT, INCLUDE, or EXCLUDE statement.

System action

Processing terminates.

User response

Correct the control statement and run the job again. It will be the statement printed before the message in the control statement listing.

Module

IQCSELCT

IQC3006E   **CONTINUATION CARD NOT BLANK COLS 1-9**

Explanation

Continuation statements must be blank in columns 1 through 9.

System action

Processing terminates.

User response

Correct the continuation statement and run the job again.

Module

IQCSELCT

IQC3007E   **DUPLICATE SELECT PARAMETER**

Explanation

Duplicate keyword parameters are not allowed for the SELECT, INCLUDE, and EXCLUDE statements.

System action

Processing terminates.

User response

Begin a new SELECT, INCLUDE, or EXCLUDE statement, or eliminate the duplicate keyword, and try the operation again.

Module

IQCSELCT

IQC3008E   **UNKNOWN SELECT PARAMETER**

Explanation

A keyword parameter is invalid for the SELECT, INCLUDE, and EXCLUDE statements.

System action

Processing terminates.

User response

Correct the control statement and try the operation again.

Module

IQCSELCT

IQC3009E   **INVALID TYPE PARAMETER**

Explanation

The SELECT statement TYPE value is invalid.

System action

Processing terminates.

User response

Correct the TYPE value and try the operation again.

Module

IQCSELCT

IQC3010E   **MULTIPLE PARAMETERS ON SELECT CARD**

Explanation

Only one parameter per statement is allowed.

System action

Processing terminates.

User response

Use continuation statements for additional parameters and try the operation again.

Module

IQCSELCT

IQC3011E INVALID HEXADECIMAL PARAMETER

Explanation

A hexadecimal value contains characters that are not valid hexadecimal digits.

System action

Processing terminates.

User response

Correct the value in error and try the operation again. Valid hexadecimal digits are: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

Module

IQCSELCT

IQC3012E INVALID SYSID IN SELECT

Explanation

An MSC-system ID must be a 1- to 5-digit decimal number from 1 to 2036, in parentheses.

System action

Processing terminates.

User response

Correct the SYSID value and try the operation again.

Module

IQCSELCT

IQC3013E BLANK NAME, INVALID IN SELECT

Explanation

A SOURCE, DESTINATION, or LUNAME value in an INCLUDE or EXCLUDE statement is invalid.

System action

Processing terminates.

User response

Correct the control statement and try the operation again.

Module

IQCSELCT

IQC3014E MULTIPLE SELECT STATEMENTS

Explanation

Multiple SELECT statements are not allowed.

System action

Processing terminates.

User response

Use no more than one SELECT statement. Specify additional selection criteria with INCLUDE or EXCLUDE statements and try the operation again.

Module

IQCSELCT

IQC3015E INVALID DECIMAL VALUE

Explanation

A decimal value contains characters that are not valid decimal digits.

System action

Processing terminates.

User response

Correct the value in error and try the operation again. Valid digits are: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

Module

IQCSELCT

IQC3016W UNEXPECTED NN RECORD IN DUMPQ

Explanation

The record type that is shown in the value of *nn* was found in the DUMPQ data set; it was not expected.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCSELCT

IQC3017E   **CHKPT/FUNCTION MISSING**

Explanation

End-of-file on the QCFIN data set was reached before reading both a CHKPT and a FUNCTION statement.

System action

Processing terminates.

User response

Supply the missing control statements and try the operation again.

Module

IQCSELCT

IQC3018E   **CHECKPOINT ID > CHECKPOINT SPECIFIED**

Explanation

A checkpoint ID greater than the requested base checkpoint was read from the LOGIN data set. Either the wrong system log was used, or the wrong checkpoint was specified in the CHKPT control statement.

System action

Processing terminates.

User response

Provide the correct system log as the LOGIN data set, or correct the CHKPT control statement, and try the operation again.

Module

IQCSELCT

IQC3019E   **CHECKPOINT NOT DUMPQ OR PURGE**

Explanation

The specified base checkpoint was found but it was not a DUMPQ or PURGE checkpoint, and the program function is RECOVERDM.

System action

Processing terminates.

User response

Change the CHKPT control statement to specify the correct base checkpoint and try the operation again.

Module

IQCSELCT

IQC3020E   **CHECKPOINT NOT DUMPQ OR SNAPQ**

Explanation

The specified base checkpoint was found, but it was not a DUMPQ, PURGE, or SNAPQ, and the program function is RECOVERAB.

System action

Processing terminates.

User response

Change the CHKPT control statement to specify the correct base checkpoint and try the operation again.

Module

IQCSELCT

IQC3021W   **DUMPQ CHECKPOINT *nnnnnnnnnn* FOUND AFTER BASE CHECKPOINT**

Explanation

The DUMPQ checkpoint that is shown in the value of *nnnnnnnnnn* was found after the checkpoint was specified in the CKPT control statement.

System action

Processing continues.

User response

If the base checkpoint specified in the CHKPT control statement is known to be correct, ignore this. If the

checkpoint should have been this, run the job again with a new CKPT.

Module

IQCSELCT

IQC3022W   4002 CONTINUE RECORD MISSING, DRRN=*nnnnn*

Explanation

A 4002 record indicated a continuation record should be next; it was not found. The value that is shown in *nnnnn* is the record number of the 4002 record.

System action

Processing continues.

User response

Examine the log data set to determine if it was prematurely terminated.

Module

IQCSELCT

IQC3023W   INTERNAL LOGIC ERROR, DRRN=*nnnnn*

Explanation

An internal error was detected at the log record that is shown in the value of *nnnnn*.

System action

Processing continues.

User response

Report the problem to IBM Software Support.

Module

IQCSELCT

IQC3024E   MISSING CONTINUATION

Explanation

End-of-file occurred on QCFIN before the continuation of a previous operand was found.

System action

Processing terminates.

User response

Supply the missing continuation, and run the job again.

Module

IQCSELCT

IQC3025E   BASE CHECKPOINT NOT FOUND

Explanation

The checkpoint specified by the CHKPT control statement was not found in the LOGIN data set.

System action

Processing terminates.

User response

Provide the correct system log as the LOGIN data set, or correct the CHKPT control statement, and run the job again.

Module

IQCSELCT

IQC3026E   EOF BEFORE CHECKPOINT COMPLETE

Explanation

The LOGIN data set did not contain the entire checkpoint specified by the CHKPT control statement.

System action

Processing terminates.

User response

Include all volumes for the specified checkpoint in the LOGIN data set, or correct the CHKPT control statement, and run the job again.

Module

IQCSELCT

IQC3027E   INCREASE MAXIMUM RECORDSIZE ON MSGQ

Explanation

The record size of the data set pointed to by the MSGQ DD statement must be at least 256 bytes more than the IMS log record size.

System action

Processing terminates.

User response

Change the record size on the VSAM allocation step and run the job again.

Module

IQCSELCT

IQC3028E   **POINT ERROR ON MSGQ**

Explanation

This is an error in the VSAM POINT logic.

System action

Processing terminates.

User response

Report the problem to IBM Software Support.

Module

IQCSELCT

IQC3029E   **CHECK ERROR ON GET TO MSGQ**

Explanation

This is an error in the VSAM GET logic.

System action

Processing terminates.

User response

Report the problem to IBM Software Support.

Module

IQCSELCT

IQC3030E   **ERROR ON BLDVRP FOR MSGQ**

Explanation

This is an error in the VSAM BLDVRP logic.

System action

Processing terminates.

User response

Increase the region size in the REGION= parameter, and run the job again. If the error persists, report the problem to IBM Software Support.

Module

IQCSELCT

IQC3031E   **MISSING RIGHT PARENTHESIS**

Explanation

An operand that should have terminated with a right parenthesis did not.

System action

Processing terminates.

User response

Correct the statement, and run the job again.

Module

IQCSELCT

IQC3032E   **NUMBER OUT OF RANGE**

Explanation

The number specified in the previous statement is outside the range for that operand.

System action

Processing terminates.

User response

See the documentation topics that describe the statement, change the number so that it is within the range, and run the job again.

Module

IQCSELCT

IQC3033E   **NAME GREATER THAN 8 CHARACTERS**

Explanation

The name specified in the previous statement is limited to 8 characters.

System action

Processing terminates.

User response

Correct the name and run the job again.

Module

IQCSELCT

IQC3034E	NAME GREATER THAN 16 CHARACTERS

Explanation

The name specified in the previous statement is limited to 16 characters.

System action

Processing terminates.

User response

Correct the name and run the job again.

Module

IQCSELCT

IQC3035E	PREMATURE EOF ON QCFIN

Explanation

End-of-file was encountered on QCFIN before all control statements were read.

System action

Processing terminates.

User response

Correct the control statements and run the job again.

Module

IQCSELCT

IQC3036I	SORTIN*x* DID NOT OPEN. FOLLOWING SHOULD BE USED AS THE SORT CONTROL RECORDS:

Explanation

IQCSELCT attempted to open SORTINA and/or SORTINB to supply the proper sort control records.

System action

Processing continues.

User response

None. The proper sort control records are in QCFPRINT.

Module

IQCSELCT

IQC3037E	UNKNOWN RELEASE OF IMS

Explanation

The record length of the first 4001 checkpoint on the LOGIN data set was not one that was recognized by the QCF program.

System action

Processing terminates.

User response

If the LOGIN data set is known to be a valid IMS checkpoint data set, report the problem to IBM Software Support. If not, run the job again with a correct IMS checkpoint data set.

Module

IQCSELCT

IQC3038E	TIMESTAMP: DATE MISSING PERIOD

Explanation

The data supplied after the D= of a UTC did not have a period as the third or fifth character. All dates must be in the form *yyyy.ddd* or *yy.ddd*.

System action

Processing terminates.

User response

Correct the data and run the job again.

Module

IQCSELCT

IQC3039E	TIMESTAMP: T= KEYWORD MISSING

Explanation

The second operand of a UTC was not T=. UTC data must be in the form D=, T=.

System action

Processing terminates.

User response

Correct the data and run the job again.

Module

IQCSELCT

IQC3040E   **TIMESTAMP: TIME MISSING COLON**

Explanation

The third character or fifth character of the data after the T= operand of a UTC was not a colon (:). The T= operand must be in the form T=*hh:mm:*.

System action

Processing terminates.

User response

Correct the data and run the job again.

Module

IQCSELCT

IQC3041E   **TIMESTAMP: TIME MISSING PERIOD**

Explanation

The ninth character of the data after the T= operand of a UTC was not a period. The T= operand must be in the form T=*hh:mm:ss.*

System action

Processing terminates.

User response

Correct the data and run the job again.

Module

IQCSELCT

IQC3042E   **TIMESTAMP: ZONE UNKNOWN SIGN**

Explanation

The first character after the *Z=* of a UTC was not a +, -, or a number. The *Z=* operand must be in the form *Z=shh*, (where *s* is *+* or *-*) or Z=*hh*.

System action

Processing terminates.

User response

Correct the data and run the job again.

Module

IQCSELCT

IQC3043E   **TIMESTAMP: ZONE MISSING COLON**

Explanation

The second, third, or fourth character after the Z= of a UTC was not a colon (:) or a blank. The Z= operand must be in the form Z=*shh:*, Z=*hh:,* or Z=*h:.*

System action

Processing terminates.

User response

Correct the data and run the job again.

Module

IQCSELCT

IQC3044E   **TIMESTAMP ZONE: UNKNOWN MINUTE OFFSET**

Explanation

The data after the colon in the Z= of a UTC was not 00, 15, 30, or 45. Minutes in the Z= of a UTC must be in quarter hour increments.

System action

Processing terminates.

User response

Correct the data and run the job again.

Module

IQCSELCT

IQC3045E **INCREASE LRECL FOR LOGOUT DATA SET**

Explanation

The logical record length of the LOGOUT data set was not large enough to contain a selected IMS log record plus the DFSMRQPF macro prefix.

System action

Processing terminates.

User response

Increase the LRECL parameter on the LOGOUT DD statement and run the job again.

Module

IQCSELCT

IQC3046E **UNKNOWN KEYWORD FOUND IN UTC TIMESTAMP**

Explanation

An unknown keyword was found in a UTC timestamp. Keywords in the UTC timestamp are D=, T=, and Z=.

System action

Processing terminates.

User response

Correct the data and run the job again.

Module

IQCSELCT

IQC3047E **CHECK ERROR ON PUT TO MSGQ**

Explanation

This is an error in the VSAM PUT logic.

System action

Processing terminates.

User response

Report the problem to IBM Software Support.

Module

IQCSELCT

IQC3048E **INCREASE LRECL FOR SCRAPSEL DATA SET**

Explanation

The logical record length of the SCRAPSEL data set was less than the IMS log record chosen, plus the length of the DFSMRQPF macro prefix.

System action

Processing terminates.

User response

Increase the LRECL of the SCRAPSEL data set and run the job again.

Module

IQCSELCT

IQC3049E **DUPLICATE INCLUDE PARAMETER**

Explanation

The INCLUDE keyword parameter on the previous record was used on a prior record.

System action

Processing terminates.

User response

Correct the INCLUDE statement and run the job again.

Module

IQCSELCT

IQC3050E **DUPLICATE EXCLUDE PARAMETER**

Explanation

The EXCLUDE keyword parameter on the previous record was used on a prior record.

System action

Processing terminates.

User response

Correct the EXCLUDE statement and run the job again.

Module

IQCSELCT

IQC3051E   **UNKNOWN INCLUDE PARAMETER**

Explanation

The INCLUDE keyword parameter on the previous record was not recognized by IQCSELECT.

System action

Processing terminates.

User response

Correct the INCLUDE statement and run the job again.

Module

IQCSELCT

IQC3052E   **UNKNOWN EXCLUDE PARAMETER**

Explanation

The EXCLUDE keyword parameter on the previous record was not recognized by IQCSELECT.

System action

Processing terminates.

User response

Correct the EXCLUDE statement and run the job again.

Module

IQCSELCT

IQC3053W   **GET FROM LOG RETURNED UNKNOWN VBS RECORD TYPE - *hh***

Explanation

Record type that is shown in the value of *hh* is an unknown VBS record type.

System action

Processing continues.

User response

Contact IBM software support.

Module

IQCSELCT

IQC3054T   **CHECKPOINT *nnnnn* - NO END OF CHECKPOINT RECORD FOUND. TERMINATING.**

Explanation

No end-of-checkpoint record was found. The value that is shown in *nnnnn* is the checkpoint number that was identified in a previous IQC3102I, IQC3107I, or IQC3108I message.

System action

Processing terminates.

User response

Provide the correct system log as the LOGIN data set, or correct the CHKPT control statement, and run the job again.

Module

IQCSELCT

IQC3055E   **TIMESTAMP: DATE OUT OF RANGE**

Explanation

The date specified in a CHKPT control statement was not in the range of 1 to 366.

System action

Processing terminates.

User response

Correct the CHKPT control statement and run the job again.

Module

IQCSELCT

IQC3056E   **TIMESTAMP: ZONE OUT OF RANGE**

Explanation

The zone specified in a CHKPT control statement was not in the range of 0 to 12.

System action

Processing terminates.

User response

Correct the CHKPT control statement and run the job again.

Module

IQCSELCT

IQC3057I   **NO SELECT STATEMENTS PRESENT, ALL MESSAGES WILL BE PROCESSED**

Explanation

No SELECT statements were present in the QCFIN data set; therefore, all of the input IMS log records will be processed.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCSELCT

IQC3058E   **TIMESTAMP: UTC FORMAT INVALID WITH VERSION 5.1**

Explanation

A timestamp in the form D=, T=, or Z= was passed to IQCSELECT, but the input IMS log was from version 5.1.

System action

Processing terminates.

User response

Correct the CHKPT control statement, or supply the correct system log as the LOGIN data set. Then run the job again.

Module

IQCSELCT

IQC3059I   **TOTAL FOR THIS DESTINATION** *nnnnnnnn*

Explanation

nnnnnnnn is the number of messages for this particular destination.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCSELCT

IQC3060I   **DEBUGGING NOW ENABLED**

Explanation

You supplied the DEBUG control statement.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCSELCT

IQC3061E   **BEGIN CHECKPOINT (TYPE 4001) NOT FOUND**

Explanation

The beginning record for a checkpoint type 4001 was not found in the LOGIN data set. The checkpoint does not start on the input data sets.

System action

Processing terminates.

User response

Check the LOGIN data sets and run the job again.

Module

IQCSELCT

IQC3062E   **TIMESTAMP: FORMAT INVALID WITH VERSION 6.1 AND UP**

Explanation

Timestamp has the format *yydddhhmmss*, which is valid for IMS 5.1 only.

System action

Processing terminates.

User response

Correct the timestamp and run the job again.

Module

IQCSELCT

| IQC3063E | TIMESTAMP: INVALID SYMBOL IN T= PARAMETER |

Explanation

Timestamp should have the format *HH:MM:SS.IJKLMN*, where H, M, S, I, J, K, L, M, and N are numeric, or the timestamp should have the format *HH:MM:SS* only.

System action

Processing terminates.

User response

Correct the timestamp and run the job again.

Module

IQCSELCT

| IQC3064E | NO MSG / SRC / DSTTYPE SPECIFIED |

Explanation

An invalid value was specified in a MSGTYPE, SRCTYPE, or DSTTYPE operand.

System action

Processing terminates.

User response

Remove the operand, or enter a valid value, and run the job again.

Module

IQCSELCT

| IQC3065E | MSG / SRC / DSTTYPE NOT RECOGNIZED OR FORMAT INCORRECT |

Explanation

The value specified in a MSGTYPE, SRCTYPE, or DSTTYPE operand was not recognized.

System action

Processing terminates.

User response

Enter valid values and run the job again.

Module

IQCSELCT

| IQC3100T | *xxxxxxxx* DD CARD FAILED TO OPEN, TERMINATING |

Explanation

Required DD statement, *xxxxxxxx* failed to open.

System action

Processing terminates.

User response

Supply the missing DD statement and run the job again.

Module

IQCSELCT

| IQC3101I | SEARCH FOR BASE CHECKPOINT |

Explanation

IQCSELCT is reading the LOGIN data set for the specified checkpoint.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCSELCT

IQC3103I   BASE CHECKPOINT FOUND AT LOGICAL RECORD *nnnnnnnn*

Explanation

The base checkpoint specified in a CHKPT control statement was found at logical IMS log record *nnnnnnnn*.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCSELCT

IQC3104I   BYPASSED SNAPQ *ccccccccccc*

Explanation

There is a SNAPQ checkpoint with an ID of *ccccccccccc* in the LOGIN data set that is more recent than the SNAPQ checkpoint being used.

System action

Processing continues.

User response

If the base checkpoint specified in the CHKPT control statement is correct, ignore this message and run the job again with a new CHKPT.

Module

IQCSELCT

IQC3105I   MESSAGES SELECTED: *nnnnnnnn*

Explanation

The value that is shown in *nnnnnnnn* is the number of messages IQCSELCT selected for processing.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCSELCT

IQC3106I   END OF SELECT

Explanation

Normal end of IQCSELCT step.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCSELCT

IQC3107I   CHECKPOINT *nnnnn* D=*yyyy.ddd*, T=*hh:mm:ss.thmiju*, Z=*shh:mm*.

Explanation

The value that is shown in *nnnnn* is the checkpoint number. D=*yyyy.ddd* is the checkpoint date stamp. T=*hh:mm:ss.thmiju* is the checkpoint time stamp. Z=*shh:mm* is the checkpoint zone.

System action

Processing continues.

User response

None. This message is informational.

Module

IQCSELCT

IQC3108I   CHECKPOINT *nnnnn* - *hh* - *chkpt_type*

Explanation

The value that is shown in *nnnnn* is the checkpoint number. The value that is shown in *hh* is the checkpoint type in hexadecimal format. *chkpt_type* indicates the checkpoint type in readable format, which can be one or more of the following character strings:

- `Coldstart in Progress`
- `Purge`
- `IMS Shutdown`

- Dump Queues
- MTO Request
- SnapQ
- Log Initiated

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCSELCT

IQC3109I	LRECL OF LOGOUT DATA SET IS TOO SMALL

Explanation

At least one IMS log record, when prefixed with the DFSMRQPF macro prefix, was larger then the logical record length of the LOGOUT data set.

System action

Processing continues.

User response

Increase the LRECL of the LOGOUT data set and run the job again

Module

IQCSELCT

IQC3111I	ASSEMBLY DATE AND STATUS OF *ssssssss*

Explanation

This informational message appears if the DEBUG option was chosen. The value that is shown in *ssssssss* is a CSECT within IQCSELCT. One message appears for each CSECT within IQCSELCT.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCSELCT

IQC3112I	CHECKPOINT *nnnnn* - END OF CHECKPOINT AT LOGICAL RECORD *nnnnnnnn*

Explanation

The end of checkpoint *nnnnn* was found at logical IMS log record *nnnnnnnn*.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCSELCT

IQC3113E	IQCSELCT EXCEEDED REAL MEMORY ALLOCATIONS

Explanation

A GETMAIN failed to give back the requested real storage.

System action

Processing terminates.

User response

Increase the REGION parameter in the EXEC statement and run the job again.

Module

IQCSELCT

IQC3114E	IQCSELCT INITIALIZATION FAILED- INCREASE REGION

Explanation

The REGION parameter in the EXEC statement is too small to run IQCSELECT

System action

Processing terminates.

User response

Increase the REGION parameter in the EXEC statement and run the job again.

Module

IQCSELCT

IQC3115T   **AUTOMATIC REBUILD FAILED - SEE MESSAGES IN JOBLOG**

Explanation

This is a notification that error messages have been displayed on the job log.

System action

Processing continues.

User response

See messages on the job log.

Module

IQCSELCT

IQC3300I   **CQS UNLOAD ROUTINE**

Explanation

This message is printed when control is passed to IQCUNLOD by IQCXCTRL.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCUNLOD

IQC3301I   **END OF UNLOAD**

Explanation

This message is printed when IQCUNLOD passes control back to IQCXCTRL.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCUNLOD

IQC3302I   **MESSAGES SELECTED:** *nnnnnnnn*

Explanation

The value that is shown in *nnnnnnnn* is the number of messages that IQCUNLOD unloaded from the message queues.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module:

IQC3303I   **MESSAGE SELECTED:** *nnnnnnnn*

Explanation

The value that is shown in *nnnnnnnn* is the message that IQCUNLOD unloaded from the message queue.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module:

IQC3304I   **UNLOAD DDCARD FAILED TO OPEN; ACTION=***REPORTONLY* **WILL BE USED**

Explanation

The UNLOAD DD statement was not in the input job stream. The UNLOAD function will proceed as if ACTION=*REPORTONLY* was specified.

System action

Processing continues.

User response

If the UNLOAD DD statement was intentionally omitted, change the SELECT control record to specify

ACTION=*REPORTONLY*. If the UNLOAD DD statement was unintentionally omitted, supply an UNLOAD DD statement and run the job again.

Module

IQCUNLOD

IQC3305W	UNLOAD DDCARD IS DD DUMMY

Explanation

The UNLOAD DD statement was specified as DUMMY. Any messages on the queues will be unloaded, leaving the user with no backup of the unloaded messages.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCUNLOD

IQC3306E	THE UNLOAD DD STATEMENT CONDITIONAL DISPOSITION CANNOT BE *DELETE*. IF UNSPECIFIED, THE NORMAL_DISP IS *KEEP* OR *CATALOG*.

Explanation

The UNLOAD DD statement in the JCL has either *DELETE* specified as the conditional disposition of the UNLOAD data set, or the conditional disposition of the UNLOAD data set has not been specified.

System action

Processing terminates. The UNLOAD function fails.

User response

Modify the JCL so that the conditional disposition of the UNLOAD data set is either *UNCATLG, CATLG, KEEP*, or *PASS* and run the job again.

Module

IQCUNLOD

IQC3307I	JOIN IS INTENDED, TOIPARMS STATEMENT MISSING.

Explanation

A JOIN operation was intended but the TOIPARMS statement was not specified.

System action

Processing terminates.

User response

Specify a TOIPARMS statement in the control statements.

See "TOIPARMS statement for UNLOAD" on page 258.

Module

IQCUNLOD

IQC3308I	STOP WILL BE ISSUED FOR: *dest*

Explanation

A STOP function will be issued for the destination that is indicated by the value of *dest*.

System action

Processing continues.

User response

None

Module

IQCUNLOD

IQC3309E	INIT FOR XCF SERVICES FAILED RC=*aaaaaaaa* RSN=*bbbbbbbb*.

Explanation

Initialization for XCF services failed with return code *aaaaaaaa* and reason code *bbbbbbbb*.

System action

Processing terminates.

User response

See *z/OS JES3 Initialization and Tuning Reference* and possibly *z/OS JES3 Diagnosis*.

Module

IQCUNLOD

IQC3310E   **CONNECT FAILED FOR XCF GROUP**

Explanation

The CONNECT function has failed for the XCF group.

System action

Processing terminates.

User response

Ensure that the TOSI customization for IMS has been done.

Module

IQCUNLOD

IQC3311E   **IMSID** *xxxxxxxx* **is not in GROUP**

Explanation

The IMSID shown in *xxxxxxxx* was not found in the group.

System action

Processing terminates.

User response

Specify a valid IMSID and run the job again.

Module

IQCUNLOD

IQC3312E   **SEND FAILED FOR STOP COMMAND.**

Explanation

The SEND function for the **STOP** command has failed.

System action

Processing terminates.

User response

See *z/OS JES3 Initialization and Tuning Reference* and possibly *z/OS JES3 Diagnosis*.

Module

IQCUNLOD

IQC3313E   **DISCONNECT FAILED.**

Explanation

The DISCONNECT function has failed.

System action

Processing terminates.

User response

See *z/OS JES3 Initialization and Tuning Reference* and possibly *z/OS JES3 Diagnosis*.

Module

IQCUNLOD

IQC3314E   **TERM FAILED**

Explanation

The termination function has failed.

System action

Processing terminates.

User response

See *z/OS JES3 Initialization and Tuning Reference* and possibly *z/OS JES3 Diagnosis*.

Module

IQCUNLOD

IQC3315I   **STORAGE REQUEST FAILED - MESSAGES CAN BE LOST IF UNLOAD DATA SET IS TOO SMALL**

Explanation

This message is displayed when storage is obtained to save messages before writing them to the unload data set.

System action

Processing continues.

User response

Increase the region size for the job to avoid getting this message in the future.

Module

IQCUNLOAD

IQC3316I   **LAST MESSAGES SAVED IN** *data set name*

Explanation

This message is displayed when the unload data set overflows and the last unloaded messages are written to the dynamically allocated emergency data set.

System action

Processing terminates.

User response

None. The last unloaded messages are written to the newly allocated data set. You must concatenate the new data set has to be concatenated to the unload data set if you want to perform subsequent actions with the unloaded messages.

Module

IQCUNLOAD

IQC3500I   **CQS QUERY ROUTINE**

Explanation

This message is printed when control is passed to the IQCQUERY routine by IQCXCTRL.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCQUERY

IQC3501I   **END OF QUERY**

Explanation

This message is printed when IQCQUERY passes control back to IQCXCTRL.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCQUERY

IQC3502I   **TOTAL AGED SECONDARY MESSAGES FOR THIS QUEUE:** *nnnnnnnn*

Explanation

The value that is shown in *nnnnnnnn* is the total number of aged secondary messages for this queue.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCQUERY

IQC3503I   **NUMBER OF DESTINATIONS QUERIED:** *nnnnnnnn*

Explanation

The value that is shown in *nnnnnnnn* is the number of destinations that IQCQUERY queried.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCQUERY

IQC3504I   **SECONDARY MESSAGES FOR THIS QUEUE:** *nnnnnnnn*

Explanation

The value that is shown in *nnnnnnnn* is the number of secondary messages that IQCQUERY found on this particular message queue.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCQUERY

IQC3505I   **TOTAL AGED PRIMARY MESSAGES FOR THIS QUEUE:** *nnnnnnnn*

Explanation

The value that is shown in *nnnnnnnn* is the number of primary message entries that IQCQUERY found on this particular message queue that matched the age specified in the AGE= keyword.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCQUERY

IQC3506I   **TOTAL PRIMARY MESSAGES FOR THIS QUEUE:** *nnnnnnnn*

Explanation

The value that is shown in *nnnnnnnn* is the number of primary entries IQCQUERY found for this particular message queue.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCQUERY

IQC3507I   **TOTAL DESTINATIONS FOR THIS QUEUE:** *nnnnnnnn*

Explanation

The value that is shown in *nnnnnnnn* is the number of destinations that IQCQUERY found for this particular message queue.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCQUERY

IQC3508I   **TOP** *nnnnnnnn* **DESTINATIONS**

Explanation

The value that is shown in *nnnnnnnn* are the number of top destinations that IQCQUERY found for this particular message queue.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCQUERY

IQC3509I   **DESTINATION** *nnnnnnnn* **Q-TYPE** *nnnnnnnn* **PRIMARY** *nnnnnnnn* **SECONDARY** *nnnnnnnn*

Explanation

This is one of a series of responses that are displayed on the Operator's console when the operator requests it.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCQUERY

IQC3510I   **TOP** *nnnnnnnn* **SHARED QUEUE DESTINATIONS**

Explanation

This is one of a series of responses that are displayed on the Operator's console when the operator requests it.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCQUERY

IQC3511I	DEST *nnnnnnnn* QT PRIMLE/ OVFLLE *nnnnnnnn* LASTACTIVE

Explanation

This is one of a series of responses that are displayed on the Operator's console when the operator requests it.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCQUERY

IQC3512I	*aaaaaaaa* QQ *nn.n%/nn.n% yy/mm/dd-hh:mm:ss.1*

Explanation

This is one of a series of responses that are displayed on the Operator's console when the operator requests it.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCQUERY

IQC3513I	Top *nnnnnnnn* SHORT QUEUE DESTINATIONS

Explanation

This is one of a series of responses that are displayed on the Operator's console when the operator requests it.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCQUERY

IQC3514I	DEST QT SHORTBUFS LASTACTIVE

Explanation

This is one of a series of responses that are displayed on the Operator's console when the operator requests it.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCQUERY

IQC3515I	*aaaaaaaa* qq *nnnnnnn/nn.n% yy/mm/dd-hh:mm:ss.1*

Explanation

This is one of a series of responses that are displayed on the Operator's console when the operator requests it.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCQUERY

IQC3516I   **TOP LONG QUEUE DESTINATIONS**

Explanation

This is one of a series of responses that are displayed on the Operator's console when the operator requests it.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCQUERY

IQC3517I   **DEST QT LONGBUFS LASTACTIVE**

Explanation

This is one of a series of responses that are displayed on the Operator's console when the operator requests it.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCQUERY

IQC3518I   *aaaaaaaa qq nnnnnnn/nn.n% yy/mm/dd-hh:mm:ss.1*

Explanation

This is one of a series of responses that are displayed on the Operator's console when the operator requests it.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCQUERY

IQC3519I   **TOP ALL QUEUE DESTINATIONS**

Explanation

This is one of a series of responses that are displayed on the Operator's console when the operator requests it.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCQUERY

IQC3520I   **DEST QT LONGB/SHRTB LASTACTIVE**

Explanation

This is one of a series of responses that are displayed on the Operator's console when the operator requests it.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCQUERY

IQC3521I   *aaaaaaaa qq nn.n%/nn.n% yy/mm/dd-hh:mm:ss.1*

Explanation

This is one of a series of responses that are displayed on the Operator's console when the operator requests it.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCQUERY

IQC3522I　　　　**TOP LAST ACTIVE DESTINATIONS**

Explanation

This is one of a series of responses that are displayed on the Operator's console when the operator requests it.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCQUERY

IQC3523I　　　　**Dest QT LastActive LongB/ShrtB**

Explanation

This is one of a series of responses that are displayed on the Operator's console when the operator requests it.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCQUERY

IQC3524I　　　　*aaaaaaaa qq yy/mm/dd-*
　　　　　　　　　　hh:mm:ss.1 nn.n%/nn.n%1

Explanation

This is one of a series of responses that are displayed on the Operator's console when the operator requests it.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCQUERY

IQC3525I　　　　**Top Last Active Shared Queue Dest**

Explanation

This is one of a series of responses that are displayed on the Operator's console when the operator requests it.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCQUERY

IQC3526I　　　　**Dest QT LastActive LastActive**
　　　　　　　　　　PrimLE/OvflLE

Explanation

This is one of a series of responses that are displayed on the Operator's console when the operator requests it.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCQUERY

IQC3527I　　　　*aaaaaaaa qq nn.n%/nn.n%*
　　　　　　　　　　yy/mm/dd-hh:mm:ss.1

Explanation

This is one of a series of responses that are displayed on the Operator's console when the operator requests it.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCQUERY

IQC4000I   **MESSAGE REQUEUER CQS CONTROLLER ROUTINE**

Explanation

This message is printed as soon as QCFPRINT is opened to indicate the start of processing.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCXREAD

IQC4001E   **CHKPT CARD MISSING**

Explanation

A CHKPT statement was not supplied for an operation or function that required one.

System action

Processing terminates.

User response

Supply the required CHKPT statement, or correct the one that is in error, and run the job again.

Module

IQCXREAD

IQC4002E   **INVALID CHKPT IDENTIFICATION NUMBER**

Explanation

The CHKPT statement value contains characters that are not decimal digits.

System action

Processing terminates.

User response

Correct the CHKPT statement and run the job again.

Module

IQCXREAD

IQC4003I   **NO FUNCTION STATEMENT PRESENT, BROWSE QUETYPE=*ALL* WILL BE PROCESSED**

Explanation

A FUNCTION control statement was not found in the QCFIN data set; therefore, FUNCTION will default to a BROWSE of all queues.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCXREAD

IQC4004E   **UNKNOWN FUNCTION PARAMETER**

Explanation

The keyword specified after the FUNCTION keyword was not recognized. FUNCTION must be one of the following:

- BROWSE
- LOAD
- QUERY
- RECOVER
- REPROCESS
- UNLOAD

System action

Processing terminates.

User response

Correct the FUNCTION control statement and run the job again.

Module

IQCXREAD

IQC4005E   **INVALID DELIMITER**

Explanation

An unknown delimiter was found in a control statement.

System action

Processing terminates.

User response

See the topics in the documentation that describe the control statement that is listed on the line before the message for its delimiters. Correct the control statement and run the job again.

Module:

IQC4006E   **CONTINUATION CARD NOT BLANK COLS 1-9**

Explanation

Continuation statements must be blank in columns 1 through 9.

System action

Processing terminates.

User response

Correct the continuation statement and run the job again.

Module

IQCXREAD

IQC4007E   **DUPLICATE SELECT PARAMETER**

Explanation

Duplicate keyword parameters are not allowed on the SELECT control statement.

System action

Processing terminates.

User response

Correct the SELECT control statement and run the job again.

Module

IQCXREAD

IQC4008E   **UNKNOWN SELECT PARAMETER**

Explanation

A keyword parameter is invalid for the SELECT statement.

System action

Processing terminates.

User response

Correct the control statement and run the job again.

Module

IQCXREAD

IQC4009E   **UNKNOWN TYPE PARAMETER**

Explanation

The SELECT statement TYPE value is invalid.

System action

Processing terminates.

User response

Correct the TYPE value and run the job again.

Module

IQCXREAD

IQC4010E   **MULTIPLE PARAMETERS NOT ALLOWED ON SAME RECORD**

Explanation

Only one parameter per statement is allowed.

System action

Processing terminates.

User response

Use continuation statements for additional parameters and run the job again.

Module

IQCXREAD

IQC4011E   INVALID HEXIDECIMAL PARAMETER

Explanation

A hexadecimal value contains characters that are not valid hexadecimal digits.

System action

Processing terminates.

User response

Correct the value in error and run the job again. Valid digits are: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

Module

IQCXREAD

IQC4012E   NO SOURCE OR DESTINATION, AND NO SYSID

Explanation

An MSC-system ID must be a 1- to 5-digit decimal number from 1 to 2036, enclosed in parentheses.

System action

Processing terminates.

User response

Correct the SYSID value and run the job again.

Module

IQCXREAD

IQC4013E   BLANK NAME, INVALID IN SELECT

Explanation

A SOURCE, DESTINATION, or LUNAME value in an INCLUDE or EXCLUDE statement is missing.

System action

Processing terminates.

User response

Correct the control statement and run the job again.

Module

IQCXREAD

IQC4014E   MULTIPLE SELECT STATEMENTS

Explanation

Multiple SELECT statements are not allowed.

System action

Processing terminates.

User response

Use no more than one SELECT statement. Specify additional selection criteria with INCLUDE or EXCLUDE statements and run the job again.

Module

IQCXREAD

IQC4015E   Invalid decimal value or hexadecimal value. Check also the length of the value.

Explanation

This error can occur in one or more of the following cases:

- A decimal value contains characters that are not valid decimal digits.
- A hexadecimal value contains characters that are not valid hexadecimal characters.
- The value exceeds maximum digits

System action

Processing terminates.

User response

Correct the value in error and run the job again.

Valid decimal digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Valid hexadecimal digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

Module

IQCXREAD

IQC4016W   UNKNOWN BROWSE PARAMETER

Explanation

An unknown keyword parameter was found in a BROWSE statement.

System action

Processing continues.

User response

Correct the control statement and run the job again.

Module

IQCXREAD

IQC4017E   UNKNOWN KEYWORD SPECIFIED

Explanation

An unknown keyword was found on a control statement.

System action

Processing terminates.

User response

Correct the control statement and run the job again.

Module

IQCXREAD

IQC4018E   CHECKPOINT ID > CHECKPOINT SPECIFIED

Explanation

A checkpoint ID that is greater than the requested base checkpoint was read from the LOGIN data set. Either the wrong system log was used, or the wrong checkpoint was specified in the CHKPT control statement.

System action

Processing terminates.

User response

Provide the correct system log as the LOGIN data set, or correct the CHKPT control statement, and run the job again.

Module

IQCXREAD

IQC4019E   UNKNOWN QUEUETYPE= *parameter*

Explanation

An unknown parameter was found after the QUEUETYPE= keyword, it is shown in *parameter*.

System action

Processing terminates.

User response

Correct the control statement and run the job again.

Module

IQCXREAD

IQC4020I   UNKNOWN RECOVER PARAMETER

Explanation

An unknown parameter was found in a RECOVER statement.

System action

Processing continues.

User response

Correct the RECOVER control statement and run the job again.

Module

IQCXREAD

IQC4021I   UNKNOWN ACTION PARAMETER

Explanation

An unknown parameter was found after the ACTION= keyword.

System action

Processing continues.

User response

Correct the ACTION control statement and run the job again.

Module

IQCXREAD

IQC4022I   UNKNOWN QUERY PARAMETER

Explanation

An unknown parameter was found in a QUERY statement.

System action

Processing continues.

User response

Correct the QUERY control statement and run the job again.

Module

IQCXREAD

IQC4023I	UNKNOWN LOAD PARAMETER

Explanation

An unknown parameter was found in a LOAD statement.

System action

Processing continues.

User response

Correct the LOAD control statement and run the job again.

Module

IQCXREAD

IQC4024E	MISSING CONTINUATION

Explanation

An end-of-file condition occurred on QCFIN before the continuation of a previous value was found.

System action

Processing terminates.

User response

Supply the missing continuation and run the job again.

Module

IQCXREAD

IQC4025E	BASE CHECKPOINT NOT FOUND

Explanation

The checkpoint that was specified by the CHKPT control statement was not found in the LOGIN data set.

System action

Processing terminates.

User response

Provide the correct system log as the LOGIN data set, or correct the CHKPT control statement, and run the job again.

Module

IQCXREAD

IQC4026E	EOF BEFORE CHECKPOINT COMPLETE

Explanation

The LOGIN data set did not contain the entire checkpoint that was specified by the CHKPT control statement.

System action

Processing terminates.

User response

Include all volumes for the specified checkpoint in the LOGIN data set, or correct the CHKPT control statement, and run the job again.

Module

IQCXREAD

IQC4027E	UNKNOWN UNLOAD PARAMETER

Explanation

An unknown parameter was found in an UNLOAD statement.

System action

Processing terminates.

User response

Correct the control statement and run the job again.

Module

IQCXREAD

**IQC4028E   UNKNOWN TRANSACTION
                  PARAMETER**

Explanation

An unknown parameter was found in a TRANSACTION statement.

System action

Processing terminates.

User response

Correct the control statement and run the job again.

Module

IQCXREAD

IQC4029E   UNKNOWN RECOVER PARAMETER

Explanation

An unknown parameter was found on a FUNCTION RECOVER control statement.

System action

Processing terminates.

User response

Correct the FUNCTION control statement and run the job again.

Module

IQCXREAD

**IQC4030I   ACTION NOT SPECIFIED,
                  DEFAULTING TO READ**

Explanation

No ACTION was specified for a FUNCTION RECOVER control statement.

System action

Processing continues.

User response

If the ACTION keyword was intentionally omitted, no response is necessary. If the ACTION keyword was

unintentionally omitted, supply an ACTION keyword and run the job again.

Module

IQCXREAD

IQC4031E   MISSING RIGHT PARENTHESIS

Explanation

An value that should have terminated with a right parentheses did not.

System action

Processing terminates.

User response

Correct the statement and run the job again.

Module

IQCXREAD

IQC4032E   NUMBER OUT OF RANGE

Explanation

The number that was specified in the previous value is outside of the range for that value.

System action

Processing terminates.

User response

See the topics in this information unit that describe the value, change the number so that it is within the range, and run the job again.

Module

IQCXREAD

**IQC4033E   NAME GREATER THAN 8
                  CHARACTERS**

Explanation

The name that was specified in the previous value is limited to 8 characters.

System action

Processing terminates.

User response

Correct the name value and run the job again.

Module

IQCXREAD

IQC4034E	NAME GREATER THAN 16 CHARACTERS

Explanation

The name that was specified in the previous value is limited to 16 characters.

System action

Processing terminates.

User response

Correct the name and run the job again.

Module

IQCXREAD

IQC4035E	ACTION DELETE AND UNLOCK MUTUALLY EXCLUSIVE

Explanation

Both DELETE and UNLOCK were specified in the ACTION= keyword; they are mutually exclusive, so a message can be deleted or unlocked but not both at the same time.

System action

Processing terminates.

User response

Correct the ACTION= keyword and run the job again.

Module

IQCXREAD

IQC4036I	NO QUEUETYPE= OR AREA=() WAS SPECIFIED.

Explanation

No QUEUETYPE= or AREA=() parameter was specified on a FUNCTION statement; therefore, the QUEUETYPE defaulted to the all of the queues.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCXREAD

IQC4037I	DEBUGGING NOW ENABLED

Explanation

You supplied the DEBUG control statement.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCXREAD

IQC4038E	TIMESTAMP: DATE MISSING PERIOD

Explanation

The data supplied after the D= of a UTC did not have a period as the third or fifth character. All dates must be in the form *yyyy.ddd* or *yy.ddd*.

System action

Processing terminates.

User response

Correct the data and run the job again.

Module

IQCXREAD

IQC4039E	TIMESTAMP: T= KEYWORD MISSING

Explanation

The second value of a UTC was not T=. UTC data must be in the form D=, T=.

System action

Processing terminates.

User response

Correct the UTC data and run the job again.

Module

IQCXREAD

IQC4040E	TIMESTAMP: TIME MISSING COLON

Explanation

The third or fifth character of the data after the T= value of a UTC was not a colon. The T= value must be in the form T=*hh:mm:*.

System action

Processing terminates.

User response

Correct the data and run the job again.

Module

IQCXREAD

IQC4041E	TIMESTAMP: TIME MISSING PERIOD

Explanation

The ninth character of the data after the T= value of a UTC was not a period (.). The T= value must be in the form T=*hh:mm:ss.*

System action

Processing terminates.

User response

Correct the data and run the job again.

Module

IQCXREAD

IQC4042E	TIMESTAMP: ZONE UNKNOWN SIGN

Explanation

The first character after the Z= of a UTC was not a *+, -,* or a number. The Z= value must be in the form Z=*shh*, where *s* is *+* or *-*, or Z=*hh*.

System action

Processing terminates.

User response

Correct the data and run the job again.

Module

IQCXREAD

IQC4043E	TIMESTAMP: ZONE MISSING COLON

Explanation

The second, third, or fourth character after the Z= of a UTC was not a colon (:). The Z= value must be in one of these forms:

- Z=*h:*
- Z=*hh:*
- Z=*shh:*

System action

Processing terminates.

User response

Correct the data and run the job again.

Module

IQCXREAD

IQC4044E	TIMESTAMP ZONE: UNKNOWN MINUTE OFFSET

Explanation

The data after the colon in the Z= of a UTC was not 00, 15, 30, or 45. Minutes in the Z= of a UTC must be specified in quarter hour increments.

System action

Processing terminates.

User response

Correct the UTC data after the colon and run the job again.

Module

IQCXREAD

IQC4045E   **INCREASE LRECL FOR LOGOUT DATA SET**

Explanation

The logical record length of the LOGOUT data set was not large enough to contain a selected IMS log record plus the IQCMRQPF macro prefix.

System action

Processing terminates.

User response

Increase the LRECL parameter on the LOGOUT DD statement and run the job again.

Module

IQCXREAD

IQC4046E   **UNKNOWN KEYWORD FOUND IN UTC TIMESTAMP**

Explanation

An unknown keyword was detected in a UTC timestamp. The valid keywords in the UTC timestamp are:

- D=
- T=
- Z=

System action

Processing terminates.

User response

Correct the data and run the job again.

Module

IQCXREAD

IQC4047E   **INCREASE LRECL FOR BROWSE DATA SET**

Explanation

The logical record length of the BROWSE data set was not large enough to contain a selected IMS log record plus the IQCMRQPF macro prefix.

System action

Processing terminates.

User response

Increase the LRECL parameter on the BROWSE DD statement and run the job again.

Module

IQCXREAD

IQC4048E   **INCREASE LRECL FOR SCRAPLOG DATA SET**

Explanation

The logical record length of the SCRAPLOG data set was less than the IMS log record chosen, plus the length of the IQCMRQPF macro prefix.

System action

Processing terminates.

User response

Increase the LRECL of the SCRAPLOG data set and run the job again. The SCRAPLOG data set should be allocated with the following DCB specifications: DCB=(LRECL=32756,BLKSIZE=32760,RECFM=VB).

Module

IQCXREAD

IQC4049E   **DUPLICATE INCLUDE PARAMETER**

Explanation

The INCLUDE keyword parameter on the previous record was used on a prior record.

System action

Processing terminates.

User response

Correct the INCLUDE statement and run the job again.

Module

IQCXREAD

IQC4050E   **DUPLICATE EXCLUDE PARAMETER**

Explanation

The EXCLUDE keyword parameter on the previous record was used on a prior record.

System action

Processing terminates.

User response

Correct the EXCLUDE statement and run the job again.

Module

IQCXREAD

IQC4051E   **UNKNOWN INCLUDE PARAMETER**

Explanation

The INCLUDE keyword parameter on the previous record was not recognized by IQCSELECT.

System action

Processing terminates.

User response

Correct the INCLUDE statement and run the job again.

Module

IQCXREAD

IQC4052E   **UNKNOWN EXCLUDE PARAMETER**

Explanation

The EXCLUDE keyword parameter on the previous record was not recognized by IQCSELECT.

System action

Processing terminates.

User response

Correct the EXCLUDE statement and run the job again.

Module

IQCXREAD

IQC4053E   **INCREASE LRECL FOR UNLOAD DATA SET**

Explanation

The logical record length of the UNLOAD data set was not large enough to contain a selected IMS log record plus the IQCMRQPF macro prefix.

System action

Processing terminates.

User response

Increase the LRECL parameter on the UNLOAD DD statement and run the job again.

Module

IQCXREAD

IQC4055E   **TIMESTAMP: DATE OUT OF RANGE**

Explanation

The date specified in a CHKPT control statement was not in the range of 1 to 366.

System action

Processing terminates.

User response

Correct the CHKPT control statement and run the job again.

Module

IQCXREAD

IQC4056E   **TIMESTAMP: ZONE OUT OF RANGE**

Explanation

The zone specified in a CHKPT control statement was not within the range of 0 to 12.

System action

Processing terminates.

User response

Correct the CHKPT control statement and run the job again.

Module

IQCXREAD

IQC4057I   **NO SELECT STATEMENTS PRESENT, ALL MESSAGES WILL BE PROCESSED**

Explanation

No SELECT statements were present in the QCFIN data set; therefore, all input IMS log records will be processed.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCXREAD

IQC4058E   **CHNGDEST UNKNOWN OPERAND**

Explanation

A value that was specified on the CHNGDEST statement was not recognized.

System action

Processing terminates.

User response

Correct the CHNGDEST statement and run the job again.

Module

IQCXREAD

IQC4059E   **DUPLICATE CHNGDEST OPERANDS**

Explanation

The same CHNGDEST value was entered more than once.

System action

Processing terminates.

User response

Remove the duplicate CHNGDEST statement value and run the job again.

Module

IQCXREAD

IQC4060E   **DUPLICATE MAXMSGCT OPERANDS**

Explanation

More than one MAXMSGCT statement was entered.

System action

Processing terminates.

User response

Remove the duplicate MAXMSGCT statement and run the job again.

Module

IQCXREAD

IQC4061E   **MAXMSGCT NO DIGITS FOUND**

Explanation

The value of the MAXMSGCT statement was not recognized.

System action

Processing terminates.

User response

Correct the value of the MAXMSGCT statement and run the job again.

Module

IQCXREAD

IQC4062E   **DUPLICATE PACING OPERANDS**

Explanation

More than one PACING statement was encountered.

System action

Processing terminates.

User response

Remove the duplicate PACING statement and run the job again.

Module

IQCXREAD

IQC4063E   **PACING MISSING PERIOD**

Explanation

The value on the PACING statement was not recognized.

System action

Processing terminates.

User response

Correct the data on the PACING statement and run the job again.

Module

IQCXREAD

IQC4064I   **TOTAL FOR THIS QUEUE:** *nnnnnnnn*

Explanation

The value of *nnnnnnnn* is the total number of messages for this queue.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCXREAD

IQC4065T   **QCF / IMS CONFLICT. QCF ABORTING.**

Explanation

IMS indicated to QCF that it encountered a catastrophic error and could not process any more QCF requests.

System action

Processing terminates.

User response

Follow your normal procedures for an IMS abend. If the condition persists, contact IBM Software Support.

Module

IQCXREAD

IQC4066E   **MAXMSGCT NUMBER TOO LARGE**

Explanation

The number that was specified on the MAXMSGCT control statement was larger than 9999999.

System action

Processing terminates.

User response

Reduce the number and run the job again.

Module

IQCXREAD

IQC4067E   **DUPLICATE MAXWAIT STATEMENTS**

Explanation

More than one MAXWAIT control statement was found.

System action

Processing terminates.

User response

Remove the duplicate MAXWAIT control statement and run the job again.

Module

IQCXREAD

IQC4068I   **NO MAXWAIT SPECIFIED; DEFAULTING TO 2 HOURS**

Explanation

No time was specified on the MAXWAIT control statement.

System action

Processing continues.

User response

If you intended to specify a time, correct the control statement and run the job again. If you did not intend to specify a time, no response is required.

Module

IQCXREAD

IQC4069E	MAXWAIT HOURS CANNOT EXCEED 24

Explanation

The first number on the MAXWAIT control statement, the hours, was greater than 24. The hours value must be between 0 and 24.

System action

Processing terminates.

User response

Correct the error and run the job again.

Module

IQCXREAD

IQC4070E	MAXWAIT MINUTES CANNOT EXCEED 60

Explanation

The second number, the minutes, on the MAXWAIT control statement was greater than 60. The minutes value must be between 0 and 60.

System action

Processing terminates.

User response

Correct the error and run the job again.

Module

IQCXREAD

IQC4071E	MAXWAIT SECONDS CANNOT EXCEED 60

Explanation

The third number, the seconds, on the MAXWAIT control statement was greater than 60. The seconds value must be between 0 and 60.

System action

Processing terminates.

User response

Correct the error and run the job again.

Module

IQCXREAD

IQC4072E	MAXWAIT TIME UNITS CANNOT BE MORE THAN 2 DIGITS LONG

Explanation

One of the numbers on the MAXWAIT control statement was more than two digits in length.

System action

Processing terminates.

User response

Reduce the number of digits on the offending MAXWAIT control statement and run the job again.

Module

IQCXREAD

IQC4073I	NO ACTION= SPECIFIED FOR RECOVER; DEFAULTING TO UNLOCK.

Explanation

The RECOVER function was specified without an ACTION= value, therefore QCF is defaulting to ACTION=UNLOCK.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCXREAD

IQC4074W   **UNKNOWN TRACE LEVEL. DEFAULTING TO 1**

Explanation

You supplied the TRACE control statement but the trace level was not recognized.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCXREAD

IQC4075I   **TRACING NOW ENABLED**

Explanation

You supplied the TRACE control statement.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCXREAD

IQC4076I   **TRACING NOW DISABLED**

Explanation

You supplied the TRACE control statement; however, the SNAPDUMP DD statement did not open.

System action

Processing continues.

User response

Supply the SNAPDUMP DD statement and run the job again.

Module

IQCXREAD

IQC4077I   **TOTAL PRIMARY MESSAGES FOR THIS QUEUE:** *nnnnn*

Explanation

The value of *nnnnn* is the number of primary entries found for this particular message queue.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCXRP61

IQC4078I   **DEBUGGING NOW DISABLED**

Explanation

You supplied the DEBUG control statement; however, the SNAPDUMP DD statement did not open.

System action

Processing continues.

User response

Supply the SNAPDUMP DD statement and run the job again.

Module

IQCXREAD

IQC4079I   **PACING NOW ENABLED.**

Explanation

You supplied the PACING control statement.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCXREAD

IQC4080I   **MAXWAIT NOW ENABLED.**

Explanation

You supplied the MAXWAIT control statement.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCXREAD

IQC4081I   **MAXMSGCT NOW ENABLED.**

Explanation

You supplied the MAXMSGCT control statement, thereby enabling MAXMSGCT.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCXREAD

IQC4082I   **UNKNOWN RECCOUNT. DEFAULTING TO NO**

Explanation

You did not supply the RECCOUNT control statement and it is therefore, defaulting to NO.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCXREAD

IQC4083I   **UNKNOWN SECONDARY BROWSE KEYWORD.**

Explanation

The CONTINUE keyword is missing from your BROWSE control statement.

System action

Processing continues.

User response

Correct the control statement and run the job again.

Module

IQCXREAD

IQC4084I   **UNKNOWN SECONDARY LOAD KEYWORD.**

Explanation

The CONTINUE keyword is missing from your LOAD control statement.

System action

Processing continues.

User response

Correct the control statement and run the job again.

Module

IQCXREAD

IQC4085I   **UNKNOWN SECONDARY QUERY KEYWORD.**

Explanation

The CONTINUE keyword is missing from your QUERY control statement.

System action

Processing continues.

User response

Correct the control statement and run the job again.

Module

IQCXREAD

IQC4086I   **UNKNOWN SECONDARY RECOVER KEYWORD.**

Explanation

The CONTINUE keyword is missing from your RECOVER control statement.

System action

Processing continues.

User response

Correct the control statement and run the job again.

Module

IQCXREAD

IQC4087I   **UNKNOWN SECONDARY RESET KEYWORD.**

Explanation

The CONTINUE keyword is missing from your RESET control statement.

System action

Processing continues.

User response

Correct the control statement and run the job again.

Module

IQCXREAD

IQC4088I   **UNKNOWN SECONDARY UNLOAD KEYWORD.**

Explanation

The secondary UNLOAD keyword was not recognized by the program.

System action

Processing continues.

User response

The CONTINUE keyword is missing from your UNLOAD control statement.

Module

IQCXREAD

IQC4089I   **UNKNOWN SECONDARY QUEUE SPACE NOTIFICATION KEYWORD**

Explanation

The CONTINUE keyword is missing from your Queue Space Notification control statement.

System action

Processing continues.

User response

Correct the control statement and run the job again.

Module

IQCXREAD

IQC4090I   **TOTAL MESSAGES FOR THIS QUEUE: *nnnnnnnn***

Explanation

The value of *nnnnnnnn* is the total number of entries found for this particular message queue.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCXREAD

IQC4091E   **UNKNOWN MSGTYPE / DESTYPE / SRCTYPE PARAMETER**

Explanation

The MSGTYPE, DESTYPE, or SRCTYPE parameter contains an invalid value.

System action

Processing terminates.

User response

See the descriptions of MSGTYPE, DESTYPE, or SRCTYPE in this information unit, correct the control statement, and run the job again.

Module

IQCXREAD

IQC4092E   **DUPLICATE WAIT STATEMENTS**

Explanation

The WAIT statement was entered more than once.

System action

Processing terminates.

User response

Remove the duplicate WAIT statement and run the job again.

Module

IQCXREAD

IQC4093E   **WAIT / MAXWAIT ARE MUTUALLY EXCLUSIVE**

Explanation

The WAIT and MAXWAIT statements were both entered.

System action

Processing terminates.

User response

Remove one of the statements and run the job again.

Module

IQCXREAD

IQC4094E   **INVALID WAIT FORMAT - *HH:MM:SS.IJKLMN***

Explanation

The format of the WAIT statement is invalid.

System action

Processing terminates.

User response

Correct the WAIT statement so that it conforms to the format that is shown in the message text and run the job again.

Module

IQCXREAD

IQC4095E   **INVALID WAIT FORMAT-*HH* GREATER THAN ONE**

Explanation

The format of the WAIT statement is invalid. The value of *HH* in the WAIT statement cannot be greater than 1.

System action

Processing terminates.

User response

Correct the WAIT statement and run the job again.

Module

IQCXREAD

IQC4096I   **NO WAIT VALUE SPECIFIED; DEFAULTING TO 0**

Explanation

The WAIT statement does not specify a wait value.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCXREAD

IQC4097I   **WAIT NOW ENABLED; WAIT TIME SET TO *HH:MM:SS.IJKLMN***

Explanation

WAIT is enabled; there will be wait time after a message is inserted.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCXREAD

IQC4098I   **WAIT ENABLED, PACING PARAMETERS IGNORED**

Explanation

WAIT is enabled; the values set in the PACING statement are ignored.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCXREAD

IQC4099E   **WAIT WITHOUT PACING SPECIFIED**

Explanation

The WAIT statement can be specified only with the PACING statement.

System action

Processing terminates.

User response

Remove the WAIT statement, or add a PACING statement; then run the job again.

Module

IQCXREAD

IQC4100T   *xxxxxxxx* **DD CARD FAILED TO OPEN. TERMINATING**

Explanation

The required DD statement, that is represented by *xxxxxxxx*, failed to open.

System action

Processing terminates.

User response

Supply the missing DD statement and run the job again.

Module

IQCLOAD0

IQC4101I   **SEARCH FOR BASE CHECKPOINT**

Explanation

IQCSELCT is reading the LOGIN data set for the specified checkpoint.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCXREAD

IQC4103I   **BASE CHECKPOINT FOUND AT LOGICAL RECORD *nnnnnnnn***

Explanation

The base checkpoint the was specified in a CHKPT control statement was found at the logical IMS log record that is shown in *nnnnnnnn*.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCXREAD

IQC4105I   **MESSAGES SELECTED: *nnnnnnnn***

Explanation

The value of *nnnnnnnn* is the number of messages that IQCSELCT selected for processing.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCLOAD0

IQC4107I   **CHECKPOINT** *nnnnn* **- D=***yyyy.ddd***, T=***hh:mm:ss.thmiju***, Z=***shh:mm*

Explanation

The value of *nnnnn* is the checkpoint number. D=*yyyy.ddd* is the checkpoint date stamp. T=*hh:mm:ss.thmiju* is the checkpoint time stamp. Z=*shh:mm* is the checkpoint zone.

System action

Processing continues.

User response

None. This message is informational.

Module

IQCXREAD

IQC4111I   **ASSEMBLY DATE AND STATUS OF** *ssssssss*

Explanation

This informational message appears if the DEBUG option was chosen. The value of *ssssssss* is a CSECT within IQCSELCT. One message appears for each CSECT within IQCSELCT.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCXREAD

IQC4112I   **CHECKPOINT** *nnnnn* **- END OF CHECKPOINT AT LOGICAL RECORD** *nnnnnnn*

Explanation

The end of checkpoint that is shown in *nnnnn* was found at logical IMS log record shown in *nnnnnnn*.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCXREAD

IQC4121E   **ERRORS IN CARD AREAA FUNCTION QC_LTBL**

Explanation

Card AREAA for the function QC_LTBL has an invalid format.

System action

Processing terminates.

User response

Correct the control statement and run the job again.

Module

IQCXREAD

IQC4122E   **ERRORS IN CARD AREAB FUNCTION QC_LTBL**

Explanation

Card AREAB for the function QC_LTBL has an invalid format.

System action

Processing terminates.

User response

Correct the control statement and run the job again.

Module

IQCXREAD

IQC4123E   **ERRORS IN CARD AREAC FUNCTION QC_LTBL**

Explanation

Card AREAC for the function QC_LTBL has an invalid format.

System action

Processing terminates.

User response

Correct the control statement and run the job again.

Module

IQCXREAD

IQC4124E   **ERRORS IN CARD AREAD FUNCTION QC_LTBL**

Explanation

Card AREAD for the function QC_LTBL has an invalid format.

System action

Processing terminates.

User response

Correct the control statement and run the job again.

Module

IQCXREAD

IQC4125E   **ERRORS IN CARD JOBSCARD FUNCTION QC_LTBL**

Explanation

Card JOBSCARD for the function QC_LTBL has an invalid format.

System action

Processing terminates.

User response

Correct the control statement and run the job again.

Module

IQCXREAD

IQC4126E   **ERRORS IN CARD PERCENTS FUNCTION QC_LTBL**

Explanation

Card PERCENTS for the function QC_LTBL has an invalid format.

System action

Processing terminates.

User response

Correct the control statement and run the job again.

Module

IQCXREAD

IQC4127W   **DEFAULT VALUES SET FOR THRESHOLDS**

Explanation

There were no values for thresholds on input; default values were set.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCXREAD

IQC4128E   **MISSING-INCORRECT CARDS-TABLE NOT INSERTED TO IMS**

Explanation

Function QC_LTBL was not executed because of missing or incorrect input values.

System action

Processing terminates.

User response

Correct the control statement and run the job again.

Module

IQCXREAD

IQC4129E   **FUNCTION QC_LTBL / QC_QTBL-BAD RETURN CODE FROM IMS**

Explanation

Function LOAD / QUERY overflow protection table failed.

System action

Processing terminates.

User response

Ensure that the function is running in a nonshared-queues environment and run the job again.

Module

IQCXREAD

IQC4130E   **LTERM AND TMEMBER / TPIPE ARE EXCLUSIVE**

Explanation

The LTERM and TMEMBER / TPIPE values are exclusive for the change destination. Only one can be specified at a time.

System action

Processing terminates.

User response

Correct the control statements and run the job again.

Module

IQCXREAD

IQC4131E   **TMEMBER OR TPIPE IS REQUIRED**

Explanation

The TMEMBER and TPIPE values are both required to be present for the change destination to the OTMA destination.

System action

Processing terminates.

User response

Correct the control statements and run the job again.

Module

IQCXREAD

IQC4132E   **BLANK NAME, INVALID IN CHNGDEST**

Explanation

A TMEMBER or TPIPE value in a CHNGDEST statement is invalid.

System action

Processing terminates.

User response

Correct the control statements and run the job again.

Module

IQCXREAD

IQC4158E   **TOIPARMS UNKNOWN OPERAND**

Explanation

IMS Queue Control Facility has detected an unknown value for TOIPARMS.

System action

Processing terminates.

User response

Correct the errant value and run the job again.

Module

IQCXREAD

IQC4159E   **DUPLICATE TOIPARMS OPERANDS**

Explanation

IMS Queue Control Facility has detected duplicate values for TOIPARMS

System action

Processing terminates.

User response

Remove the duplicate value and run the job again.

Module

IQCXREAD

IQC4160E   **TOPDSTWTO (TD) IS VALID ONLY WITH FUNCTION BROWSE**

Explanation

Parameter TOPDSTWTO (TD) can only be used with the BROWSE function.

System action

Processing terminates.

User response

Remove TOPDSTWTO (TD) from the current function statement and run the job again.

Module

IQCXREAD

IQC4161E	TOPLONGMSGWTO is valid only with function QUERY

Explanation

TOPLONGMSGWTO was detected with a function other than QUERY.

System action

Processing continues.

User response

Remove TOPLONGMSGWTO from the current function statement and run the job again.

Module

IQCXREAD

IQC4162E	TOPSHORTMSGWTO is valid only with function QUERY

Explanation

TOPSHORTMSGWTO was detected with a function other than QUERY.

System action

Processing terminates.

User response

Remove TOPSHORTMSGWTO from the current function statement and run the job again.

Module

IQCXREAD

IQC4163E	TOPALLMSGWTO is valid only with function QUERY

Explanation

TOPALLMSGWTO was detected with a function other than QUERY.

System action

Processing terminates.

User response

Remove TOPALLMSGWTO from the current function statement and run the job again.

Module

IQCXREAD

IQC4164E	TOPLASTACTIVWTO is valid only with function QUERY

Explanation

TOPLASTACTIVWTO was detected with a function other than QUERY.

System action

Processing terminates.

User response

Remove TOPLASTACTIVWTO from the current function statement and run the job again.

Module

IQCXREA

IQC4165E	TOPSM/TOPLM IS NOT VALID FOR SHARED QUEUES-SET TO TOPAM

Explanation

A specification of either TOPSM or TOPLM was detected for a shared-queues environment. TOPSM or TOPLM are invalid for shared queues, so the default setting of TOPAM is in effect.

System action

Processing terminates.

User response

Remove TOPSM or TOPLM, or TOPSM and TOPLM from the current control statement and run the job again.

Module

IQCXREAD

IQC4170E   INVALID AGE PARAMETER

Explanation

An invalid AGE parameter was detected.

The AGE parameter exceeded 365 when AGEUNITS=DAYS was specified or 31536000 when AGEUNITS=SECONDS was specified.

System action

Processing terminates.

User response

See the description of the AGE parameter in "FUNCTION statement" on page 130, enter a valid AGE parameter, and run the job again.

Module

IQCXREAD

IQC4171E   INVALID MAXMSGCT PARAMETER

Explanation

An invalid MAXMSGCT parameter was detected. MAXMSGCT input is equal to zero.

System action

Processing terminates.

User response

See the description of the MAXMSGCT parameter in this information unit, enter a valid value for the MAXMSGCT parameter, and run the job again.

Module

IQCXREAD

IQC4172E   INVALID TOP DESTINATION WTO PARAMETER

Explanation

The following top destination sets of parameters are mutually exclusive:

- TOPSHORTMSGWTO / TOPSM
- TOPLONGMSGWTO / TOPLM
- TOPLASTACTIVWTO / TOPLA
- TOPALLMSGWTO / TOPAM

Specifying more than one of these sets of parameters simultaneously, results in this error condition.

System action

Processing terminates.

User response

Remove the extraneous set of parameters and run the job again.

Module

IQCXREAD

IQC4173E   INVALID MULTIPLE EXCLUSIVE PARAMETERS, PLEASE CHECK USER GUIDE.

Explanation

The MAXMSGCNT and CHNGDEST parameters cannot appear on a control card together.

System action

Processing terminates.

User response

Remove the mutually exclusive parameters and run the job again.

Module

IQCXREAD

IQC4174E   MAXWAIT WITHOUT PACING SPECIFIED

Explanation

Specifying a MAXWAIT statement requires the PACING statement.

System action

Processing terminates.

User response

Supply a PACING statement when specifying the MAXWAIT statement

Module

IQCXREAD

IQC4175E   **INVALID AGEUNITS PARAMETER**

Explanation

An invalid AGEUNITS parameter was detected.

System action

Processing terminates.

User response

See the description of the AGEUNITS parameter in "FUNCTION statement" on page 130, enter a valid AGEUNITS parameter, and run the job again.

Module

IQCXREAD

IQC4998I   *jobname* **RC=***nn* **AIBRC=***nn* **AIBRS=***nn*

Explanation

The job name is displayed in *jobname*. The return code is shown as the RC=*nn* value. The AIB return code is shown as AIBRC=*nn* and the AIB reason code is shown as AIBRS=*nn*.

System action

Processing continues.

User response

See the description of the AIB reason code in this information unit.

Module

IQCZCVC0, IQCBCVC0

IQC4999E   *jobname* **RC=***nn* **AIBRC=***nn* **AIBRS=***nn* **MSGO=***xx*

Explanation

The job name is displayed in *jobname*. The return code is shown as the RC=*nn* value. The AIB return code is

shown as AIBRC=*nn* and the AIB reason code is shown as AIBRS=*nn*. MSGO=*xx* is message out to TSO.

System action

Processing terminates.

User response

To determine the cause of the error, see the report in the QCFPRINT data set or the description of the AIB reason code in this information unit, or both.

Correct the error and run your job again.

Module

IQCZCVC0, IQCBCVC0

IQC5000I   **CQS BROWSE ROUTINE**

Explanation

This message is printed when control is passed to IQCBROWS by IQCXCTRL.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCBROWS

IQC5001I   **END OF BROWSE**

Explanation

This message is printed when IQCBROWS passes control back to IQCXCTRL.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCBROWS

IQC5002I   **ASSEMBLY DATE AND STATUS OF** *ssssssss*

Explanation

This informational message appears if the DEBUG option was chosen. The value of *ssssssss* is a CSECT within IQCBROWS. One message appears for each CSECT within IQCBROWS.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCBROWS

IQC5003I	MESSAGES SELECTED: *nnnnnnnn*

Explanation

The value of *nnnnnnnn* is the number of messages IQCBROWS found on the selected message queues.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCBROWS

IQC5004I	BROWSE DDCARD FAILED TO OPEN; ACTION=*REPORTONLY* WILL BE USED

Explanation

The BROWSE DD statement was not in the input job stream. The BROWSE function will proceed as if ACTION=*REPORTONLY* was specified.

System action

Processing continues.

User response

If the BROWSE DD statement was intentionally omitted, none. If the BROWSE DD statement was unintentionally omitted, supply a BROWSE DD and run the job again.

Module

IQCBROWS

IQC5005I	BROWSE DDCARD IS DD DUMMY; ACTION=*REPORTONLY* WILL BE USED

Explanation

The BROWSE DD statement was specified as DUMMY. The BROWSE function will proceed as if ACTION=*REPORTONLY* was specified.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCBROWS

IQC5006I	TOP *NN* DESTINATIONS

Explanation

Header line with selected number of top destinations to be displayed.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCBROWS

IQC5007I	DESTINATION Q-TYPE PRIMARY SECONDARY

Explanation

Destination name and queue type for the destination and counters of primary and secondary messages for this destination.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCBROWS

IQC5008I   **BROWSE DATASET FULL, STATISTIC NOT COMPLETE IN QCFPRINT**

Explanation

This is an informational message indicating that a BROWSE request was interrupted because the BROWSE data set does not have sufficient space. The statistics that appear in QCFPRINT are not complete.

System action

Processing ends.

User response

None; information only.

Module

IQCXSEL6

IQC5500I   **CQS RECOVER ROUTINE**

Explanation

This message is printed when control is passed to IQCRECOV by IQCXCTRL.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCRECOV

IQC5501I   **END OF RECOVER**

Explanation

This message is printed when IQCRECOV passes control back to IQCXCTRL.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCRECOV

IQC5503I   **MESSAGES SELECTED:** *nnnnnnnn*

Explanation

The value of *nnnnnnnn* is the number of messages that IQCRECOV processed from the cold queue.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCRECOV

IQC5504I   **RECOVER DDCARD FAILED TO OPEN; ACTION=*REPORTONLY* WILL BE USED**

Explanation

The RECOVER DD statement was not found in the job stream; therefore, IQCRECOV could not copy data from the cold queue to the RECOVER data set.

System action

Processing continues.

User response

If the RECOVER DD statement was intentionally omitted, none. If the RECOVER DD statement was unintentionally omitted, supply a RECOVER DD statement and run the job again.

Module

IQCRECOV

IQC5505I   **RECOVER DDCARD IS DD DUMMY; ACTION=*REPORTONLY* WILL BE USED**

Explanation

The RECOVER DD statement was specified as DUMMY. The RECOVER function will proceed as if ACTION=*REPORTONLY* was specified.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCRECOV

IQC5507T	NO MESSAGES PASSED TO IMS HAVE BEEN COMMITTED

Explanation

None of the messages that were passed to IMS have been committed.

System action

Processing terminates.

User response

None. This message is for informational purposes only.

Module

IQCXREAD

IQC5510I	SPECIFY A QCF WAIT OR QCF MAXWAIT STATEMENT

Explanation

When using the PACING statement, it is recommended that you specify a QCF WAIT or QCF MAXWAIT statement.

System action

Processing continues.

User response

Supply a QCF WAIT or QCF MAXWAIT statement.

Module

IQCXPACE

IQC5512I	REPLACE QCF MAXWAIT STATEMENT WITH QCF WAIT STATEMENT

Explanation

A negative time differential might occur with the use of the QCF MAXWAIT statement and a PACING statement..

System action

Processing terminates.

User response

Replace the QCF MAXWAIT statement with a QCF WAIT statement.

Module

IQCXPACE

IQC5513E	TOIPARMS missing XCFGROUP parameter or unknown operand specified

Explanation

The value on the TOIPARMS statement was not recognized.

System action

Processing terminates.

User response

Specify XFCGROUP= parameter or check the spelling of XCFGROUP= parameter.

Module

IQCXREAD

IQC5514E	XCF group name missing for XCFGROUP= parameter

Explanation

No value was specified for the XCFGROUP= parameter.

System action

Processing terminates.

User response

Specify a valid XCF group name for the XCFGROUP= parameter.

Module

IQCXREAD

IQC5515E   **Duplicate TOIPARMS statement was specified**

Explanation

More than one TOIPARMS statement was encountered.

System action

Processing terminates.

User response

Remove the duplicate TOIPARMS statement.

Module

IQCXREAD

IQC5516I   **NO MESSAGES PASSED TO IMS HAVE BEEN COMMITTED**

Explanation

This message is issued when a Syncpoint Error is encountered during the LOAD step of a RECOVERDM job, or during a LOAD job without the use of the PACING statement. This message indicates that no messages have been committed to the IMS queue.

System action

Processing terminates.

User response

Restart IMS and retry the RECOVERDM or LOAD job.

Module

IQCXCTRL

IQC5517I   **DISABLESTOP keyword specified. /STO commands will not be issued**

Explanation

You supplied the DISABLESTOP keyword for the TOIPARMS statement. Therefore, /STO commands will not be issued.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCXREAD

IQC5518I   **ENABLESTOP keyword specified. /STO commands will be issued**

Explanation

You supplied the ENABLESTOP keyword for the TOIPARMS statement. Therefore, /STO commands will continued to be issued.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCXREAD

IQC5519E   **ENABLESTOP or DISABLESTOP keyword was expected**

Explanation

The value specified was not recognized. ENABLESTOP or DISABLESTOP was expected.

System action

Processing terminates.

User response

Correct the ENABLESTOP or DISABLESTOP keyword.

Module

IQCXREAD

IQC6001I   **QCF EXTENSION [ENABLED | DISABLED].**

Explanation

QCF extensions are either enabled, active, and ready to process requests; or disabled and inactive because an extensions initialization failure occurred.

System action

If the state is ENABLED, then QCF Extension initialization was successful, and QCF Extensions processing continues.

If the state is DISABLED, QCF Extension is terminated.

User response

If the state is ENABLED, no user response required.

If the state is DISABLED, you will need to determine what failed during the initialization process. See the IQC6nnnE messages in the log to determine what caused the initialization failure and correct the problem.

Module

IQCCSCH0, IQCCIN00

IQC6003E   **QCF EXTENSION HAS FAILED TO ESTABLISH AN ESTAE ROUTINE.**

Explanation

QCF was unable to establish an ESTAE routine.

System action

Processing terminates.

User response

QCF will be disabled. Report the problem to IBM Software Support.

Module

IQCCSCH0

IQC6004E   **RECEIVED AN INVALID AWE FUNCTION OF *function*.**

Explanation

QCF has received an invalid AWE function for the value that is indicated by *function*.

System action

Processing continues.

User response

QCF will be disabled. Report the problem to IBM Software Support.

Module

IQCCSCH0

IQC6005E   **MVS NAME TOKEN GET REQUEST FAILED.**

Explanation

The GET request for an MVS name token failed.

System action

Processing terminates.

User response

QCF will be disabled. Report the problem to IBM Software Support.

Module

IQCCSCH0

IQC6006E   **UNABLE TO CREATE [CIB | CLIENT THREAD].**

Explanation

QCF was not able to create either a CIB or a client thread as indicated in the message text.

System action

Processing continues.

User response

None, if the problem persists report the problem to IBM Software Support.

Module

IQCCSCH0

IQC6007E   **UNABLE TO ENQUEUE THE CLIENT AWE.**

Explanation

QCF has failed to enqueue the client AWE.

System action

Processing terminates.

User response

QCF will be disabled. If the problem persists, report the problem to IBM Software Support.

Module

IQCCSCH0

IQC6008I   **QCF EXTENSION HAS TERMINATED.**

Explanation

QCF extensions have stopped.

System action

Processing continues.

User response

None. QCF terminated and restart will begin.

Module

IQCCSCH0

IQC6009I   **QCF EXTENSION UNABLE TO NOTIFY CLIENT** *client_name* **OF TERMINATION.**

Explanation

QCF extensions were unable to notify the client that is indicated by the value of *client_name* of a termination condition.

System action

Processing terminates.

User response

None, this message is for informational purposes only.

Module

IQCCSCH0, IQCCSCH8, IQCCSCH9, IQCCSCHA

IQC6018I   **EXTENSION RECONNECTED TO SCI, IMSID=***imsid*

Explanation

This is an informational message issued by IMS extension indicating that SCI is terminated.

System action

Processing continues.

User response

None; information only.

Module

IQCCXGXA

IQC6019I   **EXTENSION DISCONNECTED FROM SCI, IMSID=***imsid*

Explanation

This is an informational message indicating that SCI is started.

System action

Processing is discontinued until the SCI is available. After the SCI is restarted, IMS extension will register automatically and continue processing. Message IQC608I is issued when IMS extension has again registered to the SCI.

User response

None; information only.

Module

IQCCXGXA

IQC6100   **IQCQSTBL IS IN ERROR, REASON CODE=***n***. DEFAULT TABLE IS USED.**

Explanation

n is one of the following reason codes:

1. Table length is incorrect.
2. Percent the was specified for IQC6101 warning message is incorrect. Valid range is 01-99.
3. Value the was specified for EXPRESS PCB is incorrect.
4. Control entry percent value was < 01.
5. Control entry percent value was > 100.
6. Control entry action code was less than the minimum value.

7. Control entry action code was greater than the minimum value.

System action

Processing continues.

User response

Run IQCTST00 to verify the content of IQCQSTBL.

Module:

IQC6101W	QUEUE SPACE USAGE WARNING. QUEUE DATA SET *data_set* USAGE COUNT IS *nnnnnnnn* WHICH IS *pp%* AS SPECIFIED IN IQCQSTBL

Explanation

The variable values in the message have the following meanings:

- *data_set* is LMSG or SMSG (large message queue or short message queue).
- *pp* is from 1 - 99, and is the value the was specified in IQSQSTBL.
- *nnnnnnnn* is *pp* % of the total space in the queue data set.

This message is displayed when the User Threshold (UT) that you defined is crossed. It does not have to be greater than the upper threshold.

System action

Processing continues.

User response

None. This message is informational.

Module:

IQC6104E	ABEND *xxxx* IN QCF EXTENSION MODULE *yyyyyyyy*, PSW=*psw1 psw2*

Explanation

An abend occurred in IMS Queue Control Facility extensions under IMS control region CTX TCB. The thread initialization module (either IQCCSCH0 or IQCCCVC0) establishes an ESTAE to protect its processing. This ESTAE attempts to retry from the abend and to continue cleaning up global resources. Message IQC6104E is issued to alert the operator that an abend occurred.

In the message text:

xxxx
    The abend code.

    For system abends, the format of code is S*xxx*, where *xxx* is the 3-digit abend code in hexadecimal.

    For user abends, the format of code is U*dddd*, where *dddd* is the 4-digit abend code in decimal.

yyyyyyyy
    The module name where abend occurs.

psw1
    The first word of the PSW at abend.

psw2
    The second word of the PSW at abend.

System action

Processing terminates.

User response

If you are unable to resolve the problem, contact IBM Software Support.

Module

IQCCSCH0 or IQCCCVC0

IQC6105I	SDUMP FAILED FOR *xxxx* ABEND, RC=*rc*, RSN=*rsn*

Explanation

ESTAE routine (IQCCEST0) issued an SDUMP call to MVS to produce a dump of the address space after an abend occurred, but the SDUMP was not successful.

In the message text:

xxxx
    The abend code for which the dump was taken.

    For system abends, the format of code is S*xxx*, where *xxx* is the 3-digit abend code in hexadecimal.

    For user abends, the format of code is U*dddd*, where *dddd* is the 4-digit abend code in decimal.

rc
    The return code from the MVS SDUMP macro.

rsn
    The reason code from the MVS SDUMP macro.

System action

Processing continues.

User response

If you are unable to resolve the problem, contact IBM Software Support.

Module

IQCCEST0

IQC6106I   **DAE SUPPRESSED DUMP FOR** *xxxx* **ABEND.**

Explanation

ESTAE routine (IQCCEST0) issued an SDUMP call to MVS to produce a dump of the address space after an abend, but the SDUMP was suppressed by MVS dump analysis and elimination (DAE).

In the message text:

xxxx
　　The abend code for which the dump was taken.

　　For system abends, the format of code is S*xxx*, where *xxx* is the 3-digit abend code in hexadecimal.

　　For user abends, the format of code is U*dddd*, where *dddd* is the 4-digit abend code in decimal.

System action

Processing continues.

User response

If you are unable to resolve the problem, contact IBM Software Support.

Module

IQCCEST0

IQC6200I   **THE NUMBER OF RECORDS IN THE [LMSG | SMSG | QBLK] DATA SET HAS EXCEEDED THE UPPER THRESHOLD.**

Explanation

The number of records in the data set that is indicated by value of LMSG, SMSG, or QBLK in the message text, has exceeded the upper threshold that you set. LMSG is the large message queue data set. SMSG is the short message queue data set. QBLK is the queue blocks data set. This message is displayed when the upper threshold (CD) that you defined is crossed.

Messages might be queuing excessively for one of the following reasons:

- One or more transactions that have stopped IMS programs
- One or more terminals is stopped or not signed on, such as a printer
- A terminal destination that was not defined to the system is causing messages to be inserted to an invalid destination
- Either an executing transaction or an intelligent input device is looping and inserting messages
- There might not be enough IMS queue space allocated for the current work load

System action

Processing continues.

User response

You might want to reduce the queue usage and avoid a possible shutdown. To determine the effect of this action, examine the percentage of use that is displayed by the **/DIS POOL** command.

Check the console to see if any IQC6000 series messages have been issued indicating that an action has been taken to alleviate the condition. This would include starting jobs that would unload part or all of the current committed messages on the queue.

Determine if enough dependent regions are available to handle the work load.

Determine if any programs are stopped that should not be stopped. Change the threshold percentages with the QCF TSO Client, define jobs to perform unloads of messages, or change area actions to prevent the system from terminating because of insufficient queue space.

Module

IQCQMRI0

IQC6201I   **THE NUMBER OF RECORDS IN THE [LMSG | SMSG | QBLK] DATA SET IS NOW BELOW THE LOWER THRESHOLD.**

Explanation

The number of records in either the LMSG, SMSG, or QBLK data set, which is indicated in the message text, has gone below the lower threshold that you set. LMSG is the large message queue data set. SMSG is the short message queue data set. QBLK is the queue blocks data set.

This message is displayed when the lower threshold (AB) that you defined is crossed.

System action

Processing continues.

User response

Determine if enough dependent regions are available to handle the work load. Determine if any programs are stopped that should not be stopped. Using the QCF TSO Client do one of the following tasks:

- Change the threshold percentages.
- Define jobs to perform unloads of messages.
- Change area actions to prevent the system from terminating because of insufficient use of queue space.

Module

IQCCIN00

IQC6202E	UNABLE TO OBTAIN *storage_type* STORAGE.

Explanation

QCF was unable to obtain the type of storage that is indicated by the value of *storage_type* in the message text.

The storage types that can be indicated by *storage_type* are:

- Dynamic storage
- AWE block storage
- CQAB block storage
- CQAT block storage
- CQTT block storage
- QSAV block storage
- PROCLIB grammar storage
- Parsing PROCLIB member storage
- XMT table storage
- CIB block storage
- SEND buffer storage
- DLI work area storage

System action

Processing terminates.

User response

Report the problem to IBM Software Support.

Module

IQCCIN00, IQCCSCH0, IQCCIN10, IQCCCVC0, IQCCPAR0

IQC6203E	UNABLE TO LOAD QCF EXTENSION MODULE *module_name*

Explanation

QCF was unable to load the QCF extensions module that is indicated in the message text by the value of *module_name*.

System action

Processing terminates.

User response

Report the problem to IBM Software Support.

Module

IQCCIN00

IQC6204E	MVS NAME TOKEN CREATE REQUEST FAILED.

Explanation

The attempt to create an MVS name token failed.

System action

Processing terminates.

User response

Report the problem to IBM Software Support.

Module

IQCCIN00

IQC6205E	ERROR [READING | PARSING] PROCLIB MEMBER *member_name*

Explanation

An error was encountered while attempting to read or parse the PROCLIB member that is indicated by the value of *member_name*.

System action

Processing continues.

User response

Review the other message that starts with IQC to determine why the parsing of the configuration file failed. Correct the PROCLIB member error and run the job again.

Module

IQCCIN00

IQC6206I	*jobname* - QCF/QSN WAIT QUEUE HAS WAITER.

Explanation

The job name that is indicated by *jobname* in the message text is being waited. The job was waited because the AREA usage percentage has been exceeded and you had selected the WAIT action.

System action

Processing continues.

User response

Determine which queue is being exceeded, correct the problem by starting the program, starting more dependent regions, starting printer, starting the job to unload queue, etc. Then go to the QCF TSO Client connection and request a resume of the waited job.

Module

IQCQMRE0

IQC6210E	UNABLE TO LOCATE [IMS SSCT BLOCK	DISPATCHER WORK AREA].

Explanation

QCF was unable to locate the resource type that is indicated in the message text; either the IMS SSCT block or the dispatcher work area.

System action

Processing terminates.

User response

None, QCF is disabled. If the problem persists, report it to IBM Software Support.

Module

IQCCIN00, IQCCSCH0

IQC6212E	UNABLE TO CREATE INITIALIZATION TASK.

Explanation

QCF was unable to create the initialization task.

System action

Processing terminates.

User response

None, QCF is disabled. If the problem persists, report it to IBM Software Support.

Module

IQCCIN00, IQCCSCH0

IQC6213E	ERROR POSTING INITIALIZATION TASK.

Explanation

QCF was unable to post the initialization task.

System action

Processing terminates.

User response

None, QCF is disabled. If the problem persists, report it to IBM Software Support.

Module

IQCCIN00, IQCCSCH0

IQC6215E	UNSUPPORTED IMS RELEASE, R=*ims_release*.

Explanation

The release of IMS that is indicated by the value of *ims_release* is not supported by this version of QCF.

System action

Processing terminates.

User response

Use a supported version of IMS.

Module

IQCCIN00

IQC6221W   DEFINING A 64 KB HIOP BUFFER SIZE IS RECOMMENDED FOR QCF FUNCTIONS

Explanation

A 64 KB HIOP buffer size is not defined. When using IMS Queue Control Facility functions, it is recommended that you define an HIOP buffer of 64 KB. The IMS Queue Control Facility functions are still available without a 64 KB HIOP buffer, but performance of the IMS control region might be affected.

System action

Processing continues.

User response

If you want to avoid additional processor usage of IMS to handle requests for HIOP buffers and to improve performance while IMS Queue Control Facility functions are run, specify an HIOP buffer of 64 KB in the DFSSPM*xx* member of the IMS PROCLIB data set, and then restart IMS.

Module

IQCCSCH0

IQC6400E   Line 1: AN ERROR OCCURRED PARSING QCF EXTENSION CONFIG FILE. Line 2: AT LINE *line_number*, CHARACTER *char_value*. or AT CHARACTER *char_value*. Line 3: FAILING TEXT: *"text_data"*. Line 4: *line4_text*

Explanation

QCF has encountered a parsing error at the line number that is indicated by the value of *line_number*, if it appears. The parsing error occurred at the character that is indicated by the value of *char_value* and the text that is indicated by the value of *"text_data"*.

Line 4 supplies you with more detail about the parsing error that is described in lines 1 through 3 of the message text. The text for line 4 (*line4_text*) can be one of the following:

- INVALID KEYWORD DETECTED
- UNKNOWN POSITIONAL PARAMETER
- "=" ENCOUNTERED WHEN "(" EXPECTED
- EARLY END OF INPUT DATA
- KEYWORD ENCOUNTERED WHEN VALUE EXPECTED

- NUMERIC VALUE OUTSIDE OF LEGAL RANGE
- DECIMAL NUMBER CONTAINED NONDECIMAL DIGITS
- HEXADECIMAL NUMBER CONTAINED NONHEX DIGITS
- UNKNOWN KEYWORD VALUE DETECTED
- DUPLICATE KEYWORD PARAMETER DETECTED
- A REQUIRED PARAMETER WAS OMITTED
- CHARACTER VALUE WAS TOO LONG FOR PARAMETER
- A REQUIRED VALUE WAS OMITTED

System action

Processing terminates.

User response

Correct the error that is indicated by *char_value* and *"text_data"*. Run the job again.

Module

IQCCPAR0

IQC6600E   ZERO INPUT BUFFER ADDRESS PASSED.

Explanation

QCF internal error. An input buffer address of zero was passed to QCF.

System action

Processing terminates.

User response

Specify a nonzero input buffer address and run the job again or contact IBM software support.

Module

IQCCRDP0, IQCCPAR0

IQC6602E   ZERO OUTPUT BUFFER [ADDRESS | LENGTH | MEMBER NAME ADDRESS] PASSED.

Explanation

QCF internal error. A zero value for either the output buffer address, length, or member name address was passed to QCF.

System action

Processing terminates.

User response

Rerun the job passing a nonzero value for either the output buffer address, length, or member name; or contact IBM Software Support.

Module

IQCCRDP0, IQCCPAR0

IQC6603E	UNABLE TO OBTAIN STORAGE FOR A BUFFER TO HOLD THE *resource_type*, MEMBER NAME=*member_name*.

Explanation

QCF was unable to obtain buffer storage to hold the resource type that is indicated by the value of *resource_type* for the member name that is indicated by the value of *member_name*.

The possible values for *resource_type* are:

- 24 BIT WORK AREA
- PDS NAMED MEMBER BLOCK DATA
- PDS NAMED MEMBER DATA

System action

Processing terminates.

User response

Check for availability of buffer storage and run the job again.

Module

IQCCRDP0

IQC6604E	UNABLE TO OPEN PDS.

Explanation

QCF was unable to open a PDS that contains the QCF configuration file, that one of the libraries in the PROCLIB definition.

System action

Processing terminates.

User response

Check that the PDS is valid and can be opened and run the job again.

Module

IQCCRDP0

IQC6605E	DATA SET WAS NOT FIXED FORMAT.

Explanation

QCF detected that the data set containing the QCF configuration data was of a variable format when it needed to be a fixed format data set.

System action

Processing terminates.

User response

Ensure that a fixed format partitioned data set is specified for the PROCLIB that contains the QCF configuration data.

Module

IQCCRDP0

IQC6606E	RIGHT MARGIN WAS NEGATIVE OR >= DATA SET LRECL.

Explanation

The right margin that you specified with QCF for an internal Parsing call was either a negative value or it was greater than or equal to the data set logical record length. This is an internal error or the PROCLIB that contains the QCF configuration data is not defined with an LRECL of 80.

System action

Processing terminates.

User response

Ensure that the definition of the PROCLIB has an LRECL of 80. If an LRECL of 80 was specified report the problem to IBM service.

Module

IQCCRDP0

IQC6607E	PDS MEMBER NAME IN THE QCF CONFIG FILE WAS NOT

FOUND IN THE PDS, MEMBER NAME=*member_name*.

Explanation

The PDS member name that is specified in the QCF configuration file and that is also indicated by the value of *member_name*, was not found in the PDS.

System action

Processing terminates.

User response

Specify a member name in the QCF configuration file that exists in the PDS.

Module

IQCCRDP0

IQC6608E	READ ERROR WHILE TRYING TO READ MEMBER *member_name*

Explanation

QCF encountered an error while attempting to read the contents of the member that is indicated by the value of *member_name*.

System action

Processing terminates.

User response

Ensure that the member that is indicated by the value of *member_name* is a valid, uncorrupted member. Correct the member content error and resubmit the job.

Module

IQCCRDP0

IQC6609E	STORAGE TOO SHORT TO READ MEMBER *member_name* INTO MEMORY.

Explanation

The amount of storage that is available to read the member that is indicated by the value of *member_name* is insufficient to be able to read the member into memory.

System action

Processing terminates.

User response

Report the problem to IBM Software Support.

Module

IQCCRDP0

IQC6610E	IQCQCF= PARM NOT SPECIFIED ON EXEC STATEMENT

Explanation

The IQCQCF=*cfg_name* parameter was not specified on the Syntax Checker MVS PARM statement.

System action

Syntax checking terminates.

User response

Add the IQCQCF= parameter to the MVS PARM= statement. Following is a valid PARM= statement:

```
// PARM='BPEINIT=IQCSINI0, IQCTYPE=type,
IQCCFG=cfg_name'
```

In the message text,

type=
    is specified as one of the following values:

    E
        QCF extension PROCLIB configuration definition member

    S
        QCF server PROCLIB configuration definition member

cfg_name=
    specifies the PROCLIB member name of the QCF server or QCF extension configuration definition to be syntax checked.

Module

IQCSIN10

IQC6611E	IQCTYPE= PARM NOT SPECIFIED ON EXEC STATEMENT

Explanation

The IQCTYPE=*type* parameter was not specified on the Syntax Checker MVS PARM= statement.

System action

The syntax checking process terminates.

User response

Specify the 1-character IQCTYPE=*type* parameter on the MVS PARM= statement. A valid MVS PARM= statement is as follows:

```
// PARM='BPEINIT=IQCSINI0, IQCTYPE=type,
IQCCFG=cfg_name'
```

In the message text,

***type*=**
> is specified as one of the following values:
>
> **E** (QCF extension PROCLIB configuration definition member)
> **S** (QCF server PROCLIB configuration definition member)

***cfg_name*=**
> specifies the PROCLIB member name of the QCF server or QCF extension configuration definition to be syntax checked.

Module

IQCSIN10

IQC6612E	TYPE SPECIFIED BY IQCTYPE= IS INVALID

Explanation

The IQCTYPE=*type* has specified an invalid type on the Syntax Checker PARM= statement.

System action

Syntax checking terminates.

User response

Check the PARM= statement to confirm that the type specified for IQCTYPE=*type* is valid. Following is a valid PARM= statement:

```
// PARM='BPEINIT=IQCSINI0, IQCTYPE=type,
IQCCFG=cfg_name'
```

In the message text,

***type*=**
> is specified as one of the following values:
>
> **E** (QCF extension PROCLIB configuration definition member)
> **S** (QCF server PROCLIB configuration definition member)

***cfg_name*=**
> specifies the PROCLIB member name of the QCF server or QCF extension configuration definition to be syntax checked.

Correct the IQCTYPE=*type* value.

Module

IQCSIN10

IQC6613E	THE IQCCFG= AND IQCTYPE= PARMS NOT SPECIFIED ON THE EXEC STATEMENT

Explanation

The IQCQCF=*cfg_name* and IQCTYPE=*type* were not specified on the Syntax Checker MVS PARM= statement.

System action

Syntax checking terminates.

User response

Add the IQCQCF=*cfg_name* and IQCTYPE=*type* parameters to the MVS PARM= statement. Following is a valid PARM= statement:

```
// PARM='BPEINIT=IQCSINI0, IQCTYPE=type,
IQCCFG=cfg_name'
```

In the message text,

***type*=**
> is specified as one of the following values:
>
> **E** (QCF extension PROCLIB configuration definition member)
> **S** (QCF server PROCLIB configuration definition member)

***cfg_name*=**
> specifies the PROCLIB member name of the QCF server or QCF extension configuration definition to be syntax checked.

Correct the IQCTYPE=*type* value.

Module

IQCSIN10

IQC6614E	IQCCFG=MEMBER NAME (*mem_name*) NOT FOUND IN PROCLIB

Explanation

The IQCQCF=*cfg_name* parameter has specified a PROCLIB member name of the QCF server or QCF extension configuration on the Syntax Checker MVS PARM= statement that was not found in the PROCLIB(s) specified for the PROCLIB DD statement.

System action

Syntax checking terminates.

User response

Check the PARM= statement to determine if the member name is correct. Following is a valid PARM= statement:

```
// PARM='BPEINIT=IQCSINI0, IQCTYPE=type,
IQCCFG=cfg_name'
```

In the message text,

type=
    is specified as one of the following values:

        E (QCF extension PROCLIB configuration definition member)
        S (QCF server PROCLIB configuration definition member)

cfg_name=
    specifies the PROCLIB member name of the QCF server or QCF extension configuration definition to be syntax checked.

Correct the IQCQCF=*cfg_name*, if it is invalid, or include the correct PROCLIB library that contains the requested IQCQCF=*cfg_name*.

Module

IQCSIN10

IQC6615E	IMSPLEX= SPECIFIED BLANK NAME FOR MEMBER *member_name*; QCF SCI REGISTRATION CALL WILL FAIL

Explanation

The IMSPLEX statement did not specify an IMSplex name in the QCF extension configuration PROCLIB member being checked by the Syntax Checker or the IMSPLEX statement was missing. The QCF SCI registration call for the QCF server or QCF extension will fail.

In the message text,

member_name
    specifies the PROCLIB member being processed by the Syntax Checker.

System action

Syntax checking continues.

User response

Change the IMSPLEX statement so that it defines the correct IMSPLEX name for the QCF extension configuration definition.

Module

IQCSIN10

IQC6616E	SYNTAX CHECKING HAS FOUND INVALID OPTION OF (*value*) FOR STATEMENT *statement_id* FOR MEMBER *member_name*

Explanation

The Syntax Checker has stopped with invalid option of *value* for statement *statement_id*.

In the message text,

value
    specifies the invalid option that was specified.

statement_id
    specifies the name of the PROCLIB statement with invalid option *value*.

member_name
    specifies the PROCLIB member being processed by the Syntax Checker.

System action

Syntax checking terminates.

User response

Correct the defined *value* for the *statement_id*.

Module

IQCSIN10

IQC6617I	SYNTAX CHECKING HAS COMPLETED SUCCESSFULLY FOR PROCLIB MEMBER=*member_name*

Explanation

The Syntax Checker has ended with no errors for the specified PROCLIB *member_name*.

System action

Syntax checking ends.

User response

No action is required.

Module

IQCSIN10

IQC6618E	DUPLICATE *type* ID OF (*name*) HAS BEEN FOUND FOR MEMBER *member_name*

Explanation

The AREA or FAILSAFE statement ID has a duplicate of the ID.

In the message text,

type
  specifies the AREA or FAILSAFE statement that is in error.

name
  specifies the duplicate ID.

member_name
  specifies the PROCLIB member being processed by the Syntax Checker.

System action

Syntax checking continues.

User response

Correct the duplicate AREA ID or FAILSAFE ID, and run the job again.

Module

IQCSIN10

IQC6619E	THE *type_ %* PERCENTAGE FOR *type_id* ID=*area_id_name*, EXCEEDS 99% FOR MEMBER *member_name*

Explanation

The AREA or FAILSAFE statement has a PERCENT=(TOTAL= value....) greater than 99%.

In the message text,

type_%
  specifies the TOTAL or USED keyword that is in error.

type_id
  specifies the AREA ID or FAILSAFE ID statement that is in error.

area_id_name
  specifies the AREA ID or FAILSAFE ID name that is in error.

member_name
  specifies the PROCLIB member being processed by the Syntax Checker.

System action

Syntax checking continues.

User response

Ensure that the PERCENT=(TOTAL=values) in all AREA and FAILSAFE statements do not exceed 99% for the TOTAL= value.

Module

IQCSIN10

IQC6620I	THE *type_ %* PERCENTAGE FOR *type_id* ID=*area_id_name*, SPECIFIES 0% FOR MEMBER *member_name*

Explanation

This is an information message only. The AREA or FAILSAFE statement specifies that PERCENT=(TOTAL=0....).

In the message text,

type_%
  specifies the TOTAL or USED keyword that is in error.

type_id
  specifies the AREA or FAILSAFE statement that is in error.

area_id_name
  specifies the AREA ID or FAILSAFE ID name that is in error.

member_name
  The PROCLIB member being processed by the Syntax Checker.

System action

Syntax checking continues.

User response

Ensure that 0% is not specified for the TOTAL= value in any AREA or FAILSAFE statements. Although this value

is valid, it implies that an action message should be sent for the first message of every transaction and/or input device, which may not be the desired action.

Module

IQCSIN10

IQC6621E   **ERROR PROCESSING THE SYNTAX CHECKER EXEC STATEMENT PARMS**

Explanation

While processing the Syntax Checker EXEC statement PARMS, an error occurred.

System action

Syntax checking terminates.

User response

See previous messages IQC6610E through IQC6614E and IQC6622E through IQC6626E for clarification. Correct the problems, and run the job again.

Module

IQCSIN10

IQC6622E   **DUPLICATE AREA AND FAILSAFE ID OF (*name*) HAS BEEN FOUND FOR MEMBER *member_name***

Explanation

The AREA or FAILSAFE statement has a duplicate ID.

In the message text,

name
　　specifies the duplicate name.

member_name
　　specifies the PROCLIB member being processed by the Syntax Checker.

System action

Syntax checking continues.

User response

Correct the duplicate AREA ID or FAILSAFE ID, and run the job again.

Module

IQCSIN10

IQC6623E   ***type_id* ID OF BLANKS HAS BEEN FOUND FOR MEMBER *member_name***

Explanation

The AREA and/or FAILSAFE statements has an ID of blanks for the named member being processed.

In the message text,

type_id
　　specifies the type of statement (AREA and/or FAILSAFE) that has an ID=. of blanks.

member_name
　　specifies the PROCLIB member being processed by the Syntax Checker.

System action

Syntax checking continues.

User response

Correct the AREA ID and/or FAILSAFE ID, and run the job again.

Module

IQCSIN10

IQC6624E   **THERE ARE *count type_id* STATEMENTS DEFINED FOR MEMBER *member_name***

Explanation

More than 10 AREA or FAILSAFE statements are defined for the named member being processed.

In the message text,

count
　　specifies the total number of AREA or FAILSAFE statements that are defined.

type_id
　　specifies the statement (AREA or FAILSAFE) which is defined more than 10 times for the named member being processed.

member_name
　　specifies the PROCLIB member being processed by the Syntax Checker.

System action

Syntax checking continues.

User response

Remove the unnecessary AREA or FAILSAFE statement, and run the job again.

Module

IQCSIN10

IQC6625E   **THE** *type_id* **STATEMENT ID NAME (***name***) IS INVALID FOR MEMBER** *member_name*

Explanation

The specified ID= *name* for the AREA or FAILSAFE statement is invalid for the member being processed.

In the message text,

type_id
 specifies the type of statement (AREA or FAILSAFE) that is in error.

name
 specifies the invalid name.

member_name
 specifies the PROCLIB member being processed by the Syntax Checker.

System action

Syntax checking continues.

User response

Specify a valid ID= name in the AREA or FAILSAFE statement.

A valid name consists of:

- uppercase alphabetic characters (A - Z)
- special characters ($, #, and @)
- numeric values (0 - 9)

The first character of the name cannot begin with a numeric value.

Module

IQCSIN10

IQC6626E   **THE** *proc_name* **PARM (***name***) FOR THE** *statement_name* **STATEMENT IS INVALID FOR MEMBER** *member_name*

Explanation

The identified statement (*statement_name*) specified an invalid PROCNAME= or JOBNAME= name (*name*) for the member being processed.

In the message text,

proc_name
 specifies the parameter (PROCNAME= or JOBNAME=) that is in error.

name
 defines the name passed for PROCNAME= or JOBNAME= on the specified *statement_name*.

statement_name
 specifies the control statement (ONTHRESHOLDAB, ONTHERSHOLDBC, ONTHRESHOLDCD, ONTHRSHOLDUT, ONWARMSTART, or ONCOLDSTART) that is in error.

member_name
 specifies the name of the PROCLIB member being processed by the Syntax Checker.

System action

Syntax checking continues.

User response

Specify a valid 1- to 8-character PROCNAME= or JOBNAME= name in the AREA or FAILSAFE statement.

A valid name consists of

- uppercase alphabetic characters (A - Z)
- special characters ($, #, and @)
- numeric values (0 - 9)

The first character of the name cannot be a numeric value.

Module

IQCSIN10

IQC6628E   **NUMERIC VALUE OUTSIDE OF ALLOWABLE RANGE FOR MAXPOOL**

Explanation

The number that was specified in the MAXPOOL parameter is outside of the range for that value.

System action

Processing terminates.

User response

Correct the MAXPOOL=value and run the job.

Module

IQCSIN10, IQCCIN00

IQC6701E   **LOCATE OF THE QCF EXTENSION CQAB BLOCK FAILED.**

Explanation

An attempt to locate the QCF extensions CQAB block was unsuccessful.

System action

Processing terminates.

User response

Report the problem to IBM Software Support.

Module

IQCCCVC0, IQCCSCH8, IQCCSCH9, IQCCSCHA, IQCCIN18, IQCCIN19, IQCCIN1A

IQC6702E   **QCF EXTENSION HAS FAILED TO ESTABLISH AN ESTAE ROUTINE.**

Explanation

QCF extensions attempt to establish an ESTAE routine has failed.

System action

Processing terminates.

User response

Report the problem to IBM Software Support.

Module

IQCCCVC0, IQCCSCH8, IQCCSCH9, IQCCSCHA

IQC6709E   **QCF EXTENSION UNABLE TO NOTIFY CLIENT WITH MEMBER NAME OF *client_name* WITH *var_text*.**

Explanation

QCF extensions was unable to notify the client with the member name that is indicated by the value of *client_name* , with one of the following error conditions that is indicated by the value of *var_text*:

- BAD AWE FUNCTION CODE ERROR MSG
- QMGR ERROR MSG, RC=*retc*
- SCI SEND ERROR MSG, RC=*retc*, RS=*rsnc*
- SCI SEND RESPONSE MSG, RC=*retc*, RS=*rsnc*

Return codes and reason codes are indicated by the values of *retc* and *rsnc* respectively. For explanation of these return and reason codes, see *IMS System Programming APIs*.

System action

Processing terminates.

User response

Correct the problem that is indicated by the value of *var_text* and run the job again.

Module

IQCCCVC0

IQC6800I   **QSUN ACTION NOTIFICATION: IMSID = *xxxx* AREA = *areaname* QUSAGE = *%use* SOURCE = *srcname* ACTION TAKEN = JTSTP**

Explanation

This is the QSUN action notification message that shows the IMSID, area, queue use percentage, source, and action taken, which is JTSTP.

System action

The source of the message is stopped by sending responses that depend on the source type.

User response

None. This message is for informational purposes only.

Module

IQCCQAT0

IQC6801I   **QSUN ACTION NOTIFICATION: IMSID = *xxxx* AREA = *areaname* QUSAGE = *%use* REGION = *deptregn* ACTION TAKEN = [WTO | WAIT | ABEND | STOP]**

Explanation

This is the QSUN action notification message that shows the IMSID, area, queue use percentage,

dependent region, and action taken as indicated in the message text.

System action

Processing continues.

User response

None. This message is informational.

Module

IQCCQAT0

| IQC6802I | QSUN ACTION NOTIFICATION: IMSID = *xxxx* AREA = *areaname* QUSAGE = *%use* LTERM = *lterm_nm* ACTION TAKEN = [WTO \| STOP \| WAIT] |

Explanation

This is the QSUN action notification message that shows the IMSID, area, queue use percentage, LTERM, and action taken as indicated in the message text.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCCQAT0

| IQC6803I | QSUN ACTION NOTIFICATION: IMSID = *xxxx* AREA = *areaname* QUSAGE = *%use* TRANSACTION = *trnsactn* ACTION TAKEN = [WTO \| STOP \| WAIT] |

Explanation

This is the QSUN action notification message that shows the IMSID, area, queue use percentage, transaction, and action taken as indicated in the message text.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCCQAT0

| IQC6804I | QSUN SOURCE ANALYSIS (CSTOP/ CSTART): IMSID = *xxxx* AREA = *areaname* REGION = *deptregn* TRANSACTION = *trnsactn* |

Explanation

This is the QSUN source analysis message for CSTOP or CSTART that shows the IMSID, area, dependent region, and transaction as indicated in the message text.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCCQAT0

| IQC6805I | QSUN SOURCE ANALYSIS (CSTOP/ CSTART): IMSID = *xxxx* AREA = *areaname* REGION = *deptregn* LTERM = *lterm_nm* |

Explanation

This is the QSUN source analysis message for CSTOP or CSTART that shows the IMSID, area, dependent region, and LTERM as indicated in the message text.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCCQAT0

| IQC6806I | QSUN DESTINATION ANALYSIS (CSTOP/CSTART): IMSID = *xxxx* AREA = *areaname* LTERM = *lterm_nm* STATUS=*statusxx* |

Explanation

This is the QSUN source analysis message for CSTOP or CSTART that shows the IMSID, area, LTERM, and STATUS as indicated in the message text.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCCQAT0

IQC6807I	**QSUN DESTINATION ANALYSIS (CSTOP/CSTART): IMSID = *xxxx* AREA = *areaname* TRANSACTION = *trnsactn* STATUS = *statusxx***

Explanation

This is the QSUN destination analysis message for CSTOP or CSTART that shows the IMSID, area, transaction, and status as indicated in the message text.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCCQAT0

IQC6808I	**QSUN SOURCE ANALYSIS (OSTOP/ OSTART): IMSID = *xxxx* AREA = *areaname* REGION = *deptregn***

Explanation

This is the QSUN source analysis message for OSTOP or OSTART that shows the IMSID, area, and REGION as indicated in the message text.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCCQAT0

IQC6809I	**QSUN SOURCE ANALYSIS (OSTOP/ OSTART): IMSID = *xxxx* AREA = *areaname* LTERM = *lterm_nm***

Explanation

This is the QSUN source analysis message for OSTOP or OSTART that shows the IMSID, area, and LTERM as indicated in the message text.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCCQAT0

IQC6810I	**QSUN DESTINATION ANALYSIS (OSTOP/OSTART): IMSID = *xxxx* AREA = *areaname* LTERM = *lterm_nm* STATUS = *statusxx***

Explanation

This is the QSUN destination analysis message for OSTOP or OSTART that shows the IMSID, area, LTERM, and status as indicated in the message text.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCCQAT0

IQC6811I	**QSUN DESTINATION ANALYSIS (OSTOP/OSTART): IMSID = *xxxx* AREA = *areaname* TRANSACTION = *trnsactn* STATUS = *statusxx***

Explanation

This is the QSUN destination analysis message for OSTOP or OSTART that shows the IMSID, area, transaction, and status as indicated in the message text.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCCQAT0

IQC6821I   **QSUN ACTION ENTRIES HAVE USED UP *mmm* POOLS OUT OF THE MAX *nnn* POOLS**

Explanation

The *mmm* value is the number of used up cell pools.

The *nnn* value is the total number of cell pools.

System action

Processing continues.

User response

None. This message is informational.

Module

IQCCQAT0

IQC6822W   **QSUN ACTION ENTRIES HAVE USED UP ALL *nnn* POOLS**

Explanation

The *nnn* value is the number of used up cell pools.

This message indicates that subsequent messages from all sources might be inserted to the message queue without QSUN processing.

System action

Processing continues.

User response

To activate QSUN processing again, recycle IMS.

Module

IQCCQAT0

IQC6900I   **QCF IMS EXTENSION WAITING FOR SCI: (PLEX NAME: *iplex_name*)**

Explanation

QCF is waiting for the structured call interface (SCI) to initialize for the IMSplex name that is indicated by the value of *iplex_name*.

System action

Processing continues. Only the AREA and FAILSAFE statements are supported until the SCI is connected to by both the IMS Queue Control Facility Extensions and the IMS Queue Control Facility server.

If the SCI is successfully started within 30 minutes of receiving this message, it might take up to six seconds to fully activate all of the IMS Queue Control Facility functions.

If the SCI is started after 30 minutes of receiving this message, it might take up to ten minutes to fully activate all of the IMS Queue Control Facility functions.

User response

You must start the SCI if more than the AREA and FAILSAFE statements are to be functional.

Module

IQCCIN10

IQC6901E   **SCI *function* CALL HAS FAILED.**

Explanation

The function call type that is indicated by the value of *function* that SCI submitted has failed. The value of *function* can be one of the following:

- READY
- QUERY
- RELEASE
- REGISTRATION
- DE-REGISTRATION

System action

Processing terminates.

User response

For explanation of these SCI call function failures, see *IMS System Programming APIs*.

Module

IQCCIN10, IQCCSCH0, IQCCCVC0

IQC6902E   **SCI REGISTRATION CALL NOT MADE. IMSPLEX NAME NOT PROVIDED**

Explanation

The SCI registration call was not made because you did not provide the IMSplex name.

System action

Processing terminates.

User response

Provide the IMSplex name and run the job again.

Module

IQCCIN10

IQC6903I   **QCF EXTENSION SEND OF PROCNAME,** *proc_name* **TO QCF SERVER [WAS SUCCESSFUL | HAS FAILED].**

Explanation

IQC6101W is also issued, but it is not part of the IQCC1ENU definition. The send of the PROCNAME was successful or has failed.

System action

Processing continues.

User response

None, if the send of the PROCNAME was successful. Otherwise, look at the QCFPRINT data set to determine the error, fix the error and run the PROCNAME send again.

Module

IQCCIN10

IQC6904I   **QCF EXTENSIONS STARTING USER SPECIFIED JOBNAME,** *jobname***.**

Explanation

QCF extensions is starting the job name that is indicated by the value of *jobname*.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCCIN10

IQC6905I   **QCF EXTENSION FAILED ISSUING MVS START COMMAND (S JOBNAME) FOR USER SPECIFIED JOBNAME** *jobname***.**

Explanation

An MVS START command, that was issued by QCF extensions, failed for the job name that you specified. The job name that you specified is indicated by the value of *jobname*.

System action

Processing continues.

User response

Look at the job log to determine what error occurred.

Module

IQCCIN10

IQC6906I   **PROCNAME** *procname***, WAS NOT SENT TO A QCF SERVER BECAUSE THERE WERE NO ACTIVE QCF SERVERS.**

Explanation

The PROC name that is indicated by the value of *procname* was not sent to a QCF server, because no QCF servers were active.

System action

Processing continues.

User response

Either activate a QCF server or get one activated. And when a QCF server is active, if you want the PROC name sent, run the job again.

Module

IQCQMRE0

IQC6913I   **QCF EXTENSION STARTING USER SPECIFIED SCIPROC,** *proc_name***)**

Explanation

IMS Queue Control Facility has issued the start command for the user-defined *proc_name*.

System action

Processing continues.

User response

If the SCI is started successfully, no action is required. However, if the SCI address space fails, you must determine and correct the problem, and issue the MVS START command for the PROCLIB to start the SCI. You must start the SCI to ensure that all statements are functional.

Module

IQCCIN10

IQC6914I	SCI *proc_name* FAILED TO BE STARTED

Explanation

IMS Queue Control Facility issued the start command for the user-defined *proc_name*. However, that PROCLIB did not start the SCI address space.

System action

Processing continues. Only the AREA and FAILSAFE statements are supported until the SCI is connected to by both the IMS Queue Control extensions and the IMS Queue Control server.

If the SCI is successfully started within 30 minutes of receiving this message, it might take up to six seconds to fully activate all of the IMS Queue Control functions.

If the SCI is started after 30 minutes of receiving this message, it might take up to ten minutes to fully activate all of the IMS Queue Control functions.

User response

You must start the SCI if more than the AREA and FAILSAFE statements are to be functional.

Module

IQCCIN10

IQC6915I	NO SCI PROC SPECIFIED

Explanation

IMS Queue Control Facility was unable to connect to the SCI address space because the SCI address space was not started and no SCI PROCLIB name was provided in the IMS Queue Control Facility extension defined in the PROCLIB.

System action

Processing continues. Only the AREA and FAILSAFE statements are supported until the SCI is connected to by both the IMS Queue Control extensions and the IMS Queue Control server.

If the SCI is successfully started within 30 minutes of receiving this message, it might take up to six seconds to fully activate all of the IMS Queue Control functions.

If the SCI is started after 30 minutes of receiving this message, it might take up to ten minutes to fully activate all of the IMS Queue Control functions.

User response

You must start the SCI if more than the AREA and FAILSAFE statements are to be functional.

Module

IQCCIN10

IQC7001I	QCF SERVER READY

Explanation

This is an informational message indicating that the QCF server has initialized successfully and is ready.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCZMST0

IQC7002E	QCF INITIALIZATION ERROR IN MODULE *module details*

Explanation

This general initialization error indicates that an early QCF initialization module was unable to perform a function that tried to start the QCF address space.

This is a variable text message. The variable text that is displayed depends on the error that was encountered.

module
> The name of the module that detected the error.

details
> The details that are related to the specific error, where the first parameter is usually the resource that failed initialization, such as module name, thread type, or some other resource. The resource that failed initialization must end in a zero. The next parameter is usually the return code from the failing function. And the last parameter is additional descriptive text, if it is needed.

The variable text can be one of the following errors:

- Unknown error
- Unable to create (BPEBPCRE) buffer pool
- Module load (BPELOAD) errors
- Module load and call (BPELOADC) errors
- Unable to BPEBPGET needed storage
- Error reading a PROCLIB data set
- Error parsing a configuration data set

System action

Processing continues.

User response

If this error persists, contact IBM Software Support.

Module

IQCZIN10, IQCZPAR0

IQC7003E	CRITICAL ERROR IN MODULE *module reason* RC= *rc* RSN=*nnnnnnnn*

Explanation

A critical error occurred in the module that is indicated by the value of *module,* for the reason that is indicated by the value of *reason,* with the return code that is indicated by the value of *rc,* for the reason that is indicated by the value of *nnnnnnnn.*

System action

Processing terminates.

User response

Report the problem to IBM Software Support.

Module

All

IQC7004I	*tcb* TCB INITIALIZATION COMPLETE

Explanation

This is an informational message that indicates that the TCB that is indicated by the value of *tcb* has successfully been initialized.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCZITCB

IQC7005I	*tcb* TCB TERMINATION COMPLETE

Explanation

This is an informational message that indicates that the TCB that is indicated by the value of *tcb* has terminated successfully.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCZITCB

IQC7006E	SCI *function* FAILED IN MODULE *module* RC=*return*, RSN=*reason*

Explanation

This is an informational message indicating that the SCI function that is indicated by the value of *function* has failed in the module that is indicated by the value of *module* with the return code that is indicated by the value of *return* for the reason that is indicated by the value of *reason*.

System action

Depending on the requested function and the return and reason codes, processing continues or processing stops and the server or batch job terminates.

User response

Check the return and reason codes for the SCI services and take appropriate action, if any. Return and reason codes can be found in *IMS System Programming APIs*.

Module

IQCBMST0, IQCBTQCF, ICZCSV0, IQCZREQ0, IQCZSAF0, IQCZCVC0

IQC7007E   **ERROR STARTING QCF ADDRESS SPACE, CHECK FOR DUPLICATE JOBNAME** *jobname*

Explanation

An error was encountered while starting a QCF address space for the job name that is indicated by the value of *jobname*. It is likely that a QCF address space already exists for the IMS control region, where only one QCF address space can be started.

System action

Processing terminates.

User response

Check for a duplicate job name that matches the value of *jobname*, change your job name and run the job again.

Module

IQCZIN10

IQC7008I   **ALLOCATE REJECTED FOR USERID=**<i>userid</i> **INSUFFICIENT AUTHORITY TO RESOURCE** *resource* **RACROUTE AUTH R15=**<i>racroute register 15</i> **RC=**<i>return_code</i> **RSN=**<i>reason_code</i>

Explanation

This is an informational message indicating that the allocate to data set was rejected for the userid that is indicated by the value of *userid*, because there was insufficient authority for the resource that is indicated by the value of *resource*. The RACROUTE R15 value is indicated by the value of *racroute register 15*, the return code is indicated by the value of *return_code*

and the reason code is indicated by the value of *reason_code*.

System action

Processing continues.

User response

Check the RACF profile. For explanations of the RACF return and reason codes, see *z/OS Security Server RACF Messages and Codes*.

Module

IQCZSAF0

IQC7009E   **DYN [ALLOC | UNALLOC] FAILED FOR** *dstype*, **RC=**<i>rc</i> **RSN=**<i>rsn</i> **DSN=**<i>dsn</i>

Explanation

Dynamic data set allocation or unallocation failed. In the message text,

dstype
    An 8-byte string, which is either a description of the data set that failed allocation or unallocation, or the DD name if no description was provided

rc
    Return code from the MVS dynamic allocation

rsn
    Reason code from the MVS dynamic allocation

dsn
    Data set name

System action

Processing continues.

User response

Check the data set that is indicated by the value of *dsn*. For explanations of the MVS return and reason codes, see *z/OS MVS Authorized Assembler Services Reference, Volume 1 (ALESERV-DYNALLOC)*.

Module

IQCZDYN0

IQC7010A   **QCF** *type* **WAITING FOR** *target*

Explanation

The QCF address space that is running is waiting for a target address space to become available so that a request can be issued.

In the message text,

type
Type of QCF member that is waiting

target
Type of address space for which the QCF member is waiting

System action

Processing continues.

User response

Check the target address space and start it, if necessary.

Module

IQCZCSV0

IQC7012I	QCF SERVER OPTIONS / STATISTICS

Explanation

The IQC subsystem has already been installed.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCZCMD2

IQC7014I	*vartextvartext*

Explanation

This is an informational message indicating the QCF server options or statistics. This message is displayed just after IQC7012I.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCZCMD2

IQC7015I	SERVER RECONNECTED TO SCI

Explanation

This is an informational message issued by the QCF server indicating that SCI is started.

System action

Processing continues

User response

None. This message is informational.

Module

IQCZXNF0

IQC7016I	SERVER DISCONNECTED FROM SCI

Explanation

This is an informational message issued by the QCF server indicating that the SCI is ended.

System action

Server processing is discontinued until the SCI is available. After the SCI is restarted, the server will register automatically and continue processing. Message IQC7015I is issued when the server has again registered to the SCI.

User response

None. This message is informational.

Module

IQCZNF0

IQC7017E	Processing has stopped due to *xxxxxxxx*

Explanation

Processing has stopped due to *xxxxxxxx*.

System action

Processing terminates.

User response

Report the problem to IBM Software Support.

Module

IQCBTQCF

IQC7018I	ACTIONS TO SOME DESTINATIONS REJECTED DUE TO INSUFFICIENT AUTHORITY USERID=userid

Explanation

This message is issued by the QCF batch jobs and indicates that some destinations are not accessible for the IMS ID and user ID.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCXCTRL

IQC7020E	NO MESSAGES PASSED TO IMS HAVE BEEN COMMITTED

Explanation

This message is issued when a Syncpoint Error is encountered during the LOAD step of a RECOVERDM job, or during a LOAD job without the use of the PACING statement. This message indicates that no messages have been committed to the IMS queue.

System action

Processing terminates.

User response

Restart IMS and retry the RECOVERDM or LOAD job.

Module

IQCBTQCF

IQC7043E	ERROR DURING CREATE UNLOAD GDG DATASET IN MODULE *module vartext*

Explanation

An error occurred during the processing of a create unload GDG data set.

The possible meanings of message variables are described in the following list:

module
> The name of the module that issued the message.

vartext

> **UNIT() SPECIFIED WITHOUT VOLSER() ON ULOADGDG PARAMETER**
> > UNIT() was specified without a VOLSER() on the EXTTRACE parameter.

> **VOLSER() SPECIFIED WITHOUT UNIT() ON ULOADGDG PARAMETER**
> > VOLSER() was specified without a UNIT() on the EXTTRACE parameter.

> **STORCLAS() AND UNIT/VOL SPECIFIED ON ULOADGDG PARAMETER**
> > STORCLAS() and UNIT or VOL was specified on the EXTTRACE parameter.

> **SPACEUNIT() AND AVGREC() SPECIFIED ON ULOADGDG PARAMETER**
> > SPACEUNIT() and AVGREC() was specified on the EXTTRACE parameter.

> **EXTERNAL PROCESSING ERROR, *ssssssss* RC=*nnnn/nnnnnnnn***
> > An external processing error occurred. *ssssssss* is the name of the z/OS service that was invoked by IMS Queue Control Facility. RC=*nnnn/nnnnnnnn* indicates the error code (return and reason codes) returned from that service.

System action

Processing terminates.

User response

Supply a valid statement and try the operation again.

Module

IQCZDYN1

IQC8001E	*variable text*

Explanation

This message indicates that an error occurred while the QCF server or batch job was attempting to access certain data sets.

The possible meanings of message variables are described in the following list:

vartext

1

UNABLE TO OPEN DATA SET, DDNAME=*ddname*

2

ERROR OCCURRED WHEN CLOSING DATA SET, DDNAME=*ddname*

3

ERROR OCCURRED WHEN READING DATA SET, DDNAME=*ddname*

4

ERROR OCCURRED WHEN WRITING DATA SET, DDNAME=*ddname*

5

SPACE PARAMETER PRIMARY ALLOCATION WAS SPECIFIED ZERO, DATA SET NOT OPENED. DDNAME=*ddname*

6

CHECK MVS MESSAGE IECXXXX IN JOB LOG FOR DETAILED INFORMATION.

where *ddname* is the DD name specified in DCB.

System action

Processing ends.

User response

Check MVS message IECXXXX for more detailed information, and check the data set indicated by *ddname*.

Module

IQCLOAD0, IQCLOD61, IQCQRY61, IQCQUERY, IQCUNLOD, IQCXCTRL, IQCXSEL6, ICXTIOT

IQC9001I	UNKNOWN PARAMETER SPECIFIED, USING DEFAULT - RC(11)

Explanation

The parameter on the ROUTECODE value was not recognized.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCPARMS

IQC9002I	ROUTE CODE TOO HIGH (> 16), USING DEFAULT - RC(11)

Explanation

The parameter on the ROUTECODE value was greater than 16. Route codes higher than 16 are not supported.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCPARMS

IQC9003I	UNKNOWN ROUTE CODE SPECIFIED, USING DEFAULT - RC(11)

Explanation

The parameter on the ROUTECODE value was not recognized.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCPARMS

IQC9004I	INVALID ROUTE CODE RANGE, USING DEFAULT - RC(11)

Explanation

The range of route codes that was specified on the ROUTECODE value was invalid.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCPARMS

IQC9005I   **ROUTE CODE 0 NOT VALID, USING DEFAULT - RC(11)**

Explanation

ROUTECODE(0) was specified. Route code 0 is not supported. The default ROUTECODE(11) is used instead.

System action

Processing continues.

User response

None. This message is for informational purposes only.

Module

IQCPARMS

IQCA001E   **Invalid first parameter for EXEC PGM=IQCABAT0**

Explanation

The first parameter should be a valid IMS ID.

System action

Processing terminates.

User response

Correct the first EXEC parameter and run the job again.

Module

IQCABAT0

IQCA002E   **Invalid second parameter for EXEC PGM=IQCABAT0**

Explanation

The second parameter should be one of the following 2-character valid function types:

AB
     RECOVERAB
DM
     RECOVERDM
PR
     REPROCESS

System action

Processing terminates.

User response

Correct the parameter and run the job again.

Module

IQCABAT0

IQCA003E   **Third parameter not D=YYYY.DDD,T=HH:MM:SS**

Explanation

The date and time format of the third parameter is not valid.

System action

Processing terminates.

User response

Correct the parameter and run the job again.

Module

IQCABAT0

IQCA004E   **Input date is in the future**

Explanation

The D=*YYYY.DDD* parameter indicates a future date; it must be in the past.

System action

Processing terminates.

User response

Correct the parameter and run the job again.

Module

IQCABAT0

IQCA005E   **RE specified without D=...., T=....**

Explanation

The RE function has been specified without D=.....,
T=..... Reprocess requires date and time to select a
checkpoint.

System action

Processing terminates.

User response

Add the missing parameter, and run the job again.

Module

IQCABAT0

IQCA006E   **QCFJCLOT DD statement missing**

Explanation

The QCFJCLOT DD statement is missing. This DD
statement points to the library in which the new
RECOVERAB / RECOVERDM / REPROCESS JCL is
stored.

System action

Processing terminates.

User response

Correct the JCL and run the job again.

Module

IQCABAT0

IQCA007E   **QCFJCLIN DD statement missing**

Explanation

The QCFJCLIN DD statement is missing. This DD
statement points to the library in which the skeleton
RECOVERAB / RECOVERDM / REPROCESS JCL is
stored.

System action

Processing terminates.

User response

Correct the JCL and run the job again.

Module

IQCABAT0

IQCA008E   **Member "ims_id"|##|"recovery
                      type" not in QCFJCLIN**

Explanation

The skeleton JCL member that is indicated is missing
in QCFJCLIN library.

System action

Processing terminates.

User response

Ensure that the skeleton JCL is in the QCFJCLIN library
and run the job again.

Module

IQCABAT0

IQCA009E   **Open failed for SYSIN data set**

Explanation

SYSPRINT is the output file from the LIST.LOG
command. If the DD for SYSPRINT is not provided, the
file is dynamically allocated.

System action

Processing terminates.

User response

Check the allocation, correct the JCL, and run the job
again.

Module

IQCABAT0

IQCA010E   **Open failed for SYSPRINT data set**

Explanation

SYSPRINT is the output file from the LIST.LOG
command. If the DD for SYSPRINT is not provided, the
file is dynamically allocated.

System action

Processing terminates.

User response

Check the allocation, correct the JCL, and run the job again.

Module

IQCABAT0

IQCA011E   **Open failed for QCFJCLOT data set**

Explanation

Open failed for library with new RECOVERAB / RECOVERDM JCL.

System action

Processing terminates.

User response

Correct the JCL and run the job again.

Module

IQCABAT0

IQCA012E   **Open failed for "ims_id"| ##|"recovery type"**

Explanation

The skeleton member for the RECOVERAB / RECOVERDM JCL in the QCFJCLIN data set failed to open.

System action

Processing terminates.

User response

Ensure that the skeleton JCL member is in the QCFJCLIN library, that it is uncorrupted, and run the job again.

Module

IQCABAT0

IQCA013E   **Invalid CHECKPOINT line in LIST.LOG output**

Explanation

An invalid line was found in the LIST.LOG output.

System action

Processing terminates.

User response

Check the SYSPRINT file for invalid lines.

Module

IQCABAT0

IQCA014E   **Open failed for QCFJCLIN data set**

Explanation

The QCFJCLIN data set failed to open.

System action

Processing terminates.

User response

Correct the problem and run the job again.

Module

IQCABAT0

IQCA015E   **Build of RECOVERDM job failed-steps / logs missing**

Explanation

The skeleton JCL does not contain all of the necessary steps to build the new RECOVERDM JCL.

System action

Processing terminates.

User response

Correct the skeleton JCL and run the job again.

Module

IQCABAT0

IQCA016E   **Build of RECOVERAB job failed-steps / logs missing**

Explanation

The skeleton JCL does not contain all of the required steps to build the new RECOVERAB JCL.

System action

Processing terminates.

User response

Correct the skeleton JCL and run the job again.

Module

IQCABAT0

IQCA017E	Non-zero return code returned from DSPURX00/DSPURXRT

Explanation

The Attach for DSPURX00/DSPURXRT failed.

System action

Processing terminates.

User response

Ensure that the STEPLIB points to IMS.RESLIB. Correct the JCL, and run the job again.

Module

IQCABAT0

IQCA018E	ATTACH failed for DSPURX00/DSPURXRT

Explanation

The STEPLIB concatenation does not contain all of the necessary libraries, or the region is too small.

System action

Processing terminates.

User response

Check the STEPLIB DD concatenation or increase the region size and run the job again.

Module

IQCABAT0

IQCA019E	WRITE/STOW error. Reallocate QCFJCLOT.

Explanation

An error occurred while saving the newly created JCL.

System action

Processing terminates.

User response

Reallocate the QCFJCLOT library and run the job again.

Module

IQCABAT0

IQCA020E	Out of space D37. Compress QCFJCLOT.

Explanation

Space for the QCFJCLOT library is full.

System action

Processing terminates.

User response

Reallocate or compress the QCFJCLOT library and run the job again.

Module

IQCABAT0

IQCA021E	Out of space E37. Reallocate QCFJCLOT.

Explanation

Space for the QCFJCLOT library is full.

System action

Processing terminates.

User response

Reallocate or compress the QCFJCLOT library and run the job again.

Module

IQCABAT0

IQCA022E	Directory full. Reallocate QCFJCLOT.

Explanation

The directory in which the QCFJCLOT library resides, is out of space.

System action

Processing terminates.

User response

Reallocate the QCFJCLOT library and increase the directory blocks.

Module

IQCABAT0

IQCA023E	**Creation of recovery procedure failed.**

Explanation

An error occurred while building the new JCL. The process cannot continue. This is an informational message.

System action

Processing terminates.

User response

Review all of the error messages that are issued prior to this message (IQCA023E) being issued. Locate the reason for the termination, correct the error, and try the operation again.

Module

IQCABAT0

IQCA024E	**Allocation failed for RECON1 via IMSDALIB / STEPLIB**

Explanation

Information is missing while trying to allocate RECON1.

System action

Processing terminates.

User response

Ensure that all of the appropriate information is provided in the JCL to allocate RECON1, and run the job again.

Module

IQCABAT0

IQCA025E	**Allocation failed for RECON2 via IMSDALIB / STEPLIB**

Explanation

Information is missing to allocate RECON2.

System action

Processing terminates.

User response

Ensure that all of the appropriate information is provided in the JCL to allocate RECON2, and run the job again.

Module

IQCABAT0

IQCA026E	**Open failed for IMSDALIB / STEPLIB**

Explanation

The open operation failed for the IMSDALIB / STEPLIB.

System action

Processing terminates.

User response

Locate the reason for the failure, correct the error, and run the job again.

Module

IQCABAT0

IQCA027E	**Member RECON1 not found in IMSDALIB / STEPLIB**

Explanation

The IMSDALIB / STEPLIB concatenations are missing the RECON1 member.

System action

Processing terminates.

User response

Check the IMSDALIB / STEPLIB concatenations to ensure that member, RECON1 is present and uncorrupted.

Module

IQCABAT0

IQCA028E	Member RECON2 not found in IMSDALIB / STEPLIB

Explanation

The IMSDALIB / STEPLIB concatenations are missing the RECON2 member.

System action

Processing terminates.

User response

Check the IMSDALIB / STEPLIB concatenations to ensure that member, RECON2 is present and uncorrupted.

Module

IQCABAT0

IQCA029E	RECOVERDM requested - no DUMPQ or PURGE found

Explanation

The LIST.LOG output does not contain a valid checkpoint.

System action

Processing terminates.

User response

Restart the job using an earlier date in EXEC parameter 3.

Module

IQCABAT0

IQCA030E	RECOVERAB requested - no DUMPQ, PURGE OR SNAPQ found

Explanation

The LIST.LOG output does not contain a valid checkpoint.

System action

Processing terminates.

User response

Restart the job using an earlier date in EXEC parameter 3.

Module

IQCABAT0

IQCA031E	Dynamic allocation for selected PRISLD/PRIOLD failed

Explanation

The allocation of the data sets failed for the selected PRISLD/PRIOLD.

System action

Processing terminates.

User response

Ensure that the selected logs are available and try the operation again.

Module

IQCABAT0

IQCA032E	Open failed for selected PRISLD/ PRIOLD

Explanation

The open operation failed for the selected PRISLD/ PRIOLD set.

System action

Processing terminates.

User response

Locate the reason for the failure in opening the PRISLD, correct the error, and run the job again.

Module

IQCABAT0

IQCA033E	End of SIMPLE checkpoint found - no related start

Explanation

An end of a SIMPLE checkpoint was encountered for which there was no starting checkpoint.

System action

Processing terminates.

User response

Review the LIST.LOG output to locate the reason for the error and run the job again.

Module

IQCABAT0

IQCA034E	End of DUMPQ checkpoint found - no related start

Explanation

An end of a DUMPQ checkpoint was encountered for which there was no starting checkpoint.

System action

Processing terminates.

User response

Review the LIST.LOG output to locate the reason for the error.

Module

IQCABAT0

IQCA035E	Error in number of volumes for log data set

Explanation

An error occurred in the number of volumes for the log data set.

System action

Processing terminates.

User response

Review the LIST.LOG output to locate the reason for the error, correct the error, and run the job again.

Module

IQCABAT0

IQCA036E	REPROCESS requested - no DUMPQ, PURGE, OR SNAPQ found

Explanation

The LIST.LOG output does not contain a valid checkpoint.

System action

Processing terminates.

User response

Restart the job using an earlier date in EXEC parameter 3.

Module

IQCABAT0

IQCA037E	Build of REPROCESS job failed- steps / logs missing

Explanation

The skeleton JCL does not contain all of the steps that are required to build the new REPROCESS JCL.

System action

Processing terminates.

User response

Correct the skeleton JCL and run the job again.

Module

IQCABAT0

IQCA038E	No PRISLD LOGS selected from RECON listing

Explanation

The LIST.LOG output does not contain PLISLD LOGS.

System action

Processing terminates.

User response

Restart the job using an earlier date in EXEC parameter 3.

Module

IQCABAT0

IQCA039E	No checkpoint found to use with requested function

Explanation

The LIST.LOG output does not contain a valid checkpoint.

System action

Processing terminates.

User response

Restart the job using an earlier date in EXEC parameter 3.

Module

IQCABAT0

IQCA040I	Compressed data set, no message recovery

Explanation

The compressed data set that was found will not be used to recover messages.

System action

Processing continues.

User response

Locate the set of logs that contain the required checkpoints.

Module

IQCABAT0

IQCA041E	RE specified with invalid date/time type (CP/MSG)

Explanation

The fourth parameter should be one of the following valid date/time types:

- DT=CP
- DT=MSG

System action

Processing terminates.

User response

Correct the parameter and run the job again.

Module

IQCABAT0

IQCA042W	Start of checkpoint found before end of previous

Explanation

This is only a warning message. Process continues with next checkpoint and disregards information that the previous checkpoint has not ended.

System action

Processing continues.

User response

None.

Module

IQCABAST0, IQCASELO

IQCA043E	PRIOLD in error/unavail and not archived

Explanation

At least one data PRIOLD data set is in error and not archived. The data set(s) cannot be used for rebuild.

System action

Processing terminates.

User response

Archive logs and run the job again.

Module

IQCABAT0, IQCASELO

IQCA048W	No PRISLDs found in RECON listing for the period

Explanation

No PRISLD logs were found for the requested time period.

System action

Processing continues with PRIOLDs.

User response

None.

Module

IQCABAST0, IQCASELO

IQCA049W   **No valid set of archived logs found**

Explanation

No set of logs were found with the checkpoint type required to perform the requested function.

System action

Processing continues.

User response

None.

Module

IQCABAT0, IQCASELO

IQCA050E   **Rebuild process failed**

Explanation

IQCSELCT failed to retrieve messages to rebuild the queues. Detailed errors are in the job log.

System action

Processing terminates.

User response

Check the job log for errors.

Module

IQCABAT0, IQCASELO

IQCA051W   **START of log does not match STOP of previous log**

Explanation

There is a time gap between stop time and start time of 2 consecutive logs.

System action

Processing continues.

User response

None.

Module

IQCABAST0, IQCASELO

IQCA052W   **Last selected log is ACTIVE, but STOP is not 00**

Explanation

Active logs have stop time of 00:00:00....

System action

Processing continues.

User response

None.

Module

IQCABAST0, IQCASELO

Chapter 21. AIB reason codes

This reference section provides detailed information about the AIB reason codes that are issued by IMS Queue Control Facility.

These topics describe the AIBRETRN and AIBREASN codes set by the IMS message requeuer modules IQCQMRQ0 and DFSQMR10, 20, 30, 40, 50, 60, 70, and A0-L0. These codes are recorded in the SCRAPLOG 6701-MRQE records. IMS Queue Control Facility reports when an error is detected while performing an IMS Queue Control Facility function.

Use the AIBREASN codes when diagnosing problems with IMS Queue Control Facility.

Related reading: For more information about how to diagnose IMS Queue Control Facility problems, see the section that discusses diagnosing problems in the Message requeuer in *IMS Diagnosis*. This information also describes how IMS Queue Control Facility communicates with certain functions in the IMS Transaction Manager and System Services.

X'000000F0' is a unique AIB return code assigned to the IMS Queue Control Facility manager message requeuer processor (IQCQMRQ0). It is set in the AIBRETRN field of the AIB by IQCQMRQ0 when an error is detected while performing an IMS Queue Control Facility function.

IQCQMRQ0 also sets the AIBREASN field in the AIB to a code indicating the type of error detected. These codes are passed back to the applicable IMS Queue Control Facility function, which stores the codes in the QCF prefix segment appended to the front of the message record that caused the error. The IMS Queue Control Facility function writes this record to the SCRAPLOG data set. IMS logs a corresponding 6701-MRQE record to the OLDS.

AIB return codes other than X'000000F0' indicate IMS errors that are not specific to message requeuing. To analyze these return codes and their associated reason codes, see the topic "AIB return and reason codes set by IMS" in *IMS Messages and Codes, Volume 4: IMS Component Codes*.

Each AIBREASN code associated with AIB return code X'000000F0' is described in the following list. Locate the unique AIBREASN code and analyze the error as described. Each AIBREASN code falls into one of three categories:

1. The error is a normal condition and AIBREASN is set for informational purposes. The message is discarded according to protocol. There are seven AIBREASN codes in this category:

   1080
   Message destination is an APPC synchronous conversation type.

   1084
   Message is a nonrecoverable type.

   1088
   Message was flagged to be canceled.

   10A4
   Message is an internal IMS message that is not recoverable.

   1100
   Message is an IMSVERIFY command message.

   2014
   Destination is an inquiry LTERM not signed on.

   A070
   AOI command response cannot be deleted.

2. The error is probably caused by an unsupported or changed IMS features, or destination or source resource names.

   An example of an error: a transaction is deleted from IMS SYSGEN and IMS Queue Control Facility subsequently tries to requeue a message destined for the deleted transaction.

The IMS Queue Control Facility processor (IQCQMRQ0) would detect that the destination no longer exists and set an AIBREASN code of 1024 or 1040. The IMS system programmer should analyze these errors (by following the explanation and programmer response guidelines found in *IMS Diagnosis*) and verify whether the resource was deleted or altered.

3. The error is an IMS or IMS Queue Control Facility internal error, and should be reported to IBM Software Support for resolution.

AIB reason codes have the following numbering scheme:

- X'0xxx' MRQ/QCF initialization error (IQCQMR10)
- X'1xxx' LOAD processing error (IQCQMR70)
- X'2xxx' PURG processing error (IQCQMR70)
- X'3xxx' SETPRFX error (IQCQMR80)
- X'4xxx' CPYPRFX error (IQCQMR80)
- X'5xxx' Error while trying to cancel the message (IQCQMR10)
- X'6xxx' MRQ/QCF detected error and requested cleanup (QCF client)
- X'7xxx' XLATPFX error (IQCQMR80)
- X'8xxx' BROWSE processing error (IQCQMR30)
- X'9xxx' QUERY processing error (IQCQMR60)
- X'Axxx' UNLOAD processing error (IQCQMR50)
- X'Bxxx' RECOVER processing error (IQCQMR40)
- X'Cxxx' Select criteria error (IQCQMR20, IQCQMRA0)
- X'Dxxx' Load processing error (IQCQMR70)
- X'Exxx' Queue space notification error (IQCQMRD0)
- X'Fxxx' Environmental errors (IQCQMRC0)

All AIB reason codes associated with AIB return code X'000000F0' are described in this section. If the AIBREASN code is not documented here, this is probably because IMS has added an additional code.

See the IMS DFSMRAEQ macro for the latest codes.

X'0004'	DEFAULT REASON CODE IF NONE SET

Explanation

The AIBREASN code in REG0 = 0 when the ERROR routine was called.

Programmer response

Trace back to the caller of the ERROR routine. This is an IMS internal error.

X'0008'	INVLD FUNC PASSED TO QMRQ0 ENTRY

Explanation

IQCQMRQ0 was called with an invalid function code in REG1.

Programmer response

Internal error. Trace back to the caller of IQCQMRQ0.

X'000C'	SID PASSED IS ZERO

Explanation

The destination system identification (SYSID) or source SYSID of the message being processed is zero.

Programmer response

Locate the destination SYSID (MSGMSOID) or source SYSID (MSGMSIID) in the message. The SYSIDs are extracted from the control block representing the resource (CNT for LTERMs, SMB for transactions) when the message was created. Verify that the resource was not changed across the restart. Except for some internal system messages, SYSID=0 is invalid and should not occur. Possible IMS internal error.

X'0010'	SID PASSED IS TOO HI VALUE

Explanation

The destination system identification (SYSID) or source SYSID of the message being processed is

higher than the maximum SYSID defined on the MSNAME macros at SYSGEN and is subsequently stored in the SCD at SCDSIDN.

Programmer response

Locate the destination SYSID (MSGMSOID) or source (MSGMSIID) in the message. SYSIDs are extracted from the control block representing the resource (CNT for LTERMs, SMB for transactions) when the message was created. Maximum SYSID is determined from maximum SYSID in the MSNAME macros at SYSGEN and subsequently stored in the SCD at SCDSIDN. Verify that the MSNAMEs were not removed from the SYSGEN and the SCDSIDN is correct.

X'0014'	SID PASSED IS UNDEFINED TO SYSTEM

Explanation

The destination system identification (SYSID) or source SYSID of the message being processed is not defined to the system.

Programmer response

Locate the destination SYSID (MSGMSOID) or source SYSID (MSGMSIID) in the message. SYSIDs are extracted from the control block representing the resource (CNT for LTERMs, SMB for transactions) when the message was created. To be valid, the SYSID must be defined in an MSNAME macro at SYSGEN time.

X'0018'	INVALID IMS OR MRQ LEVEL

Explanation

Either the message being inserted is from an unsupported IMS release, or the QCF client version issuing the call is not supported by this release of IMS.

Programmer response

Verify that:

1. The message being requeued is from a supported IMS release
2. The message is being requeued by a supported QCF version

If case "1" on page 401 is not true, the message cannot be requeued under the current IMS release. If case "2" on page 401 is not true, get the latest level of the QCF product and rerun the job. If both are true, contact your IBM representative.

X'001C'	ENTRY - INVALID CMD/GCMD/ ISRT CALL

Explanation

IQCQMRQ0 was called with an invalid CMD, GCMD, or ISRT call, or an invalid sequence of these calls. The first MRQ/QCF call to IMS must be a CMD or ISRT call. CMD calls must pass a valid command work area. ISRT calls must pass a valid MRQ prefix and IMS message.

Programmer response

If the caller is the MRQ or QCF client, then this is either an MRQ, QCF, or IMS error. This problem can also occur if the release of MRQ or QCF is not supported on the IMS release. Verify that this is a valid MRQ/QCF release for this release of IMS.

X'0020'	INVALID MRQ FUNCTION

Explanation

DFSQMR10 did not recognize the QCF function code.

Programmer response

This is an internal error. Trace back to the caller of DFSQMR10. The function code is stored in QMRWFCN2.

X'0024'	INVALID DFSQMR10 FUNCTION

Explanation

The function is not supported by DFSQMR10.

Programmer response

This is an internal error. Track back to the caller of DFSQMR10. The DFSQMR10 function is stored in QMRWFCN2.

X'0028'	INVALID SPANNED COMMAND

Explanation

Invalid spanned command data was received from the QCF client.

Programmer response

Internal error. One of the following is incorrect:

- Command segment first and/or last flags
- Work area spanned flags

The command data is in the I/O area pointed to by either REG6 or QMRWIO in MRQWORK. The first/last flags are at MRZZ2. The MRQ spanned flags are in MRQWORK flag QMRCFLG2 (address of MRQWORK is in REG5).

X'002C'   ENTRY - INVALID BUILD MRQ PREFIX CALL

Explanation

The Build QCF prefix routine in DFSQMRF0 was called to build an MRQ prefix, but the current function either did not have a prefix buffer area or a message from which to build the prefix.

Programmer response

This is an internal IMS error.

X'0030'   ERROR DURING INIT QC FUN CALL

Explanation

DFSQMR10 encountered an error during initialization of the Queue Control (QC) function.

Programmer response

Locate the QC command buffer that QCMRQCMDP points to and verify that it is a valid QC command. If using the QC-LTBL command, locate the QSN table that is being loaded and verify that it has a valid length.

X'1004'   1ST MSG NOT 1ST QUEUE BUFFER

Explanation

A new message is being inserted, and the first queue buffer message flag (MSGFFRST) is not set on.

Programmer response

Locate the message flags in the message prefix. If the message is a first buffer, then MSGFFRST should be set. Verify the original message on the log and verify that the input to MRSELECT was correct. If not, this is an internal IMS error. If so, the message might have been handled incorrectly by either IQCSELCT, IQCCANCL, or by the load function. If a BROWSE, RECOVER, or UNLOAD function is being processed, the message in error was obtained from the message queue.

X'1008'   CAN'T FIND RACF PREFIX SEGMENT

Explanation

The message was created with a RACF prefix, but the RACF prefix segment could not be located.

Programmer response

Locate the message and verify that flag MSGC1RAC is set. If the flag is set, then the RACF prefix segment with code = 88 must be present.

X'100C'   MSC NOT GEN BUT MSC SEG PRESENT

Explanation

The message was created with an MSC prefix, but MSC is not initialized.

Programmer response

Locate the message and verify that the MSC prefix is present and flag MSGC2MSC is set on. If so, MSC was defined at SYSGEN when the message was created but is not in the SYSGEN now. The flag SCDPDMUL is set on by SYSGEN if MSC is included in the SYSGEN. Do a SYSGEN with MSC.

X'1014'   FINDEST ERR FOR SOURCE=MSGIDSTN

Explanation

The local source name in the message at MSGIDSTN could not be found by the FINDEST routine.

Programmer response

Locate the MSGIDSTN name in the message, and verify that it is a valid local LTERM or MSNAME. If the extended terminal option (ETO) was included at SYSGEN and the name is a dynamic LTERM, verify that ETO is enabled. The FINDEST parameter list used to locate the name is at PSTDCA.

X'1018'   MSGIDSTN BLOCK NOT CNT/LNB/QAB

Explanation

The control block returned by FINDEST, representing the source name at MSGIDSTN, is not a CNT (LTERM), LNB (MSNAME), or QAB (LU 6.2 node).

Programmer response

Locate the MSGIDSTN name in the message and verify that it is a valid LTERM, MSNAME, or LU 6.2 node. If it is an LU 6.2 node, then MSGIDSTN begins with FEFFFFFF and the NODE name is in the LU 6.2 prefix. The control block address is in REG1 in the REG14-12 save area, and the block is at QTPDST.

X'101C'   CAN'T FIND MSC SEGMENT MSGSIPEX

Explanation

The message flag indicates the MSC prefix segment is present, but the segment cannot be located.

Programmer response

Locate the message and verify that flag MSGC2MSC is set. If the flag is set, then the MSC prefix segment with a code = 82 must be present. This is an internal IMS error.

X'1020'	FINDEST ERR FOR SOURCE=MSGMSINM

Explanation

The MSC source name in the message at MSGMSINM could not be found by the FINDEST routine.

Programmer response

Locate the MSGMSINM name in the message, and verify that it is a valid local LTERM. If the extended terminal option (ETO) was specified at SYSGEN and the name is a dynamic LTERM, verify that ETO is enabled.

Note: The MSC LTERM name is only verified if the source SYSID in the message at MSGMSIID is local. Verify that the source SYSID was not changed from a remote SYSID to a local (that is, check the MSNAME macros).

X'1024'	FINDEST ERR FOR DEST = MSGODSTN

Explanation

The local destination name in the message at MSGODSTN could not be found by the FINDEST routine.

Programmer response

Locate the MSGODSTN name in the message and verify that it is a valid local LTERM, MSNAME, or a local or remote TRANSACTION CODE. If the extended terminal option (ETO) was included at SYSGEN and the name is a dynamic LTERM, verify that ETO is enabled. The FINDEST parameter list used to locate the name is at PSTDCA.

X'1028'	MSGODSTN BLOCK NOT EXPECTED CNT

Explanation

The control block returned by FINDEST, representing the destination name at MSGODSTN, is not a CNT (LTERM) or MSC LNB (MSNAME).

Programmer response

Locate the MSGODSTN name in the message and verify that it is a valid LTERM or MSNAME. The control block address is in REG1 in the REG14-12 save area, and the control block is at QTPDST.

X'102C'	MSG DEST FLAG NOT EXPECTED LTERM

Explanation

The message destination control block is a CNT type (either an LTERM or MSC MSNAME). However, the destination type flag in the message is not a CNT type.

Programmer response

Locate the message destination type flag (MSGDFLG2) of the message; it should be a CNT type (hex 82 = CNT type, hex 81 = SMB type). If the flag is an 81 (SMB type), then the destination name at MSGODSTN in the message prefix was an SMB type when the message was originally created, but now the resource name is a CNT type. The destination control block address is in REG1 in the REG14-12 save area, and the block is at QTPDST.

X'1030'	MSG DEST NOT EXPECTED TRANSACT

Explanation

The message destination type is expected to be an SMB type.

Programmer response

Locate the message destination type flag (MSGDFLG2) of the message and it should be an SMB type (hex 81 = SMB type, hex 82= CNT type). If the flag is an 82 (CNT type), then the destination name at MSGODSTN in the message prefix was a CNT type (either an LTERM or MSNAME) when the message was created, but now the resource name is an SMB type. The destination control block address is in REG1 in the REG14-12 save area, and the block is at QTPDST.

X'1034'	DEST BLOCK NOT EXPECTED SMB

Explanation

The control block returned by FINDEST, representing the source name at MSGODSTN, is not an SMB (either a local or remote transaction code block).

Programmer response

Locate the MSGODSTN name in the message, and verify that it is a valid local or remote transaction code name. The control block address is in REG1 in the REG14-12 save area, and the block is at QTPDST.

X'1038'	ETO NEEDED BUT NOT SUPPORTED

Explanation

The extended terminal option (ETO) was determined to be needed but was not available.

Programmer response

None. This error is not currently set.

X'103C'	DEST LNB SID/DEST MSG SID NOMTCH

Explanation

The message is enqueued to an MSC logical link MSNAME, and the destination SYSID of the message does not match the destination SYSID of the MSNAME.

Programmer response

Locate the MSC destination name in the message (MSGMSONM in the MSC prefix). It should be an MSC MSNAME. The LNB control block that represents this MSNAME has a different destination SYSID than the message destination SYSID at MSGMSOID. The probable cause of the problem is that the MSNAME destination SYSID has been changed. The LNB control block address is in REG15 in the REG14-12 save area, and the block is at QTPDST.

X'1040'	FINDEST ERROR FOR DEST = MSGMSONM

Explanation

The MSC destination name in the message at MSGMSONM could not be found by the FINDEST routine.

Programmer response

Locate the MSGMSONM name in the message and verify that it is a valid local LTERM, MSNAME, or local or remote TRANSACTION CODE. If the extended terminal option (ETO) was specified at SYSGEN and name is a dynamic LTERM, verify that the ETO is enabled. The FINDEST parameter list used to locate the name is at PSTDCA.

X'1044'	MSC DEST BLOCK NOT EXPECTED CNT

Explanation

The control block returned by FINDEST representing the source name at MSGMSONM is not an LTERM CNT.

Programmer response

Locate the MSGMSONM name in the message prefix, and verify that it is a valid local LTERM. The CNT control block address returned by FINDEST is in REG1 in the REG14-12 save area, and the block is at QTPDST.

X'1048'	MSG DEST NOT EXPECTED TRANSACTION

Explanation

The message destination type flag associated with the MSGODSTN name is expected to be an SMB type because the destination control block is an SMB.

Programmer response

Locate the message destination type flag (MSGDFLG2) of the message and verify that it is an 82. This indicates the MSGODSTN destination name was a CNT type when the original message was created. However, the resource control block returned by FINDEST returned an SMB type control block. The probable cause of the problem is that the destination was changed from an LTERM or MSNAME type to a transaction code type. The control block address is in REG1 in the REG14-12 save area, and the block is at QTPDST. The parmlist passed to FINDEST is in the PSTDCA area.

X'104C'	DEST SMB SID/DEST MSG SID NOMTCH

Explanation

The message is enqueued to a transaction code SMB and the destination SYSID of the message does not match the destination SYSID of the SMB.

Programmer response

None. This error is not currently set.

X'1050'   **DEST CONV BUT NO SPA SEG IN MSG**

Explanation

The message destination is an IMS conversational transaction code, but the message does not contain a scratch pad (SPA) segment.

Programmer response

Locate the message destination name in the MSC prefix at MSGMSONM. This name is a conversational transaction code. The SMB address for the transaction code is in REG1 in the REG14-12 save area, and the SMB block is at QTPDST. The MSG2SPA flag in the MSC prefix should be set on to indicate the message contains a scratch pad (SPA). However, the flag is not set. The probable cause of the problem is that the transaction code was changed from non-conversational to conversational.

X'1054'   **DEST NOT CONV BUT MSG HAS SPASEG**

Explanation

The message flag MSG2SPA is set indicating a conversational SPA segment is included in the message, and the destination transaction code is not an IMS conversational transaction code.

Programmer response

Locate the MSG2SPA flag in the MSC prefix of the message, and it should be set on. The transaction code is in the MSC prefix at MSGMSONM. REG1 in the REG14-12 save area is the SMB address for the transaction code, and it is not an IMS conversational transaction code. The SMB block is at QTPDST. The probable cause of the problem is that the transaction code was changed from conversational to non-conversational.

X'1058'   **DEST = BLANKS AT CALL QMGR TIME**

Explanation

The destination in the modifiable TPPCB was not set.

Programmer response

The message queue manager is being called to the message to a queue manager buffer, and the destination name in the TPCB at TPCBTSYM has not been set. This is an IMS internal error.

X'105C'   **DEST NAME INVLD AT CALLQMGR TIME**

Explanation

The destination invalid flag in the TPPCB has not been reset.

Programmer response

The message queue manager is being called to the message to a queue manager buffer, and the destination invalid flag (TPCBSMBN) is still set on. This is an IMS internal error.

X'1060'   **NON ZERO RC ON ISRT CALL TO QMGR**

Explanation

The message queue manager was called to the message into a queue manager buffer, and a nonzero return code was returned.

Programmer response

The queue manager return code is in REG15 of the REG14-12 save area. The probable cause of the problem is that the message queue buffer is too small to hold the message prefix and segment. Check the large message queue data set block size, and determine if it has been reduced from the size when the message was originally created. The length of the message prefix and segment is contained in the first two bytes of the message in the I/O area. If the message queue block size is large enough, the message length is correct, and the message queue data sets are not full, then this is probably an IMS internal error.

X'1064'   **MSG CONTAINS INVALID QUEUE NUMBER**

Explanation

The queue number of the message is invalid.

Programmer response

Locate the message queue number in the message prefix at MSGFLAGS (low order 4 bits of the flag). A queue number greater than 5 is invalid. Determine the queue number source. Observe the following rules:

1. If the QCF recovery mode is RECOVERDM or RECOVERAB and the source of the message is a 4002 DUMPQ or SNAPQ record, the queue number is obtained from the 4002 record by IQCSELCT.

2. If the QCF recovery mode is RECOVERDM or RECOVERAB and the source of the message is a 01 or 03 record, the queue number is obtained from the type 35 enqueue record by IQCSELCT.

3. If the QCF recover mode is REPROCESS, the queue number is 0 in the 01 or 03 record and should have been set by IQCQMRQ0 to either:

   • 1 if the destination is a transaction code

   • 4 for all other destination types

   This is either an IMS or QCF internal error.

X'1068'	MSGMSINM BLOCK NOT CNT TYPE

Explanation

The control block returned by FINDEST, representing the source name at MSGMSINM, is not an LTERM CNT.

Programmer response

Locate the MSGMSINM name in the message prefix, and verify that it is a valid local LTERM. The CNT control block address returned by FINDEST is in REG1 in the REG14-12 save area, and the block is at QTPDST.

X'106C'	DFSSLC CALL ERR FOR DST MSGMSONM

Explanation

An error was detected while attempting to locate the resource control block for the resource name at MSGMSONM in the message prefix.

Programmer response

This is probably an IMS internal error. The return code returned by the locate call is in REG15 of the REG14-12 save area. The locate parameter list is in the PSTDCA area.

X'1070'	DFSSLC CALL ERR FOR DST MSGIDSTM

Explanation

An error was detected while attempting to locate the resource control block for the resource name at MSGIDSTN in the message prefix.

Programmer response

This is probably an IMS internal error. The return code returned by the locate call is in REG15 of the REG14-12 save area. The locate parameter list is in the PSTDCA area.

X'1074'	DFSSLC CALL ERR FOR DST MSGMSINM

Explanation

An error was detected while attempting to locate the resource control block for the resource name at MSGMSINM in the message prefix.

Programmer response

This is probably an IMS internal error. The return code returned by the locate call is in REG15 of the REG14-12 save area. The locate parameter list is in the PSTDCA area.

X'1078'	DFSSLC CALL ERR FOR DST MSGODSTN

Explanation

An error was detected while attempting to locate the resource control block for the resource name at MSGODSTN in the message prefix.

Programmer response

This is probably an IMS internal error. The return code returned by the locate call is in REG15 of the REG14-12 save area. The locate parameter list is in the PSTDCA area.

X'107C'	APPC NEEDED BUT NOT SUPPORTED

Explanation

The message was determined to be an LU 6.2 APPC type. However, the APPC message prefix segment was not present or could not be located.

Programmer response

Locate the message. The MSGC1APP flag should be set on, indicating the message is an APPC type. The APPC prefix segment with a segment type (MSGSIID) of X'87' should be present in the message prefix. This is probably an IMS internal error.

X'1080'	MSG DEST = APPC SYNC = NON RECOV

Explanation

The message destination, which is either an LU 6.2 synchronous logical unit (LU) name or an OTMA client with a send-then-commit (commit mode 1 or CM1) protocol, is considered nonrecoverable.

Programmer response

Locate the MSGODSTN name field in the message prefix; it should start with FDFFFFFF, indicating that the destination of the message is either of the following:

- Case A: An LU 6.2 (APPC) logical unit in LU 6.2 synchronous conversation mode
- Case B: An OTMA with a send-then-commit (commit mode 1 or CM1) protocol

This message is nonrecoverable according to the LU 6.2 protocol or the OTMA send-then-commit protocol and is discarded by the MRQ processor (IQCQMRQ0).

- Case A: The LUNAME destination is in the APPC message prefix segment and is extracted and reported in the load function messages discarded by the destination report.
- Case B: The TMEMBER and TPIPE names are in the APPC message prefix segment and are extracted and reported in the load function messages discarded by the destination report.

In either case, this is a normal condition and is not considered an error.

X'1084' **MSG DEST = NON RECOV**

Explanation

The message destination is nonrecoverable because of one of the following reasons:

- The destination transaction code name was defined as NORECOV
- The message was received from an LU 6.2 in synchronous conversation mode, which implies nonrecoverable
- The message was received from an OTMA client with a send-then-commit (commit mode 1 or CM1) protocol, which is always treated as nonrecoverable

Programmer response

Locate the MSGFLAGS byte in the message prefix of the message. MSGFNRQU should be set indicating that the message is nonrecoverable. Some possible reasons are:

- If the message destination is local (system is not MSC, or it is MSC and the destination SYSID at MSGMSOID in the MSC segment item is local), then determine whether the destination name at MSGODSTN is a nonrecoverable transaction code.
- If the message destination is remote (system is MSC and the destination SYSID at MSGMSOID in the MSC segment item is remote), then determine whether the destination name at MSGMSONM in the MSC

prefix segment item is a nonrecoverable transaction code.

- If the source name in the message prefix at MSGIDSTN starts with an FDFFFFFF, then the source of the message is an LU 6.2 (APPC) logical unit in LU 6.2 synchronous conversation mode. This message is not recoverable according to the LU 6.2 protocol. The LUNAME destination is in the APPC message prefix segment and is extracted and reported in the LOAD messages discarded by the destination report.
- If all of the following conditions are satisfied, the message is an OTMA transaction and the commit mode is send-then-commit (CM1):
  - The source name in the message prefix at MSGIDSTN starts with an FDFFFFFF
  - The LUP_MSG_IS_OTMA flag is set in the LU 6.2 prefix
  - The TMAMHCM1 flag is set in the state data prefix

  This message is not recoverable according to the OTMA send-then-commit protocol.

  The TMEMBER and TPIPE names are in the APPC message prefix segment in the MRPREFIX area of the 6701-MRQE trace record.

All of these conditions are considered normal, not errors.

X'1088' **MSG WAS CANCELED BY IMS**

Explanation

The original message was canceled by IMS and was logged for accounting or message queue recovery purposes. The message text itself is not recovered.

Programmer response

Locate the MSGFLAGS byte in the message prefix, and MSGFCANC should be set on indicating the message had been canceled. The MSGODSTN field is the destination name of the canceled message. If MSC was specified at SYSGEN, an MSC segment item is present, and the SYSID at MSGMSOID in the MSC prefix segment item is a remote SYSID, then MSGMSONM in the MSC prefix segment item is the remote destination name. One possible cause is that an application program inserted the message and then issued an abend or issued a ROLL or ROLB call. This is a normal condition and is not considered an error. If a BROWSE, RECOVER, or UNLOAD is being processed, the message was obtained from the message queue and this condition should not occur.

X'108C' **ERROR LOCATING APPC ASYNC DEST**

Explanation

The destination name of the message was determined to be an LU 6.2 (APPC) asynchronous destination, and a call to the IMS LU 6.2 interface routine encountered an error while trying to locate the LU destination.

Programmer response

Locate the MSGODSTN destination name in the message prefix; it should start with FEFFFFFF, indicating that the destination type is an LU 6.2 (APPC) asynchronous destination. The return code from the LU 6.2 interface is in REG15 in the REG14-12 save area. The parameter list passed is in the PSTDCA area. The message should contain an LU 6.2 prefix item with a type code of X'85'. The LU 6.2 destination name is stored in the LU 6.2 prefix item. Determine whether APPC is correctly installed and enabled, and that the destination name is an LU 6.2 logical unit. Correct these items if necessary. Otherwise, this is probably an IMS internal error.

X'1090'	MSGMRQF1 FLAG INVALID

Explanation

The MSGMRQF1 flag in the MRQ prefix passed to the IMS message requeuer processor (IQCQMRQ0) by the QCF client routine (load function) is invalid.

Programmer response

The MSGMRQF1 flag byte is in the MRQ prefix segment (MSGMRQPF), and is in front of the prefix of the message being inserted. The flag byte should be zero or a multiple of hex 4. This is either an IMS or QCF internal error.

X'1094'	MSC DEST BLOCK NOT EXPECTED LNB

Explanation

The destination of the message was determined to be an MSC MSNAME resource. However, the destination control block found by FINDEST was not an LNB.

Programmer response

Locate the message and it should have an MSC prefix segment item with a segment code of X'8C' (MSGSIID=8C). the destination SYSID in MSGMSOID in the MSC segment item should be remote. MSGODSTN is the MSNAME of the message destination, and it should be an LNB control block. REG15 in the REG14-12 save area is the address of the expected LNB, and the LNB is at QTPDST. The probable cause

of the problem is that the destination MSNAME was changed to an LTERM name or transaction code.

X'1098'	SOURCE/DEST = DFSAPPC INVALID

Explanation

The destination name of DFSAPPC is invalid.

Programmer response

This error is currently not being set.

X'109C'	LU 6.2 SCD EXTEN INVALID/ NOTAVAIL

Explanation

The message was determined to be an LU 6.2 (APPC) type. However, the APPC SCD extensions could not be located.

Programmer response

Locate the message; the MSGCFLG2 byte of the message prefix segment should be set on indicating one of the following:

- An LU 6.2 segment is present (MSGC2APP is set on)
- The destination name at MSGODSTN or MSGMSONM is DFSAPPC

Field SCDLSCD in the SCD was zero. This is either an IMS internal error, or APPC is not correctly installed.

X'10A0'	MSG NOT VALID 01/03 TYPE

Explanation

The message being passed by the load function or being processed by BROWSE, RECOVER, or UNLOAD is not a valid type 01 or 03 message.

Programmer response

Locate the message and verify that the MSGLCODE byte is either a 01 or a 03 and that the message prefix includes at least a basic segment prefix item (the length of the basic segment prefix item differs by release, see the ILOGREC macro), and a system segment prefix item (prefix segment item following the basic prefix segment, MSGSIID = 81 hex), and the MSGDFLG2 flag byte is either an 81 (transaction code type destination), or an 82 (LTERM, MSNAME, APPC/LU 6.2, or USERID type of destination). This is probably an IMS or QCF internal error. Locate and examine the original message input to IQCSELCT. If a BROWSE, RECOVER, or UNLOAD is being processed, the message was obtained from the message queue.

X'10A4'   INTERNAL IMS MESSAGE

Programmer response

Locate the message in the I/O area and verify that the destination name at MSGODSTN or MSGMSONM is an internal IMS destination. Current internal destination messages are:

MSNS1
> **/MSVERIFY** messages that are canceled with AIBREASN = X'00001100'.

MSNS2
> APPC/OTMA response messages destined to a back-end shared queues IMS environment.
>
> **Note:** These should not be canceled.

MSNS3
> Not used.

MSNS4
> Not used.

These messages are normal and are not considered errors.

X'10A8'   SOURCE/DEST NAME CHANGED

Explanation

The name in the control block representing the source name of the message (LTERM name) or destination name of the message (LTERM or TRANCODE name) does not match the name in the message.

Programmer response

The control block found by IMS representing either the source LTERM or destination LTERM or TRANCODE is pointed to by REG14 in the register save area. The message is in the I/O area and is also pointed to by REG6. The name in the control block is at offset X'1C', and it does not match either the source field (MSGIDSTN) or destination field (MSGODSTN) of the message. This is an internal IMS failure.

X'10AC'   DFSLUMIF BLDPRE ERROR

Explanation

A nonzero return code was returned by the IMS APPC LUM services routine while trying to build a new APPC prefix for an APPC message.

Programmer response

The APPC message being processed is in the I/O area, and is also pointed to by REG6 in the register save area. The nonzero return code from the LUM services routine is in REG15. This is an internal IMS failure.

X'10B0'   ERROR GETTING DFSPOOL STORAGE

Explanation

The DFSPOOL call received a nonzero return code while attempting to get or release storage from the HIOP storage pool for the QMRQWORK area or a buffer.

Programmer response

REG15 = the return code from DFSPOOL call. This is either an internal error, or there is insufficient storage available in the control region private area.

X'10B4'   ERROR GETTING AN AWE

Explanation

A DFSBCB GET or REL for an AWE block received a nonzero return code.

Programmer response

REG15 = return code from the DFSBCB GET call. This is either an internal error or insufficient storage is available in the control region private area.

X'10B8'   NO EXTENDED PREFIX PRESENT

Explanation

The message being requeued did not contain an extended prefix segment when one was expected.

Programmer response

The message was expected to contain an extended prefix segment (MSGC2EPH=1). However, none existed (QMRQEPHP=0). The QMRQWORK address is from REG5, and the message address is in REG6. If the message being processed is from IMS 5.1 or later, this prefix item should exist. Analyze the message and its prefix segments. This is probably an IMS internal error.

X'10BC'   ERROR INIT/ADDRESSING QMRQWORK

Explanation

An error occurred while getting the QMRQWORK area and initializing it with the current message information.

Programmer response

Look for a previous type 6701-MRQE error record that indicates another more specific error (in other

words, AIBREASN in DFSAIB is not equal to 10BC). There should be one. This error is logged when the caller (LOAD) receives control back from QMRQINIT and REG15 is nonzero. QMRQINIT should have logged a 6701-MRQE record when the specific error was detected.

X'10C0'	CAN'T FIND RACF SEGMENT MSGSORAC

Explanation

The message flag indicates a RACF prefix segment is present, but the segment cannot be located.

Programmer response

Locate the message and verify that flag MSGC1RAC is set. If it is set, then the RACF prefix segment with code =X'88' must be present. This is an internal IMS error.

X'10C4'	CAN'T FIND LU 6.1 SEGMENT MSGSILU6

Explanation

The message flag indicates that an LU 6.1 prefix segment is present, but the segment cannot be located.

Programmer response

Locate the message and verify that flag MSGC2LU6 is set. If it is set, then the LU 6.1 prefix segment with a code = 84 hex must be present. This is an internal IMS error.

X'10C8'	CAN'T FIND APPC SEGMENT MSGSOAPO

Explanation

Message flag indicates APPC prefix segment is present, but the segment cannot be located.

Programmer response

Locate the message and verify that flag MSGC2APP is set. If it is set, then an APPC prefix segment with a code = 85 hex must be present. This is an internal IMS error.

X'10CC'	CAN'T FIND EPH SEGMENT MSGSIEPH

Explanation

The message flag indicates that the EPH prefix segment is present, but the segment cannot be located.

Programmer response

Locate the message and verify that flag MSGC2EPH is set. If it is set, then an EPH prefix segment with a code=86 hex must be present. This is an internal IMS error.

X'10D0'	CAN'T FIND APPC SEGMENT MSGSIAPO

Explanation

The message flag indicates that the APPC prefix segment is present, but the segment cannot be located.

Programmer response

Locate the message and verify that flag MSGEAPPC is set. If it is set, then an APPC prefix segment with a code = 87 hex must be present. This is an internal IMS error.

X'10D4'	CAN'T FIND SEC SEGMENT MSGSISEC

Explanation

The message flag indicates that the SEC prefix segment is present, but the segment cannot be located.

Programmer response

Locate the message and verify that flag MSGESEC is set. If it is set, then an SEC prefix segment with a code = 88 hex must be present. This is an internal IMS error.

X'10D8'	CAN'T FIND WLM SEGMENT MSGSIWLM

Explanation

The message flag indicates that the WLM prefix segment is present, but the segment cannot be located.

Programmer response

Locate the message and verify that flag MSGEWLM is set. If it is set, then an WLM prefix segment with a code = 89 hex must be present. This is an internal IMS error.

X'10DC'	CAN'T FIND SYS EXT SEGMENT MSGSISEX

Explanation

The message flag indicates that the SYS EXT prefix segment is present, but the segment cannot be located.

Programmer response

Locate the message and verify that flag MSGESEX or MSGETMR is set. If it is set, then a SYS EXT prefix segment with a code = 8A hex must be present. This is an internal IMS error.

X'10E0'	CAN'T FIND MSC EXT SEGMENT MSGSIMEX

Explanation

The message flag indicates that the MSC EXT prefix segment is present, but the segment cannot be located.

Programmer response

Locate the message and verify that flag MSGEMEX or MSGETMR is set. If it is set, then an MSC EXT prefix segment with a code = 8B hex must be present. This is an internal IMS error.

X'10E4'	OTMA MESSAGES NOT SUPPORTED

Explanation

The IMS release message being requeued either does not support OTMA messages, or the OTMA feature is not SYSGENed.

Programmer response

Locate flag MSGFLAGA in the QMRQWORK area to determine the "from" and "to" IMS release of the message. The IMS release must be version 5.1 or later.

X'10E8'	MSC/APPC MESSAGE NOT SUPPORTED

Explanation

Message is a remote MSC message that originated from an APPC LU 6.2 session, and is not supported on this release.

Programmer response

Locate flag QMRWFLGA in the QMRQWORK area and determine the "to" IMS release. It must be version 5.1 or later. The destination SID in the message prefix (message prefix pointed to by REG6) is remote, as indicated by QMRWFLG6 in the QMRQWORK area. The problem is either because the destination of the message is changing from local to remote, or because an MSC/APPC message was requeued from IMS 5.1 or later to a lower IMS release, for example, 4.1 (IMS "from" release is also set in QMRWLAGA). The QMRQWORK pointer is in REG5.

X'10EC'	MESSAGE REROUT NOT SUPPORTED

Explanation

IQCQMRQ0 is being called with a reroute function that is not supported in this release of IMS.

Programmer response

Internal error. Trace back to caller of IQCQMRQ0.

X'10F0'	ISRT - MSC SEG ITEM NOT PRESENT

Explanation

The message is an MSC message, but the MSC or TMR message prefix segment items could not be located.

Programmer response

The destination changed from local to remote after the original message was built, or the message was incorrectly built by IMS. Locate the message and determine whether the MSGODSTN or MSGMSONM destinations are remote, or whether the MSGC2MSC flag is set in the system prefix (type 81) but the message does not contain an MSC (type 82) or TMR (type 8C) prefix segment.

X'10F4'	ERROR CREATING DYNAMIC LNB

Explanation

IQCQMRQ0 called the create dynamic LNB routine, but the create was unsuccessful.

Programmer response

Locate the message and verify that the destination name at MSGODSTN is a valid MSNAME and a unique name in the IMS system.

X'10F8'	CANNOT FIND SYS PREFIX SEG MSGSIPEX

Explanation

The system prefix segment could not be located.

Programmer response

Locate the message and verify that the system prefix is present following the basic prefix. The system prefix code is 81 in hexadecimal format. The message is not valid without a system prefix. This is an internal error.

X'10FC'	ERROR LOADING MODULE DFSTSPCO

Explanation

The UTC-to-LOCAL time conversion routine could not be loaded.

Programmer response

Verify that module DFSTSPCO is in the IMS RESLIB and is executable.

X'1100'	ISRT - /MSV CMD MESSAGE CANCELLED

Explanation

The message was an **/MSVERIFY** command message and was canceled.

Programmer response

Messages containing **/MSVERIFY** data are canceled by MRQ because the data might no longer be valid. This is a normal condition.

X'1104'	INIT - QMRQWORK ERROR SHOULD NOT OCCUR

Explanation

The QMRQWORK area:

- Could not be located
- Was invalid, or
- Should not exist (it was not freed or cleaned up after a prior use).

Programmer response

QMRQWORK is pointed to by QSAPWKAD, which should either be zero or an address in the HIOP pool. The usage varies depending on the function (QMRWFCN, QMRWFCN2) being performed. This is an internal IMS error.

X'1108'	MSGPROC - CAN'T FIND TMR PREFIX MSGMSC

Explanation

The message flag indicates the TMR prefix segment is present, but the segment cannot be located.

Programmer response

Locate the message and verify that the prefix segment exists. The TMR prefix segment code is 8C. This is an internal IMS error.

X'1110'	INIT - INVALID INCLUDE/ EXCLUDE ENTRY

Explanation

DFSQMR10 received an invalid include or exclude entry type from the QCF BMP.

Programmer response

The MRQSELCT pointer in MRQWORK points to the include or exclude table. Reg4 in QMRWESAV in MRQWORK contains the address of the entry (MRSELROW) in error. This is an IMS or MRQ error.

X'1114'	INIT - INVALID MRQWORK INIT CALL

Explanation

A DFSQMR10 QMRQINIT call was made to reinitialize the work area (MRQWORK). However, the call request is invalid.

Programmer response

Currently, MRQWORK is reinitialized at each new or command call. Reg10, byte 2, in the REG0-15 save area contains the call type (should be MRQISRT (04) or MRQCMD (38)). See macro DFSQMGR for a list of the QCF function codes (QMRWFCN). R10, byte 3, is the QMRQINIT code (1C). Trace the call back to the caller of DFSQMR10/QMRQINIT. This is an internal IMS/MRQ error.

X'1118'	ISRT - QBUFF DIDN'T FIT IN DEP RGN COMM AREA

Explanation

A message queue buffer was received from the QCF client for a load/ request, and the PSTVS0 flag was set to indicate it did not fit in the BMP-to-IMS region communications area.

Programmer response

The size of the dependent region communications area (DIRCA, also called PSBNDXSZ) is supposed to be sufficient to handle the largest QBUF, so this condition indicates an internal IMS error.

X'111C'	ISRT - DFSRAC6 ERROR GETTING UTOKEN FOR APPC

Explanation

A message is being inserted and is being converted to APPC. RACF was called and issued a RACROUTE REQUEST=VERIFYX to obtain a UTOKEN for the APPC prefix of the message. The RACF call returned a nonzero return code.

Programmer response

The return code from the call to the IMS RACF interface routine (DFSRAC60) is in R15 in the REG0-15 save area. R1 is the PARMLIST address, which is the QMRWLWA2 area within MRQWORK. PARMLIST+1C=USERID, PARMLIST+20=GROUPNAME, PARMLIST+48=APPC PLUNAME.

X'1120'	ISRT - QUEUE BUFFER FLAG ERROR

Explanation

An error was detected in either the MRQ prefix Z2 flag (MRPREZZ2) or the buffer flag2 (MSGCFLG2) passed by the QCF /load BMP.

Programmer response

The MRQ prefix that precedes the Qbuffer/Message being loaded or inserted, did not have the correct first or last flags (that is, bits MRPZZFST or MRPZZLST in flag MRPREZZ2). See the DFSMRQPF macro for MRQ prefix mapping. The MRQ prefix address is in R4 of the REG0-15 save area. Or the message Qbuffer MSGPRFX did not have the correct first or last flags (that is, bits MSGFFRST or MSGFLAST in flag MSGFLAGS). See macro QLOGMSGP for the mapping of this prefix. The Qbuffer prefix address is in R6 of the REG0-15 save area.

If a spanned buffer is being passed, then the previously inserted buffer might be the one with the incorrect flags. The previous buffer flags and status is saved in fields QMRWBF1 and QMRWBF2 of the MRQWORK area. The MRQWORK area is in R5 of the REG0-15 area and is mapped by the DFSMRQWK macro.

This is either a QCF or IMS internal error, or the data being passed to IMS using the QCF /load function is invalid.

X'1124'	ISRT - SEGMENT FLAG ERROR

Explanation

An error was detected in the segment Z1 flag (MSGXFLG1) in the queue buffer passed by the QCF / load BMP.

Programmer response

The segment of the message about to be inserted in the message queue was determined to have an invalid first or last flag (that is, bits MSGX1FST or MSGX1LST in flag MSGXFLG1). The segment address is in R4 of the REG0-15 area and is mapped by the QLOGMSGP macro. If this is a spanned segment, the incorrect flag setting might be in the previous segment of the message being inserted. The previous segment flag and status is saved in fields QMRWSG1 and QMRWSG2 of the MRQWORK area. The MRQWORK address is in R5 of the REG0-15 save area and is mapped by the DFSMRQWK macro.

This is either a QCF or IMS internal error, or the data being passed to IMS using the QCF /load function is invalid.

X'1128'	ISRT - INVALID QUEUE BUFFER DETECTED

Explanation

An invalid queue buffer was passed by the QCF client on a LOAD function.

Programmer response

The QBUFFER received by LOAD is validated by adding up the prefix length and the segment lengths and comparing them to the qbuffer length. If the two are not equal, the buffer is considered invalid. The buffer address is in REG6 in the REG0-15 save area. The buffer is mapped by the QLOGMSGP macro. This error is either a QCF or IMS internal error, or the data being passed to IMS by the QCF LOAD function is invalid.

X'112C'	INIT/CLEANUP - ISWITCH FAILURE

Explanation

Either an ISWITCH to the CTL region or an ISWITCH RETURN to the MRQ/QCF dependent region failed.

Programmer response

Save area QMRWESAV in MRQWORK contains the registers at the time of the error (R0 - R15). R15 is the ISWITCH error code. R1 is the module ID that was issuing the ISWITCH.

- QMR1 = DFSQMR10
- QMR7 = DFSQMR70
- QMR9 = DFSQMR90

If flag QMRWFLG9=QMRW9SWI is off, this indicates this is an ISWITCH to the CTL region failure. If flag QMRWFLG9=QMRW9SWI is on, this is an ISWITCH RETURN to the MRQ/QCF dependent region failure. This is an internal IMS error.

X'1130'	ISRT QBUFF THRESHOLD EXCEEDED

Explanation

The ISRT queue buffer threshold count was exceeded.

Programmer response

This condition is detected by either a user queue space notification exit routine (DFSQSPC0) or the QCF Queue Space Notification exit routine (DFSQMRI0, which is activated by link–editing IQCQMRH0 to IMS RESLIB as DFSQMRH0). In either case (user or QCF exit routine), the exit set flag QMGROFL3=QMGRO3NO in the DFSQMGR parameter list that was built in the PSTDCA area. The exit routine set this flag because the current message being inserted caused the qbuffer threshold to be exceeded. The count of long/short queue buffers being inserted by this message is at fields QMRLBCNT/QMRSBCNT in the DFSMRQWK work area.

Take one of the following actions:

- Reload the messages when queue usage is not as high
- Increase the size of the queue data sets
- Increase the value for the threshold exceeded

X'1134'	UNABLE TO FIND/CREATE TPIPE

Explanation

A nonzero return code was returned by the IMS OTMA services routine while trying to find or create a new OTMA TPIPE control block.

Programmer response

The nonzero return code from the OTMA services routine is in REG15. This is an internal IMS failure.

X'1138'	INSERT - DFSUSE NON ZERO RETURN CODE

Explanation

A DFSUSE FUNC=INUSE call for an OTMA TPIPE YQAB was issued to prevent the YQAB from being freed during an IMS CHKPT, and the DFSUSE call received a nonzero return code. The message being inserted is rejected.

Programmer response

This is probably an internal IMS error. IMS Queue Control Facility issues DFSUSE FUNC=INUSE calls when inserting to an OTMA TPIPE, to prevent the TPIPE YQAB from being freed by IMS at checkpoint, while the messages are still on the temporary destination.

The INUSE return code from DFSUSE00 is in REG15 in the REG14-12 area of the 6701-MRQE trace record.

X'2004'	PURGE PCB DEST INVALID

Explanation

The message is being purged (enqueued to a temporary destination), and the temporary destination name has not been set to valid.

Programmer response

The destination invalid flag (TPCBSMBN) in flag byte TPCBCODE is set on. This flag should have been reset during processing. If a queue manager buffer (QMBA) is allocated, the message that is being processed should be in this buffer. Otherwise, the message might have to be located on the SCRAPLOG data set where it is discarded by the Load function. The timestamp (date and time) of the message that is being processed is stored in the PST at PSTPRE1, and the timestamp can be used to locate the message on the SCRAPLOG data set, or in the original message input to IQCSELCT. This is an internal IMS or QCF error.

X'2008'	PURGE PCB DEST SET TO BLANKS

Explanation

The message is being purged (enqueued to a temporary destination), and the temporary destination name is blank.

Programmer response

The destination name in the TPPCB at TPCBTSYM is blank (hex 40s). This field should have been set to the destination name of the message during processing.

If a queue manager buffer (QMBA) is allocated, the message that is being processed should be in this buffer. Otherwise, the message might have to be located on the SCRAPLOG data set where it is discarded by the Load function. The timestamp (date and time) of the message that is being processed is stored in the PST at PSTPRE1, and the timestamp can be used to locate the message on the SCRAPLOG data set, or the original message input to IQCSELCT. This is an internal IMS or QCF error.

X'200C'   **PURGE DEST CTL BLK ADDR ZERO**

Explanation

The message is being purged (enqueued to a temporary destination), and the temporary destination control block address in the TPPCB is zero.

Programmer response

The destination name control block address is in the TPPCB at TPCBCNT and it is referred to as the QTPDST address. This field should have been set to the address of the destination name control block (the address of either the CNT, LNB, or SMB) during processing. If a queue manager buffer (QMBA) is allocated, the message that is being processed should be in this buffer. Otherwise, the message might have to be located on the SCRAPLOG data set where it is discarded by the Load function. The timestamp (date and time) of the message that is being processed is stored in the PST at PSTPRE1, and timestamp can be used to locate the message on the SCRAPLOG data set, or the timestamp can be used to locate the original message input to IQCSELCT. This is an internal IMS or QCF error.

X'2010'   **PURGE DEST NAME = DFS INVALID**

Explanation

The message is being purged (enqueued to a temporary destination) and the temporary destination name of the message starts with the reserved characters DFS.

Programmer response

The destination name in the TPPCB at TPCBTSYM starts with the characters DFS and is not a DFSAPPC destination message or other internal IMS destination. This is invalid. If a queue manager buffer (QMBA) is allocated, the message that is being processed should be in this buffer. Otherwise, the message might have to be located on the SCRAPLOG data set where it is discarded by the Load function. The timestamp (date and time) of the message that is being processed is

stored in the PST at PSTPRE1, and the timestamp can be used to locate the message on the SCRAPLOG data set, or the timestamp can be used to locate the original message input to IQCSELCT. This is probably an internal IMS error.

X'2014'   **PURGE INQUIRY DEST NOT SIGNED ON**

Explanation

The message is being purged (enqueued to a temporary destination), and the temporary destination name of the message is an inquiry type LTERM.

Programmer response

The destination name in the TPCBTSYM is an inquiry type LTERM destination and is not signed on. The destination control block CNT is in REG6 in the REG14-12 save area, and the CNT2INQ flag is set on (destination is inquiry type). The CNT control block is at QTPDST. THE CTB is in REG7 of the REG14-12 save area, and CTB1DIAL and CTB1SIGN are set off (terminal is not signed on). Messages destined to an inquiry LTERM that is not signed on are discarded according to protocol. This is considered normal operation.

X'2018'   **PURGE NON 0 RC ON QMGR ENQ CALL**

Explanation

The message is being purged (enqueued to a temporary destination), and a nonzero return code was received from the message queue manager on the enqueue call.

Programmer response

The message queue manager return code is in REG15 of the REG14-12 save area. The message queue buffer is in the QMBA area. This is probably an internal IMS error.

X'201C'   **PURGE I/O AREA INVALID**

Explanation

The I/O area passed to the IMS MRQ processor (IQCQMRQ0) by the load function on the PURG call is invalid.

Programmer response

The I/O area passed on the PURG call does not begin with a valid MRQ prefix segment (MSGMRQPF). This is an internal QCF load function error.

X'2020' **PURGE MSGMRQF1 FLAG INVALID**

Explanation

The MSGMRQF1 flag in the MRQ prefix passed to the IMS message requeuer processor (IQCQMRQ0) by the QCF client routine (load function) is invalid.

Programmer response

The MSGMRQF1 flag byte is in the MRQ prefix segment (MSGMRQPF). The MSGMRQPF segment starts at the beginning of the I/O area. The flag byte should be a multiple of hex 4. This is either an IMS or QCF internal error.

X'2024' **DEST BLK=DFSAPPC BUT MSG NOT APPC**

Explanation

The message is being purged (enqueued to a temporary destination), and the destination name is DFSAPPC. However, the destination resource type is not an LU 6.2 (APPC) destination.

Programmer response

The resource name control block in REG6 in the REG14-12 save area contains a name of DFSAPPC, but the resource type flag in the TPPCB at flag byte TPPCBFLG is not set to type = APPC (TPPCB62 is not set on). The DFSAPPC CNT block is at QTPDST. This is an internal IMS error.

X'3000' **MESSAGE PREFIX SIZE INVALID**

Explanation

Either the total prefix size or one or more of the prefix segments of the message has an invalid length.

Programmer response

Locate the message being inserted in the I/O area. The segment address is in REG1 in the REG14-12 save area. The total prefix size is at offset 10 hex in the message. The current prefix segment address of the prefix segment being checked is in REG7 in the REG14-12 save area. The prefix segment length is the first two bytes. The prefix ID (MSGSIID) is the third byte. Locate this ID in the QLOGMSG DSECT and verify the size. If the message was obtained from a supported IMS system release, then this is probably an IMS internal error.

X'4000' **PREFIX SIZE GOTTEN NOT SIZE EXPECT**

Explanation

The message queue manager failed to obtain a message prefix the same size as that of the original message.

Programmer response

Locate the message being inserted in the I/O area. Field MSGPRFLL in the message prefix is the length of the original message prefix. Field QSAPPLTH in the QSAPWKAD area contains the length of the new message prefix. The lengths should be equal. This is an internal IMS error.

X'4004' **CPYPRFX - CAN'T FIND SYS PREFIX MSGSSEGM**

Explanation

The message prefix should contain a system prefix segment, but one could not be located.

Programmer response

Locate the message and verify that the system prefix segment exists. The system prefix code is 81. REG1 in the REG0-15 save area is the address of the prefix being copied. This is an internal IMS error.

X'4008' **CPYPRFX - CAN'T FIND TMR PREFIX MSGMSC**

Explanation

The message flag indicates that the TMR prefix segment is present, but the segment cannot be located.

Programmer response

Locate the message and verify that the prefix segment exists. The TMR prefix segment code is 8C in hexadecimal format. REG1 in the REG14-12 save area is the address of the prefix being copied. This is an internal IMS error.

X'400C' **CPYPRFX - CAN'T FIND SYS EXT PREFIX MSDMSE**

Explanation

The extended prefix area should contain an extended system segment, but one could not be located.

Programmer response

Locate the message, and verify that the prefix segment exists. The system extensions prefix segment code is

X'8A'. REG1 in the REG0-15 save area is the address of the prefix being copied. This is an internal IMS error.

X'4010'	CPYPRFX - CAN'T FIND THE MSC PREFIX MSGMSC

Explanation

The message flag indicates that the MSC prefix segment is present, but the segment cannot be located.

Programmer response

Locate the message and verify that the prefix segment exists. The MSC prefix segment code is X'82'. REG1 in the REG14-12 save area is the address of the prefix being copied. This is an internal IMS error.

X'5000'	NONZERO RC ON CANCEL CALL TO QMGR

Explanation

A nonzero return code was returned by the message queue manager while attempting to cancel a message queue buffer that is being discarded (the message is being scrapped).

Programmer response

An error was detected while inserting a message to the message queue, and cleanup processing is being performed. The original error has already been logged in a prior type 6701-MRQE log record, and the queue buffer area is being released (canceled). The queue manager return code on the cancel call is in REG15 of the REG14-12 save area. This is an internal IMS error.

X'6004'	LOGREC TYPE NOT 4002, 01, OR 03

Explanation

The load function program read a log record that was not a valid type 4002 (DUMPQ or SNAPQ), 01 (input), or 03 (output) record, and discarded the record to the SCRAPLOG data set.

Programmer response

This error is detected by the load function routine, and is passed to the message requeuer processor to perform cleanup and to log the error in a 6701-MRQE record. The SCRAPLOG record that is written by the Load function needs to be located to determine if it is valid. The record might need to be traced back to the log data set that was the input to IQCSELCT. The QMBA area might contain part, or all of the message

that was being loaded when the invalid record was detected. This is either an IMS or QCF internal error.

X'6008'	NO SECONDARY LOGREC WHEN EXPECTD

Explanation

A message was being inserted that spanned multiple message queue buffers, and one of the secondary buffers could not be located.

Programmer response

This error is detected by the Load function routine, it is passed to the message requeuer processor to perform cleanup and to log the error in a 6701-MRQE record. The SCRAPLOG record that is written by the Load function needs to be located to reconstruct the chain of message buffers. The record might need to be traced back to the log data set that was the input to IQCSELCT. The QMBA area might contain part, or all of the message that is being loaded. This is either an IMS or QCF internal error.

X'600C'	SECONDARY LOGREC DEST INVALID

Explanation

A message was being inserted that spanned multiple message queue buffers, and one of the secondary buffers in the chain being processed by load function did not have the same destination name.

Programmer response

This error is detected by the load function routine, and is passed to the message requeuer processor to perform cleanup and log the error in a 6701-MRQE record. The SCRAPLOG record written by load function must be located to determine its validity and reconstruct the message buffer chain. The record may need to be traced back to the log data set input to IQCSELCT. This is either an IMS or QCF internal error.

X'6010'	MRQ/IMS - QBUF COUNT NOT EXPECTED NUMBER

Explanation

During transfer of a queue buffer between the QCF client and IMS, the count of buffers transferred (MRPCOUNT) was in error.

Programmer response

This error is detected by either IMS or the QCF client when transferring messages (QBUFs) during the

LOAD, BROWSE, RECOVER, and UNLOAD functions. The QBUF transfer count is incremented in the MRQ prefix count field (MRPCOUNT in the DFSMRQPF macro) and checked for one greater than the previous. If the count is not one greater, the error is issued and the QBUF transfer is rejected. This is either an IMS or QCF internal error.

X'6014'	MSGPROC - MSG WAS CANCELED BY IMS

Explanation

The original message was canceled by IMS and was logged for accounting or message queue recovery purposes. This error is similar to AIBREASN X'1088'; the difference is that this is a multi-buffer message, and part of the message was inserted to IMS by QCF when QCF detected the message was canceled. QCF issues a PURG DL/I call, requesting that IMS purge the message with AIBREASN X'6014'.

Programmer response

Locate the MSGFLAGS byte in the message prefix. MSGFCANC should be set on, indicating that the message was canceled. The MSGODSTN field is the destination name of the canceled message. If MSC was specified at SYSGEN and an MSC segment item is present and the SYSID at MSGMSOID in the MSC prefix segment item is a remote SYSID, then MSGMSONM in the MSC prefix segment item is the remote destination name. One possible cause is that an application program inserted the message and then either abended or issued a ROLL or ROLB call. This is a normal condition and is not considered an error if the message is being inserted (requeued) and was obtained from the log. If a BROWSE, RECOVER, or UNLOAD is being processed, the message was obtained from the message queue, and this condition should not occur.

X'7004'	CAN'T FIND PFX SEG MSGSISEX

Explanation

The message flag indicates that the system EXT prefix segment type 8A is present, but the segment cannot be located.

Programmer response

Locate the message and verify that flag MSGESEX is set. If the flag is set, then the MSC EXT prefix segment with a code equal to X'8A' must be present. The message being built that caused the error is pointed to by register 6 in the register 0-15 save area. This is an internal IMS error.

X'7008'	CANNOT FIND PFX SEG MSGSITMR

Explanation

The message flag indicates that the TMR prefix segment type 8C is present, but the segment cannot be located.

Programmer response

Locate the message and verify that the prefix segment exists.

X'700C'	CAN'T FIND PFX SEG MSGMSC

Explanation

The message flag indicates that the MSC prefix segment type 82 is present, but the segment cannot be located.

Programmer response

Locate the message and verify that the prefix segment exists. The MSC prefix segment code is 82 in hexadecimal format. This is an internal IMS error.

X'7010'	CAN'T FIND PFX SEG MSGMSCE

Explanation

The message flag indicates that the MSC prefix extensions segment type 8B is present, but the segment cannot be located.

Programmer response

Locate the message and verify that the prefix segment exists. The MSC prefix extensions segment code is 8B in hexadecimal format. This is an internal IMS error.

X'7014'	ERROR CONVERTING MESSAGE TIME

Explanation

A nonzero return code was returned from the time conversion routine while converting local time to UTC or UTC time to local.

Programmer response

The probable cause of the problem is an incorrect time field in one of the message prefixes. Locate the time conversion work area in QMRQWORK at label QMRDSTWK. If the error occurred converting local time to UTC, the return code is in QMRCVTM1. If the conversion is UTC to local, it is in QMRCVTM2. Return

codes are from either DFSCVTM or DFSTSPC. The local time being converted is at QMRWLOCL. The UTC time is at QMRDSUTC. The local time fields in the message are at MSGTMFAP and MSGMSCTS. The UTC times are at MSGUTC, MSGMSCEX, and MSGMSCTS.

X'7018'	**XLATPFX - CAN'T FIND PFX MSGEPHDR**

Explanation

The message flag indicates that the HEADER prefix extensions segment is present, but the segment cannot be located.

Programmer response

Locate the message and verify that flag MSGC2EPH is set and that the header prefix (type 86) exists. All messages from IMS 6.1 and later should contain this flag and prefix segment. This is an IMS error.

X'8004'	**QMR30 - BROWSE - SYSTEM NOT SHARED QUEUES**

Explanation

The BROWSE command was issued in a nonshared-queues environment, without QCF active.

Programmer response

The QCF BROWSE function is only supported in a nonshared-queues environment that supports QCF. Make the appropriate change.

X'8008'	**QMR30 - INVALID FUNCTION PASSED TO BROWSE**

Explanation

An invalid call was made to DFSQMR30. Register 0 did not contain X'28' (MRQCLEAN), X'34' (MRQCMD), or X'38' (MRQGCMD).

Programmer response

REG0 in the REG0-REG15 save area contains the function code. REG14 is the address of the caller of DFSQMR30 (BALR register). Trace the call back to the caller of DFSQMR30.

X'800C'	**QMR30 - BROWSE RECEIVED ERROR CODE FROM SELECT**

Explanation

DFSQMR20 returned with RC=X'08' in the QNAME selection call.

Programmer response

Trace back to DFSQMR20 to determine the cause.

X'8010'	**QMR30 - BROWSE COMMAND ERROR**

Explanation

The CQS BROWSE of the ready queue returned partial data. The size of the data object was larger than 32K.

Programmer response

Locate the message and determine its origin. REG8 in the REG0-REG15 save area contained the message address.

X'8014'	**QMR30 - BROWSE COMMAND ERROR**

Explanation

No data objects were returned when the CQS BROWSE of the READY queue was performed.

Programmer response

Dump the associated queue to verify whether any message exists. If the message exists, this is an internal error.

X'8018'	**QMR30 - BROWSE COMMAND ERROR**

Explanation

The CQS BROWSE of the ready queue failed.

Programmer response

This is an internal error. Locate the CQS reason code (CQSRSNCD) in the parameter list (QMRWLWA) to determine the cause.

X'801C'	**QMR30 - BROWSE COMMAND ERROR**

Explanation

DFSQMR20 returned with RC=X'08' in the message selection call.

Programmer response

Trace back to DFSQMR20 to determine the cause.

X'8020'	**QMR30 - BROWSE COMMAND ERROR**

Explanation

The message segment from the ready queue was not the first segment.

Programmer response

Locate the message and verify the contents. REG8 in the REG0-REG15 save area contained the message address. This is an internal IMS error.

X'8024'	QMR30 - BROWSE COMMAND ERROR

Explanation

The TMR prefix segment could not be located.

Programmer response

Locate the message and verify that the prefix segment exists. The TMR prefix segment code is X'8C'. REG8 in the REG0-REG15 save area contained the message address. This is an internal IMS error.

X'8028'	QMR30 - BROWSE COMMAND ERROR

Explanation

The CQS BROWSE of the staging queue returned partial data. The size of the data object was larger than 32K.

Programmer response

Locate the message and determine its origin. REG8 in the REG0-REG15 save area contained the message address.

X'802C'	QMR30 - BROWSE COMMAND ERROR

Explanation

No data objects were returned while CQS browsed the staging queue.

Programmer response

Dump the associated queue to verify whether any message exists. If so, this is an internal error.

X'8030'	QMR30 - BROWSE COMMAND ERROR

Explanation

The CQS BROWSE of the staging queue failed.

Programmer response

This is an internal error. Locate the CQS reason code (CQSRSNCD) in the parameter list (QMRWLWA) to determine the cause.

X'8034'	QMR30 - BROWSE COMMAND ERROR

Explanation

The message segment from the staging queue was not the middle segment.

Programmer response

Locate the message and verify the contents. REG8 in the REG0-REG15 save area contained the message address. This is an internal IMS error.

X'8038'	QMR30 - BROWSE COMMAND ERROR

Explanation

The message segment from the staging queue was not the last segment.

Programmer response

Locate the message and verify the contents. REG8 in the REG0-REG15 save area contained the message address. This is an internal IMS error.

X'803C'	QMR30 - BROWSE COMMAND ERROR

Explanation

The request was terminated because the RESYNC between IMS and CQS was not done.

Programmer response

Reissue the request after the RESYNC is done.

X'8040'	QMR30 - BROWSE INVALID DESTINATION

Explanation

The BROWSE request found that destination field BCURLNAM was zero. The destination is a required field for a BROWSE of the local queues.

Programmer response

This is an IMS error.

X'8044'   **QMR30 - BROWSE LOCAL QUEUES CONTROL BLOCK ERROR**

Explanation

The BROWSE request found that the QDEST block was in error. One of the following conditions was found:

- QDFLG1 indicated that there were messages on this control block, and QDQCBDQ was zero.
- QDFLG1 indicated that there were messages on this control block, and QDQCBDQ did not point to a queue block if the destination was a CNT type block.

Programmer response

This is an IMS error.

X'8048'   **QMR30 - BROWSE LOCAL QUEUES DESTINATION TYPE ERROR ON A MULTI-RECORD MESSAGE.**

Explanation

During processing of the second through the *n*th record of a multi-record message, the BROWSE request found that the destination address in the BCURLNAM field and in the current PCB were not the same.

Programmer response

This is an IMS error.

X'804C'   **QMR30 - BROWSE LOCAL QUEUES CONTINUATION TYPE ERROR ON A MULTI-RECORD MESSAGE.**

Explanation

During processing of the second through the *n*th record of a multi-record message, the BROWSE request found that the field pointed to by BMRQQPCB did not contain a valid token in the QTPRRN field.

Programmer response

This is an IMS error.

X'8050'   **QMR30 - QSN BLOCK ADDRESS IS ZERO**

Explanation

The BROWSE function was called to process the Queue Space Notification (QSN) queue and was passed an invalid or zero QSN block address in the MRCURQQSN field.

Programmer response

If MRCURQQSN is zero, this is probably an internal browse or select error. Trace back to where the field was set. If the field is invalid, this is most likely a bad QQSN block on the QQSN chain or an overlaid QQSN block. Verify this chain and the blocks on it. This is an internal IMS error.

X'8054'   **QMR30 - BROWSE AREA PARM NOT SET**

Explanation

The BROWSE request could not find a valid area to process. Valid areas are LOCAL, GLOBAL, OVERFLOW, and QUEUE SPACE NOTIFICATION (QSN). GLOBAL and OVERFLOW are valid for shared queues only.

Programmer response

Verify that a valid AREA is specified as LOCAL, GLOBAL, or OVERFLOW, or that QSN was specified on the function control card passed to QCF. If AREA is valid, verify that the area value was passed to IMS in the MQTYAREA field. If the area value is valid, then verify that IMS is processing a valid area as indicated by;

- Flag BMRQFLG2 if a BROWSE function was requested,
- Flag QMRQFLG2 if a QUERY function was requested and QUERY called BROWSE internally.

This is either a QCF or IMS error.

X'9004'   **QMR60 - QUERY - SYSTEM NOT SHARED QUEUES**

Explanation

A QUERY command was issued in a nonshared-queues environment.

Programmer response

The QCF query function is only supported in a shared-queues environment.

X'9008'   **QMR60 - INVALID FUNCTION PASSED TO QUERY**

Explanation

An invalid call was made to DFSQMR60. Register 0 did not contain X'28' (MRQCLEAN), X'34' (MRQCMD), or X'38' (MRQGCMD).

Programmer response

REG0 in the REG0-REG15 save area contains the function code. REG14 is the address of the caller of DFSQMR60 (BALR register). Trace the call back to the caller of DFSQMR60.

X'900C'	QMR60 - QUERY RECEIVED ERROR CODE FROM SELECT

Explanation

The call to DFSQMR20 to select a message queue name resulted in an error return code of 8 or more.

Programmer response

REG15 in the REG0-REG15 save area contains the return code. CMDQNAME in MRQWORK is the queue name being processed or the last queue name successfully processed. Trace the error back to DFSQMR20.

X'9010'	QMR60 - QUERY - CMD QUEUE TYPE INVALID

Explanation

An invalid queue type or no queue type was passed on the QUERY command call from the QCF client.

Programmer response

MQTYPQUE in MRQCMDWK contains either zero or the invalid queue types. The cold queue is an invalid query queue type. REG8 in the REG0-REG15 save area contains the MRQCMDWK address, which contains the query command from the QCF BMP.

X'9014'	NO MESSAGE RETURNED ON INTERNAL CALL TO BROWSE

Explanation

While querying either the APPC, OTMA, or cold queues, the query processor called Browse internally to get the message and extract information for the CQSQRYQT entry for the queue name. Browse returned a code indicating there were no messages on that particular queue.

Programmer response

REG15 in the REG0-15 save area contains the Browse return code of 4. REG6 is the address of the DFSSQQRY buffer containing queue names that have messages. REG7 is the current queue name entry that encountered the error. REG2 has the queue type from MQCURQNM in the MRSELWK area; the queue type is one of the following:

- 01 = APPC
- 02 = COLD
- 08 = OTMA

This is probably either an IMS or CQS error.

X'9018'	RETURN CODE ERROR ON INTERNAL CALL TO BROWSE

Explanation

While querying either the APPC, OTMA, or cold queues, the query processor called Browse internally to get the message and extract information for the CQSQRYQT entry for the queue name. Browse returned an error code for that particular queue.

Programmer response

REG15 in the REG0-15 save area contains the Browse return code. REG6 is the address of the DFSSQQRY buffer containing queue names that have messages. REG7 is the current queue name entry that encountered the error. REG2 has the queue type from MQCURQNM in the MRSELWK area; the queue type is one of the following:

- 01 = APPC
- 02 = COLD
- 08 = OTMA

This is probably either an IMS or CQS error.

X'901C'	ERROR LOCATING APPC/OTMA PFX

Explanation

While querying either the APPC or OTMA queues, a message was returned by an internal Browse call; a DFSMGPL request was issued to locate either the APPC or OTMA prefix of the message, to extract information from the prefix for the CQSQRYQT entry for the queue name. The DFSMGPL call encountered an error while trying to locate the prefix.

Programmer response

REG1 in the REG0-15 save area is the address of the message, which is in Browse buffer MRQBROMC. REG6 is the address of the DFSSQQRY buffer containing queue names that have messages. REG7 is the current queue name entry that encountered the error. The APPC/OTMA prefix is in the extended prefix area of the message and is a type X'87'. The message prefixes are mapped by macro QLOGMSGP. The message needs

to be analyzed to determine the error. This is an IMS error.

X'9020'   ERROR LOCATING TMR PREFIX

Explanation

While querying either the APPC or OTMA queues, a message was returned by an internal Browse call; a DFSMGPL request was issued to locate the transaction manager routing (TMR) prefix in the message to extract information from the prefix for the CQSRYQT entry for the queue name. The DFSMGPL call encountered an error while trying to locate the prefix.

Programmer response

REG1 in the REG0-15 save area is the address of the message in Browse buffer MRQBROMC. REG6 is the address of the DFSSQQRY buffer containing queue names that have messages. REG7 is the current queue name entry that encountered the error. The TMR prefix is in the extended prefix area of the message and is a type X'8C'. The message prefixes are mapped by macro QLOGMSGP. The message needs to be analyzed to determine the error. This is an IMS error.

X'9024'   SHOULD NOT OCCUR ERROR

Explanation

Query called the select processor (DFSQMR20) to select a queue to query. Select found that there are queues to process but did not return a query buffer or the COLDQ to process.

Programmer response

This condition should not occur. The select processor needs to be analyzed to determine the error.

X'9028'   ERROR FREEING BUFFER DURING CLEANUP

Explanation

During clean up of a query request at termination of the QCF client, the query buffer obtained by the DFSSQQRY call was freed with a DFSPOOL request; a nonzero return code was returned on the call.

Programmer response

The query buffer address is in REG3 of the REG0-15 save area. The return code is in REG15. This is an IMS error.

X'902C'   QMR60 - QUERY - QUERY CALLED BROWSE WITH A GET COMMAND.

Explanation

QUERY (DFSQMR60) called BROWSE (DFSQMR30) with a get command. BROWSE is not set up to handle this call.

Programmer response

This condition should not occur. The QUERY and/or BROWSE routine must be analyzed to determine the error. This is an IMS error.

X'9030'   QMR30 - BROWSE - QUERY CALLED BROWSE WITH A DESTINATION OF ZERO.

Explanation

QUERY (DFSQMR60) called BROWSE (DFSQMR30) with a destination of zero. A destination is required for the LOCAL queue.

Programmer response

This condition should not occur. The QUERY and/or BROWSE routine must be analyzed to determine the error. This is an IMS error.

X'9034'   QMR30 - BROWSE COMMAND ERROR

Explanation

The APPC/OTMA prefix segment could not be located.

Programmer response

Locate the message and verify that the prefix segment exists. The APPC/OTMA segment code is X'87'. REG8 in the REG0-REG15 area contained the message address. This is an internal IMS error.

X'9038'   QMR60 - QUERY - QUERY CALLED BROWSE WITH AN INVALID QNAME

Explanation

QUERY called BROWSE with a request to retrieve either an APPC or OTMA with an invalid queue name.

Programmer response

Locate the select work area (QMRQSETP) and validate that the queue name is invalid. This is an internal IMS error.

X'903C'   QMR60 - QUERY - QUERY CALLED BROWSE WITH AN INVALID QUEUE SPACE NOTIFICATION BLOCK

Explanation

QUERY (DFSQMR60) called BROWSE (DFSQMR30) with a queue space notification block address of zero. A queue space notification block address (MRCURQQSN) is required for a query of QSN.

Programmer response

This condition should not occur. The QUERY and/or BROWSE routine must be analyzed to determine the error. This is an IMS error.

X'A004'   QMR50 - UNLOAD - SELECT QUEUE NAME ERROR

Explanation

The call to DFSQMR20 to select a message queue name resulted in an error return code of 8 or more.

Programmer response

REG15 in the REG0-REG15 save area contains the return code. CMDQNAME in MRQWORK is the queue name being processed or the last queue name successfully processed. Trace the error back to DFSQMR20.

X'A008'   QMR50 - UNLOAD - EXCEEDS ECSA HIGH WATER MARK

Explanation

During an unload processing of shared (GLOBAL) queues, the extended common service area (ECSA) storage reached the high water mark.

To avoid an IMS control region abend, the QCF unload job stops and issues the A008 code when the ECSA storage is at 85% usage.

Programmer response

You can resubmit the unload request after the ECSA storage problem is solved. To run a QCF unload job, the ESCA buffer usage must be lower than 85%.

As a guideline, you can calculate the required ECSA storage size based on the destination with the largest number of messages, as shown in the following formula:

*(required ECSA storage) = (200 bytes) * (the number of messages)*

X'A00C'   QMR50 - UNLOAD - GU CALL ERROR

Explanation

The Get Unique (GU) call to the QMGR returned an error code.

Programmer response

REG15 in the REG0-REG15 save area contains the return code. CMDQNAME in MRQWORK is the queue name being processed. QMRWLWA in MRQWORK contains the QMGR parameter list. Trace the error to QMGR GU processing.

X'A010'   QMR50 - UNLOAD - GN CALL ERROR

Explanation

The Get Next (GN) call to the QMGR returned an error code.

Programmer response

REG15 in the REG0-REG15 save area contains the return code. CMDQNAME in MRQWORK is the queue name being processed. QMRWLWA in MRQWORK contains the QMGR parameter list. REG4 is the address of the QTPPCB passed to QMGR. The first two words of the QTPPCB contain the DRRN and buffer address of the message being processed. Trace the error to QMGR GN processing.

X'A014'   QMR50 - UNLOAD - REJECT CALL ERROR

Explanation

The reject (REJ) call to QMGR returned an error code.

Programmer response

REG15 in the REG0-REG15 save area contains the return code. CMDQNAME in MRQWORK is the queue name being processed. QMRWLWA in MRQWORK contains the QMGR parameter list. REG4 is the address of the QTPPCB passed to QMGR. The first two words of the QTPPCB contain the DRRN and buffer address of the message being rejected. The message may or may not have been successfully rejected (deleted). Trace the error to QMGR reject processing.

X'A018'   QMR50 - UNLOAD - RELEASE CALL ERROR

Programmer response

REG15 in the REG0-REG15 save area contains the return code. CMDQNAME in MRQWORK is the queue name being processed. QMRWLWA in MRQWORK contains the QMGR parameter list. PSTQIMSG contains the DRRN of the message or message chain being released. Some of the messages may remain locked on the message queue (in other words, not released). Trace the error to QMGR REL processing.

X'A01C'	QMR50 - UNLOAD - INVALID CALL TYPE RECEIVED

Explanation

An invalid call was made to DFSQMR50. REG0 did not contain X'28' (CLEANUP), X'2C' (GU), X'30' (GN), or X'34' (CMD) call.

Programmer response

REG1 in the REG0-REG15 save area contains the call type. Trace the problem to the caller of DFSQMR50.

X'A020'	QMR50 - UNLOAD - INVALID CALL SEQUENCE

Explanation

Invalid sequence of calls to DFSQMR50. The error is set if a GN call is issued for the next message buffer and no message is being processed (GU).

Programmer response

REG1 in the REG0-REG15 contains the call type. REG6 or MRQUNLMC in MRQWORK points to the unload buffer. The first two bytes are zero if no message is in progress. The prior message returned on unload may still be in the buffer. The problem may be that the last message contained incorrect first/last flags or an error in logic between QCF and IMS.

X'A024'	QMR50 - UNLOAD - SELECT MESSAGE ERROR

Explanation

The call to DFSQMR20 to select a message returned an error code of 8 or higher.

Programmer response

REG15 in the REG0-REG15 save area contains the return code. CMDQNAME in MRQWORK is the queue name being processed. MRQUNLMC in MRQWORK points to the unload message being selected. Trace the problem to select routine DFSQMR20.

X'A028'	QMR50 - UNLOAD - SYSTEM NOT SHARE QUEUES

Explanation

Unload could not process the request because the shared queue is not active.

Programmer response

The function is valid only in a shared-queues environment.

X'A02C'	QMR50 - UNLOAD - CMD QUEUE TYPE INVALID

Explanation

An invalid or no queue type was passed on the unload command call from the QCF client.

Programmer response

MQTYPQUE in MRQCMDWK contains either zero or an invalid queue types. The cold queue is an invalid unload queue type. REG8 in the REG0-REG15 save area contains the MRQCMDWK address, which contains the unload command from the QCF client.

X'A030'	QMR50 - UNLOAD - QUEUENAME INVALID

Explanation

An invalid queue name was detected by the queue manager.

Programmer response

This is an internal IMS error.

X'A034'	QMR50 - UNLOAD - DEST IS INVALID/WAITERS ON QUEUE

Explanation

An invalid destination address was detected while processing an UNLOAD request for the local queues.

This invalid destination might be due to the existence of waiters on the QSN WAIT queue.

Programmer response

UCURLNAM in MRQCMDWK contains either a zero or an invalid destination address.

X'A038'	QMR50 - UNLOAD - CONFLICT BETWEEN QDFLG1 AND QDQCBDQ

Explanation

A conflict occurred between QDFLG1 and QDQCBDQ while processing an UNLOAD request for the local queues. QDFLG1 indicated that messages were on the destination, but QDQCBDQ did not point to a DRRN of a Queue Block or a message.

Programmer response

UMRQQBLK in MRQCMDWK contains the work area that detected this condition. This is an internal IMS error.

X'A03C'	QMR50 - UNLOAD - 1ST RECORD RETURNED NOT 1ST OF MESSAGEQ

Explanation

A message is being retrieved from the local queues, and message flag (MSGFFRST) is not set on.

Programmer response

Locate the message flags in the message prefix. If the message is a first buffer, then MSGFFRST should be set. If not, this is an internal IMS error.

X'A040'	QMR50 - UNLOAD - MESSAGE CHAIN IS BROKEN

Explanation

A message is being retrieved from the local queues, and the chain of messages is broken.

Programmer response

This is an internal IMS error.

X'A044'	QMR50 - UNLOAD - ERROR GET/REL DFSBCB STORAGE

Explanation

A DFSBCB call received a nonzero return code while trying to get or release storage from the storage pool for a work area.

Programmer response

R15 is the return code from the DFSBCB call. This is either an internal IMS error or insufficient storage is available in the control region private area.

X'A048'	QMR50 - UNLOAD - QDQCBDQ DOES NOT POINT TO A QUEUE BLOCK

Explanation

A message is being retrieved from the local queues, and the QDQCBDQ field does not point to a Queue Block.

Programmer response

This is an internal IMS error.

X'A04C'	QMR50 - UNLOAD LOCAL QUEUES CONTINUATION TYPE ERROR, PRIOR UNLOAD CALL WAS IN ERROR.

Explanation

A message is being unloaded from the local queues; the prior unload request terminated with an error.

Programmer response

This is an internal IMS error.

X'A050'	QMR50 - UNLOAD LOCAL QUEUES CONTINUATION TYPE REQUEST, THE SMB SUSPEND QUEUE WAS DRAINED DURING THE PROCESS OF BEING UNLOADED.

Explanation

A message is being unloaded from the local queues SMB suspend queue. The queue was drained during the unload request. The queue should be empty. No action is required.

Programmer response

This message is issued for your information.

X'A054'	QMR50 - UNLOAD LOCAL QUEUES CONTINUATION TYPE REQUEST, THE SMB SUSPEND QUEUE WAS MODIFIED DURING THE PROCESS OF BEING UNLOADED.

Explanation

A message is being unloaded from the local queues SMB suspend queue. The queue was modified during the unload request. The unload request must be resubmitted if the SMB suspend queue is to be unloaded.

Programmer response

This warning message is issued for your information.

X'A058'   **QMR50 - UNLOAD LOCAL QUEUES CONTINUATION TYPE REQUEST, THE CNT QUEUE WAS DRAINED DURING THE PROCESS OF BEING UNLOADED.**

Explanation

A message is being unloaded from the local queues CNT queue. The queue was drained during the unload request. The queue should be empty. No action is required.

Programmer response

This warning message is issued for your information.

X'A05C'   **QMR50 - UNLOAD LOCAL QUEUES CONTINUATION TYPE REQUEST, THE CNT DEQUEUE POINTER WAS MODIFIED DURING THE PROCESS OF BEING UNLOADED.**

Explanation

A message is being unloaded from the local queues SMB queue. The queue was drained during the unload request. The queue should be empty. No action is required.

Programmer response

This warning message is issued for your information.

X'A060'   **QMR50 - UNLOAD LOCAL QUEUES CONTINUATION TYPE REQUEST, THE SMB QUEUE WAS DRAINED DURING THE PROCESS OF BEING UNLOADED.**

Explanation

A message is being unloaded from the local queues SMB queue. The queue was drained during the unload request. The queue should be empty. No action is required.

Programmer response

This warning message is issued for your information.

X'A064'   **QMR50 - UNLOAD LOCAL QUEUES CONTINUATION TYPE REQUEST, THE SMB DEQUEUE POINTER WAS MODIFIED DURING THE PROCESS OF BEING UNLOADED.**

Explanation

A message is being unloaded from the local queues SMB queue. The queue was modified during the unload request. The unload request must be resubmitted if the SMB queue is to be unloaded.

Programmer response

This warning message is issued for your information.

X'A068'   **QMR50 - UNLOAD LOCAL QUEUES, REQUESTED DESTINATION IS BEING READ BY ANOTHER TASK.**

Explanation

A message is being unloaded from the local queues CNT queue. The queue is currently being read by another task.

The unload of this local queue is terminated and the next queue is then unloaded.

The unload request must be resubmitted if the CNT queue is to be unloaded.

Programmer response

This warning message is issued for your information. The unload of the other local queues is successful.

X'A06C'   **QMR50 - ERROR TERMINATING IMS CONVERSATION**

Explanation

While unloading (deleting) a message associated with an IMS conversational transaction, an error was encountered during termination of the conversation. The message was deleted, but the conversation may not have been terminated.

Programmer response

This is probably an IMS internal error. Get the 6701-MRQE log record with the AIBREASN=0000A06C error. Fields QCFDIAG1 and QCFDIAG2 will indicate the type of error detected in the conversation termination routine (DFSCON20).

X'A070'   **QMR50 - COMMAND RESPONSE MESSAGE CAN'T BE DELETED**

Explanation

The message being unloaded (deleted) is an AOI command response message for the active application program that issued the AOI command

(flag MSGFPADL=MSGSACMD is set). This type of message cannot be deleted.

Programmer response

This is a normal condition. The message was not unloaded (deleted) by QCF. The message will be deleted by IMS when the AOI application program finishes processing the response, reaches a sync point, or is terminated. IMS logs a type 6701-MRQE record for this condition and then skips to the next destination. The message can be found in the 6701-MRQE record, in the buffer labeled UMRQMSG.

X'A074'	QMR50 - BUFFER USAGE EXCEEDS MAX

Explanation:
One or more failing destinations caused a buffer overflow because the QBUFMAX value has been exceeded.

Programmer response:
Retry the failing destination.

X'A078'	QMR50 - SHARED QUEUE MSG IS BEING READ.

Explanation

A message is being unloaded from the shared queues. The queue is currently being read by another IMS system.

The unload of this shared queue is terminated and the next queue is then unloaded.

The unload request must be resubmitted if this queue is to be unloaded.

Programmer response

This warning message is issued for your information. The unload of the other shared queues is successful.

X'B004'	QMR40 - RECOVER COMMAND ERROR

Explanation

The RECOVER command was issued in a nonshared-queues environment.

Programmer response

Make the appropriate change.

X'B008'	QMR40 - RECOVER COMMAND ERROR

Explanation

An invalid call was made to DFSQMR40. Register 0 did not contain X'28' (MRQCLEAN), X'34' (MRQCMD), or X'38' (MRQGCMD).

Programmer response

Correct the problem, and reissue the command.

X'B00C'	QMR40 - RECOVER COMMAND ERROR

Explanation

The CQS BROWSE of the cold queue returned partial data. The size of the data object was larger than 32K.

Programmer response

Locate the message and determine its origin. REG8 in the REG0-REG15 save area contains the message address.

X'B010'	QMR40 - RECOVER COMMAND ERROR

Explanation

The CQS BROWSE of the cold queue failed.

Programmer response

This is an internal error. Locate the CQS reason code (CQSRSNCD) in the parameter list (QMRWLWA) to determine the cause. If reason code = NO DATA OBJECTS RETURNED, dump the cold queue to verify whether any message exists. If so, this is an internal error.

X'B014'	QMR40 - RECOVER COMMAND ERROR

Explanation

DFSQMR20 returned with RC = X'08' for message selection.

Programmer response

Trace back to DFSQMR20 to determine the cause.

X'B018'	QMR40 - RECOVER COMMAND ERROR

Explanation

Subroutine MR4DELET detected that the message segment from the cold queue was not the first segment.

Programmer response

Locate the message and verify the contents. REG8 in the REG0-REG15 save area contained the message address. This is an internal IMS error.

X'B01C'	QMR40 - RECOVER COMMAND ERROR

Explanation

Subroutine MR4DELET detected that the TMR prefix segment could not be located.

Programmer response

Locate the message and verify that the prefix segment exists. The TMR prefix segment code is X'8C'. REG8 in the REG0-REG15 save area contained the message address. This is an internal IMS error.

X'B020'	QMR40 - RECOVER COMMAND ERROR

Explanation

CQSRECVR FUNC=DELETE of a message from the cold queue failed.

Programmer response

Internal error. Locate the CQS reason code (CQSRSNCD) in the parameter list (QMRWLWA) to determine the cause.

X'B024'	QMR40 - RECOVER COMMAND ERROR

Explanation

The CQS DELETE of a message from the staging queue failed.

Programmer response

Internal error. Locate the CQS reason code (CQSRSNCD) in the parameter list (QMRWLWA) to determine the cause.

X'B028'	QMR40 - RECOVER COMMAND ERROR

Explanation

Subroutine MR4UNLCK detected that the message segment from the cold queue was not the first segment.

Programmer response

Locate the message and verify the contents. REG8 in the REG0-REG15 save area contained the message address. This is an internal IMS error.

X'B02C'	QMR40 - RECOVER COMMAND ERROR

Explanation

The CQSRECVR FUNC=UNLOCK of a message from the cold queue failed.

Programmer response

Internal error. Locate the CQS reason code (CQSRSNCD) in the parameter list (QMRWLWA) to determine the cause.

X'B030'	QMR40 - RECOVER COMMAND ERROR

Explanation

Subroutine MR4READ detected that the message segment from the cold queue was not the first segment.

Programmer response

Locate the message and verify the contents. REG8 in the REG0-REG15 save area contained the message address. This is an internal IMS error.

X'B034'	QMR40 - RECOVER COMMAND ERROR

Explanation

Subroutine MR4READ detected that the TMR prefix segment cannot be located.

Programmer response

Locate the message and verify that the prefix segment exists. The TMR prefix segment code is X'8C'. REG8 in the REG0-REG15 save area contained the message address. This is an internal IMS error.

X'B038'	QMR40 - RECOVER COMMAND ERROR

Explanation

The CQS BROWSE of the staging queue returned partial data. The size of the data object was larger than 32K.

Programmer response

Locate the message and determine its origin. REG8 in the REG0-REG15 save area contained the message address.

X'B03C'	QMR40 - RECOVER COMMAND ERROR

Explanation

No more data objects for QNAME were found during the CQS BROWSE of the staging queue for a multi-buffer message.

Programmer response

The message may have been deleted by another requester while being browsed. All previously returned segments of this message should be discarded.

X'B040'	QMR40 - RECOVER COMMAND ERROR

Explanation

The CQS BROWSE of the staging queue failed.

Programmer response

Internal error. Locate the CQS reason code (CQSRSNCD) in the parameter list (QMRWLWA) to determine the cause.

X'B044'	QMR40 - RECOVER COMMAND ERROR

Explanation

Subroutine MR4READ detected that the message segment from the staging queue was not a middle segment.

Programmer response

Locate the message and verify the contents. REG8 in the REG0-REG15 save area contained the message address. This is an internal IMS error.

X'B048'	QMR40 - RECOVER COMMAND ERROR

Explanation

Subroutine MR4READ detected that the message segment from the staging queue was not the last segment.

Programmer response

Locate the message and verify the contents. REG8 in the REG0-REG15 save area contained the message address. This is an internal IMS error.

X'B04C'	QMR40 - RECOVER COMMAND ERROR

Explanation

The request was terminated because RESYNC was not done between IMS and CQS.

Programmer response

Reissue the request after RESYNC is done.

X'C000'	QMR20 - SELECT SHOULD NOT OCCUR ERROR

Explanation

The selection criteria routine (DFSQMR20) was called, and it detected that include/exclude processing was to be performed; however, the routine did not find any rows (INCL/EXCL) to process.

Programmer response

The error is detected at label MRSEL500 in DFSQMR20. The Select routine reached this routine, and the INCL/EXCL register (R6) was zero. R6 should be the address of one of the INCL/EXCL rows mapped by DFSMRQCT. R7 is the select work area (DFSMRQSW) that anchors the rows. MRINCTTR in DFSMRQSW is the number of include entries. MREXCCTR is the number of exclude entries. Flag SMRQ0INC=1 in MRQWORK (R5) means the include chain is being processed; if SMRQ0INC=0, the exclude chain is being processed. This is an IMS/QCF internal error.

X'C004'	SELECT CRITERIA DFSSQQRY ERROR

Explanation

The selection criteria routine (DFSQMR20) was called by BROWSE, QUERY, RECOVER, or UNLOAD and issued a DFSSQQRY call to query the message queues (shared-queues environment) to determine which queues have messages. DFSSQQRY returned a nonzero return code.

Programmer response

REG1 in the REG0-15 save area and the CMDCQSRC field in MRQWORK (macro DFSMRQWK) contain the

DFSSQQRY return code. REG4 is the address of the parameter list passed to DFSSQQRY. DFSSQQRY return codes are also listed in the DFSSQQRY macro. The current codes follow:

```
SQQRRC_OK          EQU  0   CALL SUCCESSFUL
SQQRRC_SOME        EQU  4   SUCCESSFUL ONLY FOR
                             SOME RESOURCES
SQQRRC_NONE        EQU  8   SUCCESSFUL FOR NO
                             RESOURCES
SQQRRC_CQS_NOT_AVAIL EQU 12 CQS IS NOT AVAILABLE
SQQRRC_IMS_STG_ERR EQU 16   IMS STORAGE ERROR
SQQRRC_CQS_ERR     EQU 20   CALL UNSUCCESSFUL -
                             CQS ERROR
SQQRRC_IMS_ERR     EQU 24   CALL UNSUCCESSFUL -
                             IMS ERROR
SQQRRC_QTP_NOMSG   EQU 28   NO MSGS FOR QTYPE
```

This is probably a CQS error.

X'C008'	QMR20 - SELECT CRITERIA DFSPOOL ERROR

Explanation

The selection criteria routine (DFSQMR20) was called by either BROWSE, QUERY, RECOVER, or UNLOAD and issued a DFSPOOL call to free storage; DFSQMR20 received a nonzero return code.

Programmer response

REG15 in the REG0-15 save area contains the DFSPOOL return code. REG3 is the address of the storage being freed, and REG4 is the address of the parameter list passed to DFSPOOL. The storage being freed is the storage occupied by the DFSSQQRY buffer obtained on a DFSSQQRY call. This is an IMS internal error.

X'C00C'	QMR20/QMRA0 - INVALID CMD CALL

Explanation

DFSQMR20 or DFSQMRA0 was called with an invalid command call.

Programmer response

Check QMRWFLG0 in DFSMRQWK for the command in progress, which should be BROWSE, QUERY, RECOVER, or UNLOAD.

X'C010'	QMRA0 - DFSCBTS SCAN/FIND ERROR

Explanation

The DFSCBTS call resulted in a return code greater than 4.

Programmer response

This is an internal IMS error.

X'D004'	QMR70 - LOAD/ - INVALID CALL TYPE REC

Explanation

An invalid call was made to DFSQMR70. REG0 did not contain either X'04' (LOAD), X'08' (PURG), X'0C' (CANCEL), X'1C' (REROUTE), X'24' (REROUTE PURG), X'28' (CLEANUP), or X'34' (COMAND).

Programmer response

REG1 in the REG0-REG15 save area contains the call type. Trace the problem to the caller of DFSQMR70. The function being processed should be a load and the call should be one of the above.

X'D008'	QMR70 - LOAD - ERROR CANCELING MESSAGE

Explanation

At cleanup time when the QCF client ended, the load/ routine detected a message or partial message that had not been enqueued; the load/ routine tried to cancel it. A nonzero return code was returned on the QMGR cancel call. Because the BMP was ending, this AIBREASN code was not returned to the QCF client.

Programmer response

REG15 in the REG0-REG15 save area contains the QMGR return code from the cancel. REG2 contains the QTPPCB used for the cancel call. The nonzero cancel return code is an MRQ/QMGR internal error. The QCF client terminating with an message in progress may be a QCF client internal error.

X'D00C'	QMR70 - XFER - ERROR TRANSFERING MESSAGE

Explanation

While processing a Reset or Cleanup command from the QCF client, a Transfer (XFER) call was issued to transfer inserted messages from the temporary to the permanent destination. A nonzero return code was returned on the XFER call.

Programmer response

REG15 in the REG0-REG15 save area contains the QMGR return code from the XFER. REG2 contains the QTPPCB used for the XFER call. The nonzero XFER return code is a QCF or QMGR internal error.

X'E000'   **DFSQMR00 - QSN EXIT STARTED TOO MANY BMPs**

Explanation

The QCF QSN exit routine (DFSQMRI0) started more than one concurrently-executing client.

Programmer response

QSN exit routine DFSQMRI0 detected that the message queue threshold was reached and started a client to process the message queue. However, a previously-started client had not completed. This AIBREASN code is displayed to the client processing the lowest message queue threshold (threshold A to B, B to C, or C to D) and this client is terminated. To eliminate message queue thrashing, only the client processing the highest threshold is allowed to run. This condition is not considered an error unless the client processing the lower threshold (the client that receives this code) is stalled. Check to ensure that the client that was scheduled for a higher threshold completes successfully. R5 is the address of MRQWORK; MRQWORK flag QMRWFL02 = QM2BMPAB, QM2BMPBC, or QM2BMPCD indicates which threshold-exceeded condition this client was processing.

X'E004'   **DFSQMRD0 - QC/QSN INVALID CMD CALL**

Explanation

DFSQMRD0 was called with an invalid command call.

Programmer response

Check QMRWFL00 in DFSMRQWK for the command in progress. It should be QC-ABE, QC-REL, QC-SND, or QC-SUS. This is an internal IMS or QCF error.

X'E008'   **DFSQMRD0 - QC/QSN SUPPORTED ONLY IN QCF ENVIRONMENT**

Explanation

DFSQMRD0 was called in a nonQCF environment.

Programmer response

This is an internal IMS or QCF error.

X'E00C'   **DFSQMRD0 - QC/QSN CMD CALL NO QQSN BLOCK**

Explanation

DFSQMRD0 was called with a QC/QSN CMD call but no QQSN block.

Programmer response

This is an internal IMS or QCF error.

X'E010'   **DFSQMRD0 - QC/QSN COMMAND ACTION INVALID**

Explanation

QC/QSN command action is invalid.

Programmer response

Check QCMRQFL0 in DFSMRQWK for the command in progress. It should be QC-ABE, QC-REL, QC-SND, or QC-SUS. This is an IMS or QCF error.

X'E014'   **DFSQMRD0 - QC/QSN CMD CALL INVALID ITASK**

Explanation

QC/QSN command call ITASK type is invalid.

Programmer response

Check MRPTASK1 and MRPTASK2 in the DFSMRQPF macro for the valid ITASK types. This is an internal IMS or QCF error.

X'E018'   **DFSQMRD0 - ERROR GET/REL AN AWE**

Explanation

DFSBCB GET or REL for an AWE block received a nonzero return code. R15 is a return code from a DFBCB GET call.

Programmer response

This is either an error or insufficient storage is available in the control region private area.

X'E01C'   **DFSQMRD0 - QC/QSN CMD CALL RECEIVED ERROR CODE FROM SELECT**

Explanation

DFSQMRA0 returned an RC= X'08' in the QQSN selection call.

Programmer response

Trace back to DFSQMRA0 to determine the cause.

X'E020'	QMRG0 - INVALID FUNCTION PASSED TO QC LOAD AND QUERY QUEUE SPACE NOTIFICATION TABLE

Explanation

An invalid call was made to DFSQMRG0. Register 0 did not contain X'28' (MRQCLEAN), X'34' (MRQCMD), or X'38' (MRQGCMD).

Programmer response

The REG0-REG15 save area contains the function code. REG14 is the address of the caller of DFSQMRG0 (BALR REG). Trace the call back to the caller of DFSQMRG0.

X'E024'	DFSQMRG0 - QC LOAD CMD CALL RECEIVED ERROR GET/REL DFSPOOL STORAGE SERVICES

Explanation

The DFSPOOL call RECEIVED a nonzero return code while trying to get or release storage from the HIOP STORAGE POOL for a DFSMRQTB work area.

Programmer response

R15 contains the return code from the DFSPOOL call. This is either an internal error or insufficient storage is available in the control region private area.

X'E028'	DFSQMRG0 - QC LOAD CMD PROCESSING - THE VALUE FOR QUOTNOTF IS INVALID

Explanation

The percentage for QUOFNOTF passed to IMS on a QC LOAD CMD call was invalid.

Programmer response

This is either an internal QCF or IMS error.

X'E02C'	DFSQMRG0 - INVALID CMD CALL RECEIVED, ONLY /QC-LTBL IS CURRENTLY SUPPORTED

Explanation

An invalid call was made to DFSQMRG0. The flag QCMRQFL0 in DFSMRQWK did not indicate that the CMD was /QC-LTBL.

Programmer response

Check flag QCMRQFL0 in DFSMRQWK for the command in progress. It should be QC-LTBL. This is an IMS or QCF internal error.

X'E030'	DFSQMRG0 - INVALID QUEUE UPPER AND/OR LOWER THRESHOLD PERCENT

Explanation

Invalid queue upper and/or lower threshold percents were detected. The DFSMRQTB work table was constructed using the default upper (75%) and lower (60%) values.

Programmer response

Check QUOFQTU and QUOTQTL in DFSMRQQO for the command in progress.

X'E034'	DFSQMRG0 - ERROR GET/REL AN AWE

Explanation

A DFSBCB GET or REL for an AWE block received a nonzero return code.

Programmer response

R15 contains the return code from the DFBCB GET call. This is either an internal error, or insufficient storage was available in the control region private area.

X'E038'	DFSQMRG0 - /QC-LTBL AND /QC-QTBL NOT ACTIVE

Explanation

The required function is not available on the current active IMS system.

Programmer response

If the function is required, module IQCQMRH0 must be linked into IMS RESLIB (replacing IMS module DFSQMRH0) or linked into a user RESLIB as DFSQMRH0. The function is not currently supported in the shared-queues environment.

X'E03C'	DFSQMRD0 - QC/QSN COMMAND IS INVALID

Explanation

The required function is not active or not supported on the current active IMS system.

Programmer response

If the function is required, module IQCQMRH0 must be linked into IMS RESLIB (replacing IMS module DFSQMRH0) or linked into a user RESLIB as DFSQMRH0. The function is not currently supported in the shared-queues environment.

X'F004'	QMRC0 - INVALID FUNCTION PASSED TO ENVIRONMENT STATISTICS ROUTINE

Explanation

The call to DFSQMRC0 was invalid. Register 0 did not contain X'28' (MRQCLEAN), X'34' (MRQCMD), or X'38' (MRQGCMD).

Programmer response

REG0 in the REG0-REG15 save area contains the function code. REG14 is the address of the caller of DFSQMRC0 (BALR REG). Trace the call back to the caller of DFSQMRC0.

X'F008'	QMRC0 - IMS IS IN THE PROCESS OF SHUTDOWN OR QUIESCING.

Explanation

IMS is in the process of shutting down or quiescing, and the CQS query command is not allowed at this time.

Programmer response

This is an IMS information AIBREASN code.

X'F00C'	QMRC0 - SHARED QUEUES ENVIRONMENT, NO SHARED QUEUES MASTER CONTROL BLOCK (SCDSQM).

Explanation

IMS is running in a shared-queues environment. The pointer to the shared queues MASTER CONTROL BLOCK is zero.

Programmer response

This is an internal IMS error.

X'F010'	QMRC0 - SHARED QUEUES ENVIRONMENT, NO STRUCTURE BLOCK (SQMSQSM).

Explanation

IMS is running in a shared-queues environment, The pointer to the shared queues structure block is zero.

Programmer response

This is an internal IMS error.

X'F014'	QMRC0 - IMS INTERNAL ERROR.

Explanation

DFSSQI30 returned an unsupported return code.

Programmer response

This is an internal IMS error.

X'F018'	QMRC0 - CQS NOT AVAILABLE TO PROCESS THE CQS QUERY REQUEST.

Explanation

CQS is not available to process the CQS QUERY request.

Programmer response

This message is issued for your information.

X'F01C'	QMRC0 - CQS RETURNED AN UNSUCCESSFUL RETURN CODE FOR THE CQS QUERY REQUEST.

Explanation

CQS returned an unsuccessful return code on the CQS QUERY request. Refer to CQSRRQRY for return codes.

Programmer response

This is an internal IMS error.

X'F020'	QMRC0 - IN PROCESSING THE QCF ENVIRONMENT STATISTICS REQUEST STORAGE WAS NOT OBTAINED.

Explanation

DFSPOOL call received a nonzero return code attempting to get or release storage from the HIOP storage pool for a work area or buffer.

Programmer response

R15 has the return code from the DFSPOOL call. This is either an error, or insufficient storage is available in the control region private area.

X'F024'	**QMRC0 - IN PROCESSING THE QCF ENVIRONMENT STATISTICS THE LIST PASSED TO CQS CONTAINED AN INVALID STRUCTURE NAME.**

Explanation

The DFSSQQRY list contained an invalid CQS structure name.

Programmer response

This is an internal IMS error.

X'F028'	**QMRC0 - IN PROCESSING THE QCF ENVIRONMENT STATISTICS REQUEST THE DFSSQQRY**

RETURNED A NON-ZERO RETURN CODE.

Explanation

DFSSQQRY returned a nonzero return code.

Programmer response

This is an internal IMS error.

X'FFFF'	**MSG COUNT EXCEEDED MAX MSG COUNT *nnnn*. MESSAGES SCRAPPED BY AIB RC 00F0 BY REASON CODE FFFF**

Explanation

The LOAD function MAXMSGCT control statement value that you specified was reached. Subsequent messages are placed in the SCRAPLOG data set with a return code of X'F0'.

Programmer response

None. This is an informational message only.

Chapter 22. Return codes

This reference section provides detailed information about IMS Queue Control Facility return codes.

Return codes are reflected as follows:

- In the batch job, as the condition code
- In the batch job Report Output for those batch jobs that generate a report
- In one of the following messages:
  - IQC7003E
  - IQC7006E
  - IQC4999E

Return code	Meaning
00	Job successful. 1. Functions QUERY/BROWSE/UNLOAD/RECOVER successfully completed. AIB return and reason codes are zero. 2. Function LOAD - No request for messages to be sent to SCRAPLOG data set. 3. Function LOAD - Request for messages to be sent to SCRAPLOG data set and only requested messages sent to SCRAPLOG data set. 4. Function LOAD - Specifying certain types of messages to be loaded. They are loaded; all other messages go to SCRAPLOG data set.
04	Job successful. 1. Function LOAD - No request for messages to be sent to SCRAPLOG data set. However, messages were sent to SCRAPLOG data set. 2. Function LOAD - Specifying certain type of messages to be loaded. Other messages to go to SCRAPLOG data set. However, some messages that should be loaded go to SCRAPLOG data set. 3. Function LOAD - Unable to open SCRAPLOG data set. 4. Functions BROWSE/UNLOAD/RECOVER - Unable to open BROWSE/UNLOAD/RECOVER data sets. 5. Function RECOVER for shared queues - There are no messages on the cold queue. 6. Function BROWSE/LOAD/UNLOAD/RECOVER - Action to some destinations rejected due to insufficient authority.
08	Job fails. 1. PARSE QCFIN DD card failed. 2. Function RECOVER - Function is executed in a nonshared-queues environment. 3. Client timed out. 4. IMS address space is down after requested function has started. 5. SCI address space is down after requested function has started. 6. No RACF authorization for requested function or data sets.
12	Job fails. 1. Unable to open QCFIN or QCFPRINT data sets. 2. Function LOAD - Unable to open LOAD data set.

Return code	Meaning
16	Job fails.
	1. No SCI address space.
	2. No IMS address space.
	3. Nonzero return codes from SCI requests.

Chapter 23. Recovering from AREA and FAILSAFE actions

This section provides information on how to recover from a situation where all dependent regions and/or input devices have been waited, abended, or stopped as a result of AREA= or FAILSAFE= actions.

Actions that might cause you to begin recovery steps include the following:

- AREA=(ACTION=(WAIT | ABEND | STOP))
- FAILSAFE=(ACTION=(WAIT | ABEND | STOP))

where the ACTION keyword in the AREA control statement can be either CSTOPACTION, CSTARTACTION, OSTOPACTION, or OSTARTACTION.

The following information is a set of guidelines only. All possible steps to return an IMS system to normal processing are not covered. Rather, basic concepts are offered regarding actions to take after queue space usage has reached a critical state.

Topics:

- "Troubleshooting reference for AREA and FAILSAFE actions" on page 439
- "Using commands and TSO client for recovery" on page 445

Troubleshooting reference for AREA and FAILSAFE actions

This topic provides troubleshooting references to help you recover from actions taken by AREA and FAILSAFE statements.

First, determine the cause of the problem.

The reason could involve one or more of the following conditions:

- No dependent regions are active
- All active dependent regions were terminated
- Not enough dependent regions were active from the start, or some were terminated
- Transaction is in a loop, sending messages to output device or alternate destination (LTERM or transaction)
- Output devices, such as output printer, are not connected to IMS
- Intelligent device is in a loop, sending messages to IMS
- Programs, not transactions, are in an IMS stopped state

Although other reasons could exist, the list above identifies the main reasons for the system to require recovery because of AREA and/or FAILSAFE statement actions.

Your response would depend on what conditions caused the action to occur.

Table 19. Recovery actions

Condition	Action
If any of the following conditions exists: • No dependent regions are active, or • All active dependent regions were terminated, or • Not enough dependent regions were active from the start, or some were terminated...	The best action would be: 1. Submit an IMS Queue Control Facility batch UNLOAD job to unload some messages from the queue. Unload the messages which are most likely to relieve the queue space problem. If you start dependent regions before reducing the number of messages on the message queue, it could result in a transaction removing only one message from the queue. This might temporarily resolve the space problem; however, the transaction may then require more queue buffers to ISRT messages back to the queue than were freed from the queue with the IMS DLI call of GU. The ACTION specified on the AREA/FAILSAFE statement would be taken on this dependent region, resulting in a reoccurrence of the same condition. 2. If **ACTION=WAIT** was specified, and there were no dependent regions active at the time of the AREA/FAILSAFE action, then only input devices might appear on the WAIT list. You should start the dependent regions, and then RESUME those devices. If there were not enough dependent regions running, there could be transactions in the WAIT list as well as input devices. You should start the dependent regions; then RESUME the transactions first, and, finally, RESUME the input devices. 3. If **ACTION=ABEND** was specified, use the IMS **/START** command to start the transactions/programs which may have been stopped as part of the ABEND process. Input devices will not have been ended abnormally but, instead, **ACTION=STOP** would have been taken by IMS Queue Control Facility. 4. If **ACTION=STOP** was specified, start the dependent regions. No additional action is required. **ACTION=STOP** is a request to notify the IMS transaction to stop sending messages by sending an A7 PCB status code. The status code sends two signals to the transaction: (1) The ISRT request was rejected; that is, the message was not inserted in the message queue. (2) The transaction should take an error path to terminate, and discontinue inserting (ISRT) messages to the message queue. 5. If any messages were unloaded in step "2" on page 440, and the queue space problem was resolved, reload those messages that were unloaded in step "1" on page 440.

Table 19. Recovery actions (continued)

Condition	Action
If the transaction is in a loop, sending messages to the inputting device or alternate device...	The best action would be:

The best action would be:

1. Issue an IMS **/STOP TRAN** *tran_name* command for the IMS transaction that is looping. This will ensure that the condition does not reoccur. Later, you should analyze the transaction, and correct the problem before running the transaction again.

2. After the transaction stops and if the transaction is on the WAIT list, then use the TSO client to show the WAIT list. Locate the looping transaction, and request that it be ended abnormally. This will free all uncommitted messages placed on the queue by the looping transaction. This could solve the problem.

3. If the messages were committed by the looping transaction, it is very important to determine to which message destination the transaction was looping. The queues that represent the looping message destination must be unloaded and analysis must be performed to determine which messages are valid and which are the messages on which the application was looping.

4. For assistance in determining where the messages were being sent, analyze the displayed IMS Queue Control Facility source/destination error messages.

5. If you can determine the destination, then submit a batch UNLOAD job to unload messages placed on the queue by this transaction. Analyze these messages later to determine if any should be returned to the message queue.

After messages are unloaded, then:

1. If **ACTION=ABEND** was specified, use the IMS **/START** command to start the transactions/programs which were stopped. Input devices will not have been ended abnormally but, instead, **ACTION=STOP** would have been taken by IMS Queue Control Facility.

2. If **ACTION=STOP** was specified, no additional action is required.

continued...

Table 19. Recovery actions (continued)

Condition	Action

	If you cannot determine the destination of the messages for the looping transaction, you must decide which message queue destinations to unload. It can be useful to unload all messages until you can determine which ones are associated with the looping transaction.
	
	1. If you decide to unload a portion of the destination messages (those which you know are not from the looping transaction), then submit an IMS Queue Control Facility batch UNLOAD job to unload the selected messages from the message queue. This may allow you to QUERY the rest of the queue.
	
	2. Next, use the TSO client to QUERY the message queue to assist in determining which messages are duplicates from the looping transaction.
	
	3. After identifying duplicate messages, you should submit a batch UNLOAD job to unload the duplicates, and then determine if the initial UNLOAD (step 7) should be reloaded prior to taking any other action.
	
	4. After concluding that the remaining messages on the queue are valid and all messages inserted in error by the looping transaction have been removed, then:
	
	  a. If **ACTION=WAIT** was specified, use the TSO client to show the WAIT list and begin to RESUME those transactions and LTERMs which are waiting selectively, or with the ALL option.
	
	  b. If **ACTION=ABEND** was specified, use the IMS **/START** command to start the transactions/programs that were stopped. Input devices will not have been ended abnormally but, instead, **ACTION=STOP** would have been taken by IMS Queue Control Facility.
	
	  c. If **ACTION=STOP** was specified, no additional action is required.

If output devices, such as an output printer, are not connected to IMS...	The best action would be:
	
	1. Perform installation steps to connect/start the output device in a way that ensures that it can request retrieval of messages
	
	2. Depending on the action required by this output device, it may be necessary to unload some messages with a batch Unload job first. After the output device starts removing messages from the queue, the condition should be resolved.
	
	3. If you removed messages from the queue, and the condition caused by the AREA/FAILSAFE statement has been resolved, then unloaded messages can be reloaded.
	
	  a. Next, if **ACTION=WAIT** was specified, use the TSO client to show the WAIT list and begin to RESUME those transactions and LTERMs which are waiting selectively, or with the ALL option.
	
	  b. If **ACTION=ABEND** was specified, use the IMS **/START** command to start the transactions/programs which were stopped. Input devices will not have been ended abnormally but, instead, **ACTION=STOP** would have been taken by IMS Queue Control Facility.
	
	  c. If **ACTION=STOP** was specified, no additional action is required.

Table 19. Recovery actions (continued)

Condition	Action
If an intelligent device is in a loop, sending messages to IMS...	The best action would be:

1. First, issue an IMS **/STOP LTERM** *lterm_name* command to ensure that the condition does not reoccur once the AREA/FAILSAFE condition has been resolved.

2. It is important to determine to which message destination the transaction was looping. The queues that represent the looping message destination must be unloaded and analysis must be performed to determine which messages are valid and which are the messages on which the input device application was looping.

3. Analyze the IMS Queue Control Facility source/destination error messages which have been displayed by IMS Queue Control Facility to help determine where messages were being sent. This analysis will help determine which messages, if any, should later be returned to the message queue.

4. After messages created by the looping condition are unloaded, then:

   a. If **ACTION=WAIT** was specified, use the TSO client to show the WAIT list and begin to RESUME those transactions and LTERMs which are waiting selectively, or with the ALL option.

   b. If **ACTION=ABEND** was specified, use the IMS **/START** command to start the transactions/programs which were stopped. Input devices will not have been ended abnormally but, instead, **ACTION=STOP** would have been taken by IMS Queue Control Facility.

   c. If **ACTION=STOP** was specified, no additional action is required.

5. Before allowing the application to run again, analyze the intelligent input device application, and correct the problem.

continued...

Table 19. Recovery actions (continued)

Condition	Action
	If you cannot determine the destination of the messages in the looping input device, you must decide which message queue destinations to unload. It can be useful to unload all messages until you can determine which ones are associated with the looping transaction.

1. If you decide to unload a portion of the destination messages (those which you know are not from the looping device), then submit an IMS Queue Control Facility batch UNLOAD job to unload the selected messages from the message queue. This may allow you to QUERY the rest of the queue.

2. Next, use the TSO client to QUERY the message queue to assist in determining which messages are duplicates from the looping input device.

3. After identifying the duplicate messages, you should submit a batch UNLOAD job to unload the duplicates, and then determine if the initial UNLOAD (step 1) should be reloaded prior to taking any other action.

4. After concluding that the remaining messages on the queue are valid and all messages inserted in error by the looping input device have been removed, then:

   a. If **ACTION=WAIT** was specified, use the TSO client to show the WAIT list. Locate the looping intelligent input device, and request that it be stopped. This will cause the system to send a DFS error message to the device. If the application on the device has been designed to terminate on receipt of an error message, the input device will stop.

   b. If **ACTION=ABEND** was specified, use the IMS **/START** command to start the transactions/programs which were stopped. Input devices will not have been ended abnormally but, instead, **ACTION=STOP** would have been taken by IMS Queue Control Facility.

   c. If **ACTION=STOP** was specified, no additional action is required.

Table 19. Recovery actions (continued)

Condition	Action
If programs (not transactions) are in an IMS stopped state...	The best action would be:

The best action would be:

1. Submit a batch UNLOAD job to unload messages from the queue. These should be messages which are likely to relieve the queue space problem. Starting the stopped programs before reducing the number of messages could result in a transaction removing one message from the queue. This might temporarily resolve the space problem; however, the transaction may then require more queue buffers to ISRT messages back to the queue than were freed from the queue with the IMS DLI call of GU. The ACTION specified on the AREA/FAILSAFE statement would be taken on this dependent region, resulting in a recurrence of the same condition.

2. If **ACTION=WAIT** was specified then, using the TSO client, display the WAIT list and begin to RESUME those transactions and LTERMs which are waiting selectively, or with the ALL option.

3. If **ACTION=ABEND** was specified, use the IMS **/START** command to start the transactions/programs which were stopped. Input devices will not have been ended abnormally but, instead, **ACTION=STOP** would have been taken by IMS Queue Control Facility.

4. If **ACTION=STOP** was specified, no additional action is required.

5. After the queue space problem is resolved, you should reload the messages that were unloaded in step 1.

Using commands and TSO client for recovery

You can usually use the TSO client and a batch UNLOAD job to resolve problems related to AREA/FAILSAFE ACTION=WAIT statements. However, occasionally you may need to use IMS commands.

If ACTION=WAIT has been issued, and the dependent region is waited, you can use the TSO client to access the wait list. Then, you can choose one of the following:

- ABEND - causes the dependent region to be terminated. This might resolve the problem.
- RESUME - causes the dependent region to be POSTed, and processing resumes. However, if the problem which caused the dependent region to be WAITed has not been corrected, the dependent region can be WAITed again.

Using the TSO client or UNLOAD command for recovery

You can use the TSO client to change configuration options or to view the wait list and request actions against the list. You can also use MVS commands to, for example, start more dependent regions.

You can also submit a batch job to unload part or all of the messages in the message queue. This might resolve the problem. If you submit a batch job to QUERY the message queue, the batch job will be forced to the ACTION specified on the AREA/FAILSAFE statements that brought about the problem. It is not recommended that you submit any IMS Queue Control Facility batch job other than a batch UNLOAD job.

Using IMS commands

In attempting to correct the problem, you may need to use IMS commands, such as **/DISP PROGRAM** *program_name*. Use of such commands will help determine whether the program associated with one of the transaction codes that is reported to be causing the condition, is stopped. It is recommended that you not issue IMS commands which generate several lines of output, such as **/DISP ACTIVE**.

A **/START PROGRAM** *program_name* should run to normal completion and not put the MVS console (LTERM name of WROR) in a WAIT state, if WAIT was specified as the action.

Some IMS commands use Queue Manager buffers and can result in being placed in a WAIT state as well. This can be identified by the failure of the DFS996I IMS READY message to appear on the MVS console. Using the TSO client and looking at the wait list, you will be able to find the master terminal ID, WTOR, and request that it be resumed. The IMS command will complete, display a response message on the console, and reissue the WTOR to display the DFS996I message.

If the DFS996I message does not appear, recheck the TSO client wait list to determine if the WTOR is again in the wait list and, if it is, request that it be resumed. The IMS command issued and the number of IMS Queue Manager buffers it uses will dictate the number of times you must resume the WTOR from the wait list.

Chapter 24. Diagnosing requeuing problems

IMS Queue Control Facility provides several diagnostic aids to create log records of errors that are detected during processing, load processing, and during select IMS Queue Control Facility function processing.

The IMS MRQ/QCF processor module (IQCQMRQ0), that is part of the IMS Transaction Manager (TM) component, provides diagnostics for errors that can occur while you are running IMS Queue Control Facility. While problems can be diagnosed separately in IMS Queue Control Facility using SCRAPLOG records and in IQCQMRQ0 using 6701-MRQE diagnostic records, at times it might be necessary to use both of these diagnostic aids.

IMS writes a 6701 log record to the OLDS whenever an error occurs during load processing of nonshared or shared queues. IMS might also write a 6701 log record during browse, query, unload, and recover processing if errors are detected.

IMS passes the unique error code (AIBREASN code) that is associated with the error to IMS Queue Control Facility, and IMS Queue Control Facility prints the results in the Messages Scrapped report.

In addition, IMS Queue Control Facility copies the message that is in error to a data set so that you can correct the message before requeuing the message.

Module flow (nonshared queues)

The message requeuing component of IMS Queue Control Facility consists of the several modules.

IQCSELCT
Selects messages for requeuing

IQCCANCL
Analyzes and cancels messages

LOAD function
Inserts messages back to IMS for requeuing into message queue data sets

The following figure shows module flow in the nonshared queue environment.
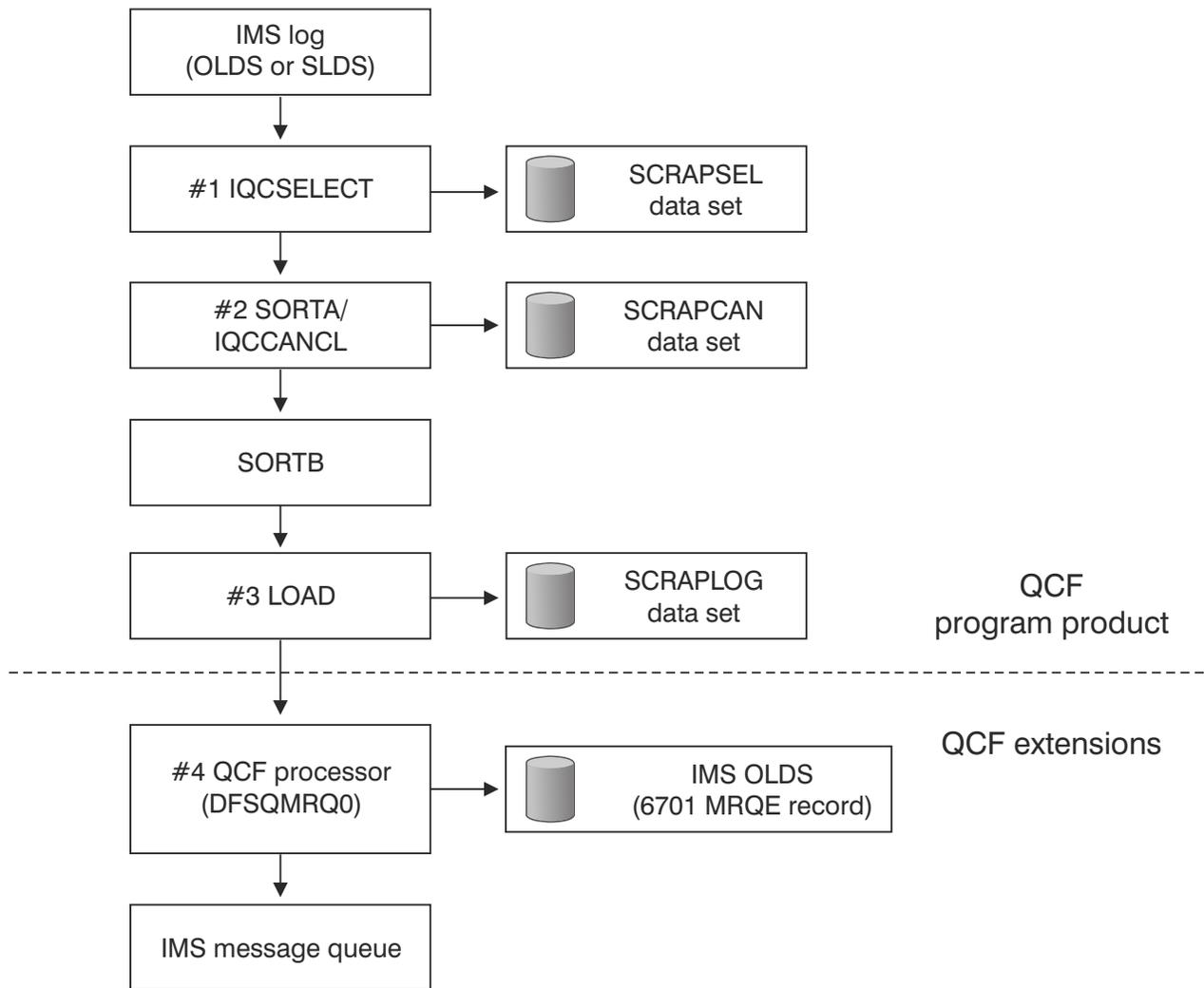
Figure 85. Relationship of QCF Program Product to QCF Extensions (nonshared queues)

IMS Queue Control Facility module IQCSELCT selects messages to be requeued from the IMS OLDS or SLDS **(#1)**. Based on the recovery mode, messages are analyzed, sorted, and collected.

Some messages might be canceled by the IQCCANCL module **(#2)**.

The messages to be read are passed to the Load function for insertion into the IMS message queues **(#3)**.

IQCSELCT, IQCCANCL, and the sort utilities run as stand-alone MVS jobs or steps.

The Load function uses an alternate modifiable teleprocessing control block (ALT TPPCB) and an application interface block (AIB) to issue ISRT calls to IMS TM to requeue the messages **(#5)**.

Module flow (shared queues)

The shared queues component of IMS Queue Control Facility is made up of several modules.

The following figure shows module flow in the shared queue environment.

The LOAD function selects messages from the input LOAD data set based on your specifications. The LOAD function runs as a QCF client (#1).

The input to the load function is the output of either the BROWSE function or UNLOAD logic.

The LOAD function passed data to IMS Queue Control Facility extensions module IQCQMRQ0. The DC call handler calls IMS Queue Control Facility processor module IQCQMRQ0 to reinsert and requeue the messages (#3).

The LOAD function uses an alternate modifiable teleprocessing control block (ALT TPPCB) and an application interface block (AIB) to issue GCMD calls to IMS TM to reinsert and requeue messages to the shared queues.
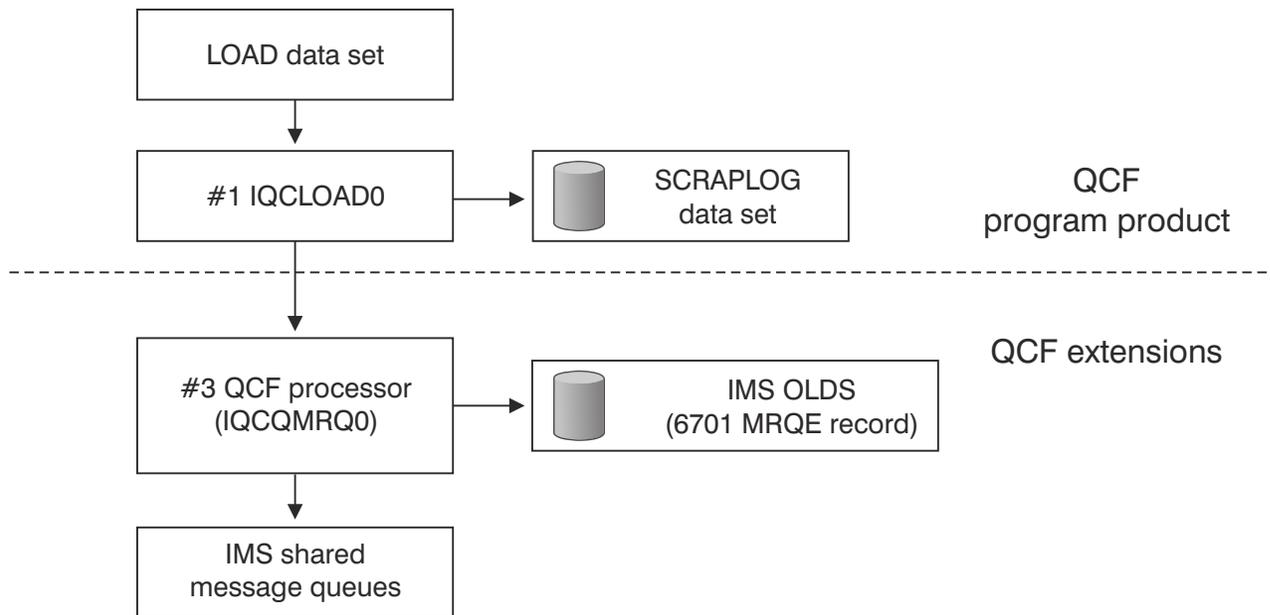
Figure 86. Relationship of QCF Program Product to QCF Extensions (shared queues)

Using diagnostics from processor module IQCQMRQ0

Use the diagnostics from processor module IQCQMRQ0 to help you isolate and solve problems with IMS Queue Control Facility.

When the IMS Queue Control Facility processor in IMS TM detects an error while reinserting a message, the following diagnostics are provided:

- The TPCBSTAT code in the QCF alternate PCB is set to MR.
- The AIB return code (AIBRETRN) is set to X'000000F0'.
- An AIB reason code (AIBREASN) is set to a unique hexadecimal value for each type of error.
- The TPPCB, AIB, I/O area (containing the message being inserted), and other pertinent control blocks are logged to the OLDS in the form of a type 6701-MRQE log record.
- The TPCBSTAT, AIBRETRN, and AIBREASN codes are passed back to the load function.
- The load function program records the error in a QCF prefix and writes the QCF prefix and the message being inserted into the SCRAPLOG data set.
- The load function routine keeps a count of messages discarded and groups them by reason code and destination.

  These groupings are shown in a QCFPRINT report when the batch load function finishes executing.

  The QCFPRINT report can be used, in combination with SCRAPLOG data sets and 6701-MRQE records logged to the IMS log data set, to analyze the error.

  When the error is corrected, it might be possible to rerun the load function program (using the SCRAPLOG data set as input) and reinsert the messages that failed.

When the Load function processor module detects an error, both SCRAPLOG records and 6701-MRQE diagnostic records are written. You need details about both of these types of records to help diagnose the problem.

Using SCRAPLOG diagnostic records

Use the SCRAPLOG diagnostic records to diagnose problems with the message requeuer.

These topics provide the following information:

- An explanation of SCRAPLOG records
- A sample SCRAPLOG record
- Information about which SCRAPLOG fields are of special interest
- Instructions for printing SCRAPLOG records

SCRAPLOG records

The details that you can identify in the SCRAPLOG records can help with diagnosing and solving IMS Queue Control Facility errors.

The SCRAPLOG record consists of a 310-byte (hexadecimal 140) QCF prefix, followed by the actual message being inserted.

The actual message is either a 4002 record (that is, a message from a DUMPQ or SNAPQ checkpoint) or a 01 (input) or 03 (output) message record.

Sample record written to SCRAPLOG by the LOAD function

The LOAD function creates a hexadecimal dump of a record to the SCRAPLOG data set for errors.

The following sample record is a hexadecimal dump of a record written to the SCRAPLOG data set by the load function routine.

The record is a 01 input record. The LOGREC type (4002, 01, or 03) is at offset X'26' (or X'27' if X'26' is X'40') in the QCF prefix segment and at offset X'104' (which is offset 4 in the scrapped record).

Note: The IMS Log File Select utility (DFSERA10) reports this record as a 5B record because offset 4 is a 5B. However, this record is not actually a log type record because the record is written to the QCF SCRAPLOG data set and not to the IMS log data set.

```
DFSERA30 - FORMATTED LOG PRINT
5B RECORD
 00000000 000000    02BE0000 5BD8C3C6  D4E2C700 06100101   01400000 00000000   00000000 00000000    *.=..$QCFMSG....................*
 00000020 000020    00000000 00000100  1999309F 01312317   8211032D E2E8E2F1   40404040 B319825D    *.........R......B...SYS1    ..B)*
 00000040 000040    2C8A80C9 E2E8E2F1  40404040 B319825D   2C8A80C9 00000000   00000000 00000000    *...ISYS1    ..B)...I............*
 00000060 000060    00010014 00000000  08000006 D3E3C5D9   D4F44040 C5E3D9C1   D5F1F940 E3C5D9D4    *...........LTERM4  ETRAN19 TERM*
 00000080 000080    F4404040 00000000  00000000 81000000   0000D4D9 000000F0   00001084 00030003    *4       .A.....MR...0...D....*
 000000A0 0000A0    00000000 00000000  00000000 00000000   00000000 00000000   D8C3C6E5 F1D9F140    *...........................QCFV1R1 *
 000000C0 0000C0    00000000 00000000  00000000 00000000   00000000 00000000   00000000 00000000    *...............................*
 000000E0 0000E0    00000000 00000000  00000000 00000000   00000000 00000000   00000000 00000000    *...............................*
 00000100 000100    00000000 00000000  00000000 00000000   00000000 00000000   00000000 00000000    *...............................*
 00000120 000120    00000000 00000000  00000000 00000000   00000000 00000000   00000000 00000000    *...............................*
 00000140 000140    016E0000 01D18110  08000006 08000006   016E8000 E2E8E2F1   40404040 B319825D    *.>...JA..........>..SYS1    ..B)*
 00000160 000160    2C8A80C9 E2E8E2F1  40404040 B319825D   2C8A80C9 00000000   00000000 00000000    *...ISYS1    ..B)...I............*
 00000180 000180    00408100 C8000000  E3C5D9D4 F4404040   00010000 00000000   00000000 00000000    *. A.H...TERM4    ...............*
 000001A0 0001A0    D3E3C5D9 D4F44040  C5E3D9C1 D5F1F940   00000000 00000000   C4C6E2D4 D6F24040    *LTERM4  ETRAN19 ........DFSM02  *
 000001C0 0001C0    00108600 00EE7C00  00000000 00000000   00148800 D3E3C5D9   D4F44040 40404040    *..F...@...........H.LTERM4      *
 000001E0 0001E0    40404040 00108900  00010000 B319825D   2C8A47C9 00188A00   1999309F 01312317    *   ..I.......B)...I.....R......*
 00000200 000200    8211032D 00000000  00000000 005C8B00   00000000 00000000   00000000 00000080    *B............*..................*
 00000220 000220    00000000 00000000  00000000 00000000   00000000 00000000   00000003 00000000    *...............................*
 00000240 000240    00000000 00000000  00000003 00030000   E2E8E2F1 40404040   B319825D 2C8A80C9    *....................SYS1    ..B)...I*
 00000260 000260    00000000 00000000  00468C00 00000003   00000000 00000001   C5E3D9C1 D5F1F940    *......................ETRAN19 *
 00000280 000280    D3E3C5D9 D4F44040  0303014C 00080000   00000000 00000000   00000000 00000000    *LTERM4  ...<...................*
 000002A0 0002A0    00000000 00000000  00000000 0610B319   825D2C8E F0C90000   00000000 018D        *.................B)..0I........ *
```

Figure 87. Sample SCRAPLOG record written by the Load function

Key fields of SCRAPLOG records and their offsets

The following table shows some key fields in the IMS Queue Control Facility records and their offsets. The current copy of this record is in the DFSMRQPF macro in IMSVS.MACLIB.

Note: All offsets are in hexadecimal, all lengths are in decimal, all values are self-defining (in other words, if they are not obviously character, then they are hexadecimal).

Table 20. Key fields of diagnostic records and their offsets

Offset	Length	Value	Description
03	1	03	Flags saying first and last segment
04	8	$QCFMSG, X'00'	Identifier saying that this is an MRQ/QCF header
0C	2	0610	IMS release level
0E	2	0101	QCF release level
26	1	01	Log code: 01 = input
27	1	00	Log subcode if 26 = 40
28	12	1999309F 01312317 8211032D	Universal time stamp
34	32	SYS1 B319825D and others	Unit of work - This is useful in tracing where the message came from, and where the message was processed
68	4	08000006	Message DRRN - This can also be used in tracing where the message came from
6C	8	LTERM4	Source CNT name
74	8	ETRAN19	Destination CNT or SMB name
7C	8	TERM4	LU name for LU 6.2, VTAM, or OTMA
84	4	00000000	Date stamp replaced by UTC at offset 28 in IMS 6.1 and later
88	4	00000000	Time stamp replaced by UTC at offset 28 in IMS 6.1 and later
8C	1	81	Destination type: 81 = SMB
92	3	MR	TPCB status: MR = IQCQMRQ0 detected an error
94	4	000000F0	AIB return code = IQCQMRQ0 detected an error
98	4	00001084	AIB reason code = unique reason code for message being discarded (scrapped), 1084 indicates that the message is nonrecoverable
9C	2	0003	Destination system ID - Two bytes for IMS 6.1 and later
9E	2	0003	Source system ID - Two bytes for IMS 6.1 and later
140	Variable	Variable	Start of the 01 or 03 log record that was scrapped, this area maps to the 6701-MRQE I/O AREA, starting at offset 140

Sample JCL for printing SCRAPLOG records

You can submit JCL to print SCRAPLOG records.

Use these SCRAPLOG records to help diagnose problems with the message requeuer.

Sample JCL for printing SCRAPLOG records

The following example shows the JCL that you can use to print SCRAPLOG records:

```
//SCRAPPRT  JOB
//*  PRINT IQCSELCT SCRAPLOG
//JOBLIB   DD DISP=SHR,DSN=IMS810.RESLIB
```

```
//SELECT   EXEC PGM=DFSERA10,REGION=0K
//QCFPRINT  DD SYSOUT=A
//SYSUT1 DD DSN=QCF.SCRAPSEL,DISP=SHR
//QCFIN     DD *
CONTROL  CNTL
OPTION   PRINT E=DFSERA30
END
/*
//CANCEL   EXEC PGM=DFSERA10,COND=EVEN,REGION=0K
//*  PRINT IQCCANCL SCRAPLOG
//QCFPRINT  DD SYSOUT=A
//SYSUT1 DD DSN=QCF.SCRAPCAN,DISP=SHR
//QCFIN     DD *
CONTROL  CNTL
OPTION   PRINT E=DFSERA30
END
//LOAD  EXEC PGM=DFSERA10,COND=EVEN,REGION=0K
//*  PRINT LOAD FUNCTION SCRAPLOG
//QCFPRINT  DD SYSOUT=A
//SYSUT1 DD DSN=QCF.SCRAPLOG,DISP=SHR
//QCFIN     DD *
CONTROL  CNTL
OPTION   PRINT E=DFSERA30
END
/*
```

You need to use your SCRAPLOG records in combination with 6701-MRQE records to effectively diagnose IMS Queue Control Facility problems.

Using 6701-MRQE diagnostic records

You can print the 6701 record to help you diagnose IMS or QCF problems.

During LOAD processing, when an error results in the AIBRETRN being set to X'000000F0' and a SCRAPLOG record being written by the load function, the IMS requeuing processor writes a corresponding type 6701-MRQE diagnostic record to the OLDS.

IMS might also write a 6701 log record during browse, query, unload, and recover processing if errors are detected.

The 6701 record contains IMS control blocks and IMS Queue Control Facility input to IMS.

You can print the 6701 record to help you diagnose IMS or IMS Queue Control Facility problems.

Viewing successfully requeued messages

You can interpret log records to see successfully requeued messages.

Messages that are successfully requeued by the message requeuer are logged to the OLDS with data identical to the original log record, with the exception of the following: MSGCFLG3=MSGC3MRQ is set to indicate that this message was requeued by the message requeuer.

The offset of MSGCFLG3 is X'45', and the MSGC3MRQ flag is X'40'. This flag is propagated to other messages that originate from this message. (That is, if the message is an input transaction message, the flag is propagated to the output response messages when the transaction message is processed. Or if the message is an MSC message, the message is propagated to messages in other IMS/MSC systems when the message is sent across the MSC link.)

The following sample log record shows an input transaction message to TRANCODE=LINK13M2 from input LTERM=IMSUS01 that was requeued by the message requeuer to IMS:

```
01 RECORD
  00000000 000000    01E20000 01C48290   08000003 08000003   01B60000 C9D4E2F2   40404040 B0AAC57B   *.S...DB.............IMS2    ..E#*
  00000020 000020    AB106A02 C9D4E2F2   40404040 B0AAC57B   AB106A02 80000800   00000000 00000000   *....IMS2     .E#...............*
  00000040 000040    00408100 C8600000   C4E3E2D3 E4F2F0F1   00010000 00000000   0097239F 2248017F   *. A.H-..DTSLU201.......P...."*
  00000060 000060    C9D4E2E4 E2F0F140   D3C9D5D2 F1F3D4F2   00000000 00000000   40404040 40404040   *IMSUS01 LINK13M2........     *
  00000080 000080    00108600 01367E00   00000000 00000000   00148800 E4E2D9E3   F0F0F240 E2E8E2F1   *..F...=...........H.USRT002 SYS1*
  000000A0 0000A0    40404040 00108900   00000000 AF2D2987   22BA8401 00188A00   1997240F 05480170   *    ..I........G..D......P......*
  000000C0 0000C0    0000028D 00000000   00000000 005C8B00   1997240F 05480170   0000028D 000000C0   *.............*...P.............*
  000000E0 0000E0    00000000 00000000   00000000 00000000   00000000 00000000   00000E01 00000000   *...............................*
  00000100 000100    00000000 00000000   00000E01 0022000E   C9D4E2F2 40404040   B0AAC57B AB106A02   *...............IMS2    ..E#....*
  00000120 000120    00000000 00000000   00668C00 0000000E   00000000 00000001   C1D7D6D3 F1F14040   *........................APOL11  *
  00000140 000140    C9D4E2E4 E2F0F140   220E0158 40080000   00000000 00000000   00000000 00000000   *IMSUS01 .... ..................*
  00000160 000160    00000000 00000000   00000000 00000000   00000000 00000000   00000000 000008D3   *..............................L*
  00000180 000180    C9D5D2F1 F3D4F240   40404040 40400028   8D000000 00000000   00000000 00000000   *INK13M2       .................*
  000001A0 0001A0    00000000 00000000   00000000 00000000   00000000 0000001C   0300C1D7 D6D3F1F1   *........................APOL11*
  000001C0 0001C0    40D4C5E2 E2C1C7C5   40E3D640 C1D7D6D3   F1F1B0AA C57BAB19   CC020000 00000000   * MESSAGE TO APOL11..E#.........*
  000001E0 0001E0    0EBF                                                                            *..                             *
```

Figure 88. Sample log record showing successfully requeued message

Part 5. References

The topics in this section provide you with technical references for the Base Primitive Environment (BPE) and other reference information.

Topics:

- Chapter 25, "Base Primitive Environment (BPE) configuration PROCLIB member," on page 457
- Chapter 26, "How to read syntax diagrams," on page 465

Chapter 25. Base Primitive Environment (BPE) configuration PROCLIB member

This topic describes the syntax and format of the IMS Queue Control Facility BPE configuration PROCLIB member.

Topics:

BPE PROCLIB member overview

You can specify the settings of several BPE runtime parameters through the use of BPE PROCLIB members.

For example, you can set the level of BPE and IMS component trace tables, and you can associate user exit routines with an IMS component user exit routine type. The following rules apply to the format of all BPE PROCLIB members:

- The PROCLIB data set must have an LRECL of at least nine (80 is typical) and a fixed record format.
- The rightmost eight columns of each record are ignored, you can use them for sequence numbers or any other notation.

  Code the keyword parameters in the remaining columns.

  For example: if your record size is 80, use columns 1 through 72 for your configuration data.

  You can use columns 73 through 80 for sequence numbers or you can leave them blank.
- You can specify keywords with leading and trailing blanks.
- You can specify multiple keywords in each record.
- Use commas or spaces to delimit keywords.
- Use an asterisk (*) or pound sign (#) in column one to begin a comment.

  You can also include a comment anywhere within a statement by enclosing the comment between a slash-asterisk and an asterisk-slash pair.

  Comments between slash-asterisk and asterisk-slash pairs can span multiple lines.

  For example:

  ```
  /*This is an example of a comment within a statement*/
  ```

  ```
  /* This is an example
     of a comment that
     spans muliple lines */
  ```

- You can continue statements across multiple lines by breaking the statement at a word boundary and continuing the statement on the next line.

  For example:

  ```
  TRCLEV=(AWE,
  HIGH,BPE)
  ```

- The values that are coded in PROCLIB members are case-sensitive.

Use uppercase characters for all PROCLIB members.

BPE configuration PROCLIB member parameters

You can use the BPE configuration PROCLIB member to define BPE run environment settings for the address space being started.

Specify the PROCLIB member name by coding BPECFG=*member_name* on the EXEC PARM= statement in the address space startup JCL, as shown in the following example:

```
EXEC IQCINI0,PARM='BPECFG=BPECFGCQ'
```

You can use the BPE configuration PROCLIB member to specify the following items:

- The language used for BPE and IMS component messages
- The trace level settings for BPE and IMS component internal trace tables
- The name of a BPE exit list PROCLIB member where configuration information for IMS component user exit routines is stored
- The time interval between calls to the BPE statistics exit routines

The following parameters are available for the BPE configuration PROCLIB member:

- LANG=
- TRCLEV=
- EXITMBR=
- STATINTV=

Recommendation: Avoid coding statements in the BPE configuration member that specify definitions for the same resources multiple times. For example, avoid multiple TRCLEV statements for the same trace table type, or multiple EXITMBR statements for the same IMS component type. BPE uses the last statement that it encounters in the member. Any values that are specified on earlier duplicate statements are ignored. A message, BPE0017I is issued for each duplicate statement found.

LANG parameter

The LANG parameter specifies the language used for IMS Queue Control Facility BPE component messages.

The syntax for the BPECFG= LANG parameter is as follows:

▶▶— LANG=ENU —▶◀

ENU is for US English, which is currently the only supported language.

TRCLEV parameter

The TRCLEV parameter specifies the trace level for a trace table

The syntax of the BPECFG= TRCLEV parameter is as follows:

▶▶— TRCLEV= —— (*type* ,*level* ,*ims_component* ———————————————) —▶◀
                                          └─ ,PAGES= *num_pages* ─┘

The TRCLEV parameter specifies the trace level for a trace table and, optionally, the number of storage pages allocated for the trace table. TRCLEV= controls the level of tracing (the amount of detail traced) for each specified trace table type. BPE-managed trace tables are areas in storage where BPE, and the IMS component that uses BPE, can trace diagnostic information about events occurring within the address space.

BPE-managed trace tables are internal in-core tables only. Trace records are not written to any external data sets. Some trace table types are defined and owned by BPE itself. These are known as *system trace tables*, and are present in all IMS component address spaces that use BPE. The IMS component can also define its own trace tables. These are known as component trace tables or user-product trace tables, and are only present in address spaces of the defining IMS component. For example, trace table types defined by Common Queue Server (CQS) are only present in a CQS address space.

You can share one BPE configuration parameter PROCLIB member among several different IMS component address spaces. Any TRCLEV statements you code for system trace tables apply identically to all of the address spaces that share the PROCLIB member. TRCLEV statements for a particular IMS component trace table are processed only by address spaces running that component. For example, you could have a BPE configuration parameter PROCLIB member containing TRCLEV statements for BPE, CQS, and Resource Manager (RM) trace table types. When you start a CQS address space, only the BPE and CQS TRCLEV statements are processed. When you start an RM address space, only the BPE and RM TRCLEV statements are processed.

type
Specifies the type of trace table. Each trace table has a four-character type. A trace table's type refers to the kind of events that are traced into that table. For example, the BPE DISP trace table contains entries related to events in the BPE dispatcher.

level
Controls how much tracing is done in the specified trace table. Each trace entry that is performed has a level associated with the entry. Each trace table has a level setting that is controlled by the value that you specify on the TRCLEV statement for the table.

A trace entry is written only if the trace entry's level is less than or equal to the table's level setting. For example, if the trace entry level is MEDIUM, the trace entry is added to the trace table only if the table's level is MEDIUM or HIGH. Thus, the level that you specify controls the volume (number) of trace entries that are written to a given table.

A low setting of the level parameter results in fewer trace entries being written to the table. The trace table does not overwrite itself (wrap) as quickly as with a higher setting (which means that diagnostic information remains available for a longer period of time), and the performance impact is minimized. However, the trace information is not as detailed as with higher settings, so the captured information might not be sufficient to solve a problem.

A high setting of the level parameter results in more trace entries being written to the table. This can provide additional diagnostic information for solving a problem; however, the trace table tends to wrap more frequently, and higher settings can cause additional CPU usage.

Choose one of the following for the level parameter:

NONE
No tracing.

Recommendation: Do not specify NONE because no tracing, not even tracing for error conditions, is done for the specified table.

ERROR
Only trace entries for error conditions are made. ERROR is the default.

LOW
Low-volume tracing (key component events). This is the minimum recommended trace level setting for normal operation.

MEDIUM
Medium-volume tracing (most component events).

HIGH
High-volume tracing (all component events).

ims_component
Specifies the IMS component that defines the trace table type. The only possible value at this time is IQC.

PAGES=*num_pages*

> An optional parameter that can be used to specify the number of 4 KB pages to be allocated for the table type.
>
> Specify a value from 1 to 32767 pages for this parameter.
>
> If BPE is unable to get the amount of storage that you requested for a trace table, BPE will try to get a smaller number of pages to enable some tracing to be done.
>
> You can see the actual number of pages that IMS Queue Control Facility BPE obtained for each trace by issuing the **DISPLAY TRACETABLE** command.
>
> If you do not use this PAGES=*num_pages* parameter, then the trace table has the default number of pages, as specified under the description of each trace table type.

BPE exit list PROCLIB member

Use the PROCLIB members that are specified by the EXITMBR= parameter in the BPE configuration parameter PROCLIB member to define user exit routines to BPE.

BPE Exit List PROCLIB members are IMS-component specific. You specify one EXITMBR statement for each IMS component that provides user exit routines through BPE services.

Each EXITMBR statement specifies the name of a PROCLIB member that contains the definitions for exit routines for that IMS component.

You can have a separate exit list PROCLIB member for each IMS component, or you can share one exit list PROCLIB member among several IMS components.

A BPE exit list PROCLIB member associates a user exit routine type with a list of one or more user exit routines. Use the EXITDEF statement to define the exit routine modules to be called for a particular exit routine type.

The BPE exit list PROCLIB member is processed by BPE during address space initialization. The BPE exit list PROCLIB member is also processed when you enter a **REFRESH USEREXIT** command.

Recommendation: Avoid coding statements in the BPE exit routine list member that specify definitions for the same exit routine type multiple times. BPE always uses the last statement that it encounters in the member for a particular exit routine. Any earlier statements for the same exit routine are ignored. Message BPE0017I is issued for each duplicate statement found.

If you code the same user exit routine name more than once in the exit routine list (EXIT=) of any single EXITDEF= statement, BPE always uses the first occurrence of the exit routine module name to determine the order for calling the exit routines. Duplicate names are ignored, and a message, BPE0018I is issued for each duplicate name.

EXITDEF parameter

The EXITDEF statement associates an exit routine type with a list of one or more exit routine modules to be called.

The syntax of the BPE EXITMBR= EXITDEF parameter is as follows:
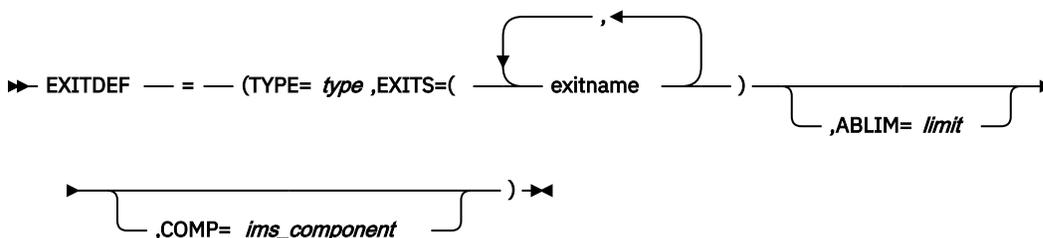
The modules are called in the order listed. The EXITDEF statement consists of a sublist (enclosed in parentheses) containing the keywords TYPE, EXITS, ABLIM, and COMP.

TYPE=*type*
Specifies the type of exit routine. The IMS component defines the types of exit routines that are supported.

EXITS=(*exitname,...***)**
Specifies a list of one or more exit routine module names. The position of the exit routine in the list determines the order in which the exit routine is driven. When an exit routine returns to its caller, the exit routine indicates whether additional exit routines are to be called.

ABLIM=*limit*
A number from 0 to 2147483647 that specifies the abend limit for the type of exit routine being defined. If the number of abends for an exit routine module reaches the abend limit for the exit routine type, the module is removed from the exit routine list and is not called until the exit routine type is refreshed.

This parameter is optional; the default is 1. If you specify a value of 0, there is no limit to the number of abends that can occur for the type of exit routine that you are defining.

COMP =*ims_component*
An optional parameter that specifies the type of IMS component that owns the exit routine being defined. Possible values are:

BPE
> Base Primitive Environment

BPE processes only EXITDEF statements that:

- Do not have COMP coded
- Have COMP=*ims_component* coded, where *ims_component* matches the IMS component specified on the EXITMBR statement that points to the BPE user exit PROCLIB member that is currently being processed.

  For example, if BPE were processing the BPEEXIT0 PROCLIB member specified on the EXITMBR=(BPEEXIT0,BPE) statement, it would only process EXITDEF statements that had no COMP= specified, and those that had COMP=BPE specified.

  If BPE were processing the CQSEXIT0 PROCLIB member specified on the EXITMBR=(CQSEXIT0,CQS) statement, it would only process EXITDEF statements that had no COMP= specified, and those that had COMP=CQS specified.

  For any given IMS component address space, BPE only processes BPE user exit PROCLIB members for EXITMBR statements that specify BPE, and those that specify the IMS component name of the address space that is running (for example, CQS, DBRC, HWS, ODBM, OM, RM, or SCI).

The EXITDEF *types* provided by the various IMS components are described in the following topic.

BPE EXITDEF types

There are 2 types of BPE EXITDEF types.

INITTERM
Called once during early BPE initialization, and once during normal termination.

STATS
Called periodically (timer-driven) and once during normal address space shutdown with statistics about BPE system functions. Optionally, the IMS component running on top of BPE can provide statistics that are specific to its operation.

Important: All BPE-owned user exit routines are available to all IMS address spaces which are running with BPE.

Sample BPE user exit list PROCLIB member

The BPE user exit list PROCLIB member defines the BPE initialization/termination and statistics exit routines.

A sample BPE user exit list PROCLIB member is shown in the following example. The sample defines the following exit routines:

- One BPE initialization/termination exit routine
- One BPE statistics exit routine

Example of a BPE user exit routine list PROCLIB member

```
**********************************************************************
* BPE USER EXIT LIST PROCLIB MEMBER                                  *
**********************************************************************
#------------------------------------------------------------------#
# Define one BPE init/term exit: MYINIT00.                          #
#------------------------------------------------------------------#
EXITDEF(TYPE=INITTERM,EXITS=(MYINIT00))
#------------------------------------------------------------------#
# Define 1 BPE Statistics exit: HHGSTAT0 with an abend limit of 42 #
#------------------------------------------------------------------#
EXITDEF(TYPE=STATS,EXITS=(HHGSTAT0),ABLIM=42)
```

IMS Queue Control Facility-specific TRCLEV statements

Set the required IMS Queue Control Facility-specific TRCLEV statements for the IMS Queue Control Facility BPE configuration PROCLIB member.

Specifying the IMS Queue Control Facility trace table types

IMS Queue Control Facility provides a set of trace table types for tracing processing within the IMS Queue Control Facility control address space.

About this task

To trace processing within the IMS Queue Control Facility control address space, a set of trace table types are provided. These IMS Queue Control Facility trace table types are present only in an IMS Queue Control Facility control address space.

TRCLEV statements specifying a component of IMS Queue Control Facility are ignored for any other address space type.

*

Specifying a type of * enables you to set the default trace level and optionally, the default number of pages per trace table for all of the IMS Queue Control Facility-defined trace table types. If you use the * type, make sure that the * type is the first TRCLEV statement for IMS Queue Control Facility-defined trace table types in your PROCLIB member. You can then code additional TRCLEV statements for specific IMS Queue Control Facility types to selectively override the defaults.

Recommendation: Code a TRCLEV statement with a type of * for IMS Queue Control Facility traces and also specify a minimum level of LOW as your first TRCLEV statement for IMS Queue Control Facility-defined trace table types. Using this coding scheme ensures that tracing is done for all IMS Queue Control Facility trace tables. Using this coding scheme also ensures that any new trace table types that are added in the future will be turned on in your system, even if you have not modified your BPE configuration parameter PROCLIB member to explicitly add a TRCLEV statement.

ERR

The ERR trace table traces error events within an IMS Queue Control Facility address space. The default number of pages for this table is 4.

Restriction: You cannot set the level for the ERR trace table. BPE forces the level to HIGH to ensure that error diagnostics are captured. Any level that you specify for the ERR trace table is ignored. You can, however, specify the number of pages for the ERR trace table on the TRCLEV statement.

INI

The INI trace table traces events that are related to IMS Queue Control Facility initialization processing in the IMS Queue Control Facility server address space. The default number of pages for this table is 4.

MST

The MST trace table traces events that are related to IMS Queue Control Facility master control service processing in the IMS Queue Control Facility server address space. The default number of pages for this table is 4.

CSV

The CSV trace table traces events that are related to IMS Queue Control Facility common service processing in the IMS Queue Control Facility server address space. The default number of pages for this table is 4.

CVC

The CVC trace table traces events that are related to IMS Queue Control Facility client conversation processing in the IMS Queue Control Facility server address space. The default number of pages for this table is 4.

REQ

The REQ trace table traces events that are related to IMS Queue Control Facility client request processing in the IMS Queue Control Facility server address space. The default number of pages for this table is 4.

RSP

The RSP trace table traces events that are related to the response from IMS Queue Control Facility extensions. The default number of pages for this table is 4.

SAF

The SAF trace table traces events that are related to the SAF manager function during IMS Queue Control Facility server address space. The default number of pages for this table is 4.

CMD

The CMD trace table traces events that are related to the command processor during the IMS Queue Control Facility server address space. The default number of pages for this table is 4.

SCI

The SCI trace table traces message activity between the IMS Queue Control Facility server and any active IMS Queue Control Facility clients that request or respond to manipulate the IMS message queues. SCI messages are used to transmit control information from the IMS Queue Control Facility client and message queues output from IMS systems to the IMS Queue Control Facility server address space. The default number of pages for this table is 4.

Chapter 26. How to read syntax diagrams

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

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

  ▶▶─ *required_item* ─▶◀

- Optional items appear below the main path.

  ▶▶─ *required_item* ─────────────────▶◀
                      └─ *optional_item* ─┘
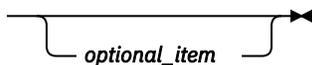
  If an optional item appears above the main path, that item has no effect on the execution of the syntax element and is used only for readability.

                      ┌─ *optional_item* ─┐
  ▶▶─ *required_item* ─┴─────────────────┴─▶◀
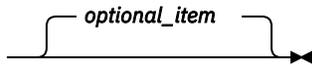
- If you can choose from two or more items, they appear vertically, in a stack.

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

  ▶▶─ *required_item* ─┬─ *required_choice1* ─┬─▶◀
                      └─ *required_choice2* ─┘
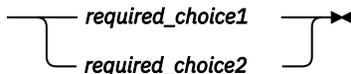
  If choosing one of the items is optional, the entire stack appears below the main path.

  ▶▶─ *required_item* ─────────────────────▶◀
                      ├─ *optional_choice1* ─┤
                      └─ *optional_choice2* ─┘
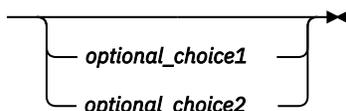
  If one of the items is the default, it appears above the main path, and the remaining choices are shown below.

                      ┌─ *default_choice* ──┐
  ▶▶─ *required_item* ─┼─────────────────┼─▶◀
                      ├─ *optional_choice* ─┤
                      └─ *optional_choice* ─┘
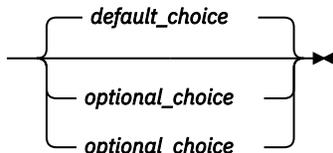
- An arrow returning to the left, above the main line, indicates an item that can be repeated.

             ┌─────────────────┐
  ▶▶─ *required_item* ─▼─ *repeatable_item* ─┴─▶◀
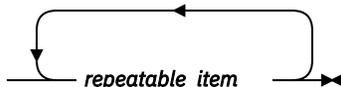
If the repeat arrow contains a comma, you must separate repeated items with a comma.

▶▶─ *required_item* ──┬─ *repeatable_item* ─┬─▶◀
                    └──────← , ──────┘
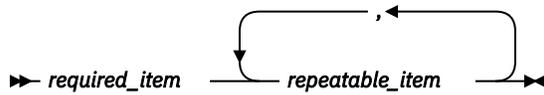
A repeat arrow above a stack indicates that you can repeat the items in the stack.

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

▶▶─ *required_item* ──│ fragment-name │─▶◀

fragment-name

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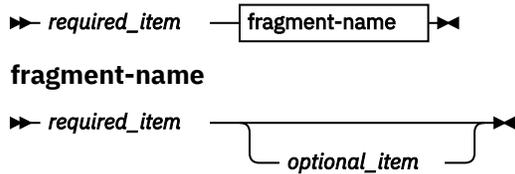
- A b symbol indicates one blank position.
- Keywords, and their minimum abbreviations if applicable, appear in uppercase. They must be spelled exactly as shown. Variables appear in all lowercase italic letters (for example, *column-name*). They represent user-supplied names or values.
- Separate keywords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

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 Director of Licensing
IBM Corporation
North Castle Drive

Armonk, NY 10504-1785
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.

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.

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at http://www.ibm.com/legal/copytrade.shtml.

Other company, product, and service names may be trademarks or service marks of others.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions:

Applicability: These terms and conditions are in addition to any terms of use for the IBM website.

Personal use: You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use: You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights: Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED,

INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

Privacy policy considerations

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at http://www.ibm.com/privacy and the section titled "Cookies, Web Beacons, and Other Technologies" in IBM's Online Privacy Statement at http://www.ibm.com/privacy/details. Also, see the "IBM Software Products and Software-as-a-Service Privacy Statement" at http://www.ibm.com/software/info/product-privacy.

Index

Numerics

6701-MRQE diagnostic records 452

A

accessibility
　　overview 9
ACTIONPOOL control statement 56, 57
AIB reason codes 399
architecture 11
AREA and FAILSAFE actions
　　recovering from 439
　　recovery using commands and TSO client 445
　　reference 439
AREA and FAILSAFE settings, queue overflow protection 99
AREA and FAILSAFE statement guidelines 113
AREA control statement 56, 58

B

Base Primitive Environment (BPE)
　　configuration PROCLIB member 457
　　configuration PROCLIB member parameters 458
　　exit list PROCLIB member 460
　　EXITDEF parameter 460
　　EXITDEF types 461
　　LANG parameter 458
　　PROCLIB member overview 457
　　sample BPE user exit list PROCLIB member 462
　　specifying trace table types 462
　　TRCLEV parameter 458
　　TRCLEV statements 462
BMP test tool 47
BPE
　　configuration PROCLIB member 457
　　configuration PROCLIB member parameters 458
　　exit list PROCLIB member 460
　　EXITDEF parameter 460
　　EXITDEF types 461
　　LANG parameter 458
　　PROCLIB member overview 457
　　sample BPE user exit list PROCLIB member 462
　　specifying trace table types 462
　　TRCLEV parameter 458
　　TRCLEV statements 462
BPE PROCLIB member, example 30
BPECFG 19, 28
BPEINIT 19, 28
BROWSE function
　　browse processing 150
　　control statement abbreviations 152
　　control statements 151
　　description 149
　　FUNCTION BROWSE INCLUDE statement syntax 153
　　FUNCTION BROWSE SELECT statement syntax 156
　　FUNCTION BROWSE statement syntax 153

BROWSE function *(continued)*
　　JCL for the BROWSE function 157
　　overview 117
browse processing 150
browsing the message queues 149
buffer definitions for HIOP
　　overriding 52
business scenarios 11, 14

C

cancel logic 202
coexistence 77
comment (*) statement 125
commit data set 24
common control statements 121
configuration members for message processing 24
control processing control statements 122
control statement abbreviations 123
cookie policy 467

D

destination control security for TSO client and batch job users 39, 42
dialog boxes
　　Sort Columns 282
documentation
　　accessing 8
　　sending feedback 8

E

error messages 309
EXCLUDE statement 125
EXEC statement PARM field 121
EXITDEF parameter 460
EXITDEF types 461
extensions configuration 34

F

FAILSAFE and AREA actions
　　recovering from 439
　　recovery using commands and TSO client 445
　　reference 439
FAILSAFE and AREA settings, queue overflow protection 99
FAILSAFE and AREA statement guidelines 113
FAILSAFE control statement 56, 64
fallback 77
FUNCTION statement 130
functions
　　automatically locating checkpoints 120
　　BROWSE function 118
　　comment (*) statement 125
　　common control statements 121

IBM ®

Product Number: 5697-N50