IBM MQSeries Workflow

Administration Guide

Version 3.3

SH12-6289-04

IBM MQSeries Workflow

Administration Guide

Version 3.3

SH12-6289-04

Note!

Before using this information and the product it supports, be sure to read the general information under "Notices" on page 101.

Fifth Edition (March 2001)

This edition applies to version 3, release 3 of IBM MQSeries Workflow (product number 5697-FM3) and to all subsequent releases and modifications until otherwise indicated in new editions.

This edition replaces SH12-6289-03

Contents

About this book

This book provides information about administration functions and practices within an IBM MQSeries Workflow system or system group. It explains the basic concepts of system administration and describes how to use the MQSeries Workflow administration utility to administer and oversee an MQSeries Workflow system or system group. It also describes the audit trail and the command-line application used to delete records from the audit trail.

It is assumed that you have read the *IBM MQSeries Workflow: Concepts and Architecture* book and are familiar with the MQSeries Workflow system structure. You should also understand how MQSeries Workflow uses the database to store domain, system group, and system properties.

Hereafter IBM MQSeries Workflow is referred to as MQ Workflow.

Who should read this book

This book is intended for a system administrator who is the first person defined in an MQ Workflow system. A system administrator does the following:

- Installs and configures MQ Workflow and its prerequisite and corequisite products.
- Administers MQ Workflow databases and the day-to-day operation of MQ Workflow.

This book does not describe installation and configuration of MQ Workflow products. It assumes that your MQ Workflow system has already been set up using the *IBM MQSeries Workflow: Installation Guide*.

Conventions and terminology used in this book

Convention used	How it is used
Book titles are shown in italics.	*IBM MQSeries Workflow: Administration Guide*
Commands, menu options, and push buttons are shown in bold. Bold is also used to highlight important points.	Click **OK**.
Variables are shown in italics. New terminology and keywords are also shown in *italics*.	If you do not specify a value for *timestamp*, you are prompted to enter a value.

This symbol flags suggestions, important notes, and practical techniques.

How this book is organized

"Part 1. System administration" on page 1 introduces the concepts and components of system administration in an MQ Workflow system and explains how to start and use the MQ Workflow administration utility.

"Part 2. Problem determination" on page 33 gives details regarding error logging and explains how to use the trace facility of MQ Workflow. Further, the MQ Workflow audit trail and its command-line application are described.

The appendixes contain properties and tables, system message codes, shortcuts for selecting items from the administration utility menus, and information regarding a record and playback facility.

The back of the book includes a glossary that defines terms as they are used in this book, a bibliography, and an index.

How to get additional information

Visit the MQ Workflow home page at http://www.software.ibm.com/ts/mqseries/workflow

For a list of additional MQ Workflow publications, refer to "MQSeries Workflow publications" on page 111.

How to send your comments

Your feedback is important in helping to provide the most accurate and high-quality information. If you have any comments about this book or any other MQ Workflow documentation, choose one of the following methods:

- Send your comments by e-mail to: swsdid@de.ibm.com. Be sure to include the name of the book, the part number of the book, the version of MQ Workflow, and, if applicable, the specific location of the text you are commenting on (for example, a page number or table number).
- Fill out one of the forms at the back of this book and return it by mail, by fax (+49-(0)7031-16-4892), or by giving it to an IBM representative.

How to read the syntax diagrams

In this manual diagrams are used to illustrate programming syntax. To use a diagram, follow a path from left to right, top to bottom, adding elements as you go. In these diagrams, all spaces and other characters are significant.

Each diagram begins with a double right arrowhead and ends with a right and left arrowhead pair.

The following rules apply to the syntax diagrams used in this book:

- The ▶▶── symbol indicates the beginning of a statement.

  The ──▶ symbol indicates that the statement syntax is continued on the next line.

  The ▶── symbol indicates that a statement is continued from the previous line.

  The ──▶◀ symbol indicates the end of a statement.

  Diagrams of syntactical units other than complete statements start with the ▶── symbol and end with the ──▶ symbol.
- Required items appear on the horizontal line (the main path).

  ```
  ▶▶──required_item──────────────────────────────────────────────────────▶◀
  ```

- Optional items normally 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 statement and is used only for readability.

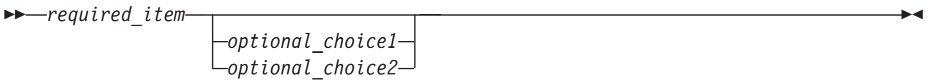
  ```
                     ┌─optional_item─┐
  ▶▶──required_item──┴───────────────┴───────────────────────────────────▶◀
  ```

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

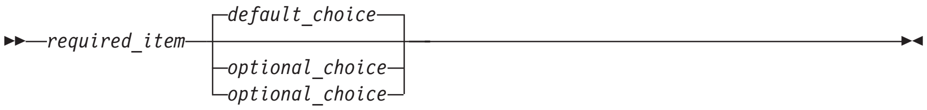
  If you *must* choose one of the items, one item of the stack appears on the main path.

  ```
  ▶▶──required_item──┬─required_choice1─┬────────────────────────────────▶◀
                     └─required_choice2─┘
  ```

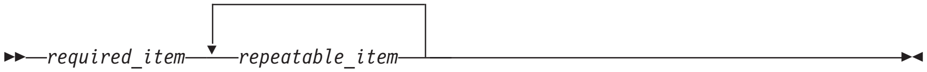
  If choosing one of the items is optional, the entire stack appears below the main path.



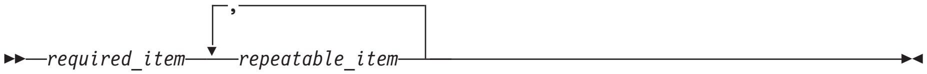
If one of the items is the default, it appears above the main path and the remaining choices are shown below.



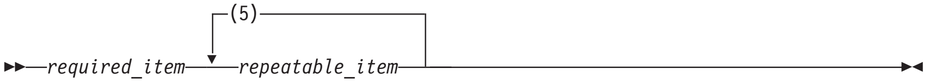
- An arrow returning to the left, above the main line, indicates an item that can be repeated.



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



If the repeat arrow contains a number in brackets, the number represents the maximum number of times that item can appear.



A repeat arrow above a stack indicates that you can repeat the items in the stack.

- Keywords appear in uppercase (for example, FROM). Variables appear in all lowercase letters (for example, *column name*). They represent user-supplied names or values.
- If punctuation marks, parentheses, arithmetic operators, or other such symbols are shown, you must enter them as part of the syntax.
- Syntax diagrams may be broken into fragments. A fragment is indicated by vertical bars with the name of the fragment between the bars. The fragment is shown following the main diagram, like so:

```
├──┤ A Fragment ├──────────────────────────────────────────────────────┤
```

A Fragment:

```
├──Keyword──value──────────────────────────────────────────────────────┤
```

Summary of changes

Changes for MQ Workflow Version 3.3

The changes in this book include both editorial changes as well as technical updates. The following changes have been made:

- The book has been restructured to group the chapters for the system administration and the problem determination of MQ Workflow.
- “Chapter 2. How to administer the MQ Workflow system” on page 11 now includes a description of the following utilities:
  - How to start a Program Execution Agent (see “Starting a Program Execution Agent” on page 28).
  - How to stop a Program Execution Agent (see “Stopping a Program Execution Agent” on page 30).
- “Chapter 4. How to use the system trace facility” on page 41 explains the different trace modes and describes how the external trace writer works. The chapter now includes a list of the available environment variables showing in which trace mode they can be used.
- “Analyzing the audit trail” on page 63 now also contains examples for SQL queries you can perform when using Oracle.
- “What is the audit trail?” on page 55 introduces the new audit options **AUDIT_TO_DB** and **AUDIT_TO_MQ**. Further, the new option value **Filter** is described.

Part 1. System administration

Chapter 1. Introduction to MQSeries Workflow system administration

This chapter introduces you to system administration in an IBM MQSeries Workflow system and describes the two main system administration components, namely, the administration server and the administration utility, and the method they use to communicate.

System administration in an MQSeries Workflow system

System administration is implemented to control and manage any MQ Workflow system within a system group. It is performed by an administration component which provides vital management, control, security, and operational functions that govern the running of a particular selected system within a system group. The administration component is made up of an administration server and an administration utility.

MQ Workflow has a hierarchical structure. The domain is the highest level in the hierarchy and can only contain one system group. Each system group is made up of one or more systems. Each system contains an administration server. Each system also contains an administration utility if it is selected as one of the installed components during the MQ Workflow installation. The administration utility can be installed on the same physical machine as the administration server or on a separate machine.

An MQ Workflow system has a tiered structure:

- **Tier 1 — Client tier**

  Tier 1 contains the MQ Workflow system clients, application programming interfaces, and Buildtime. They use the MQ client manager and client message layer to connect with the second tier. The administration utility sits within this tier and uses the administration API (application programming interface) to communicate with the administration server located in tier 2.
- **Tier 2 — Server tier**

  Tier 2 contains the group of servers that are collectively referred to as the MQ Workflow server. This is the working center where all the scheduling, distribution, cleanup, administration, server communication, and execution is done. The administration server is located in this tier. It communicates with the administration utility located in tier 1 and all other components in the system using MQSeries techniques, as explained in “Administration server/utility communications” on page 7.

- **Tier 3 — Database tier**

  Tier 3 contains the MQ Workflow relational database. This holds system status and setup information and stored procedures for the complete system. The administration server accesses system tables within the database and returns the contents back to the administration utility when requested to do so. The database is automatically updated by the administration server in response to system events.

The administration utility is the user interface to the administration server and is used by a system administrator to request services from the administration server. "Chapter 2. How to administer the MQ Workflow system" on page 11 explains how to use the administration utility.

For further details on the system structure see the *IBM MQSeries Workflow: Concepts and Architecture* book.

System administration client/server components

Every MQ Workflow system has an administration server. Using the administration utility, any authorized system administrator can access an administration server within any specified system as long as those systems are members of system groups that are contained within the same domain.

Figure 1 on page 5 illustrates the implementation of the administration component within an MQ Workflow system. All administration components within a system group are implemented in a similar way.

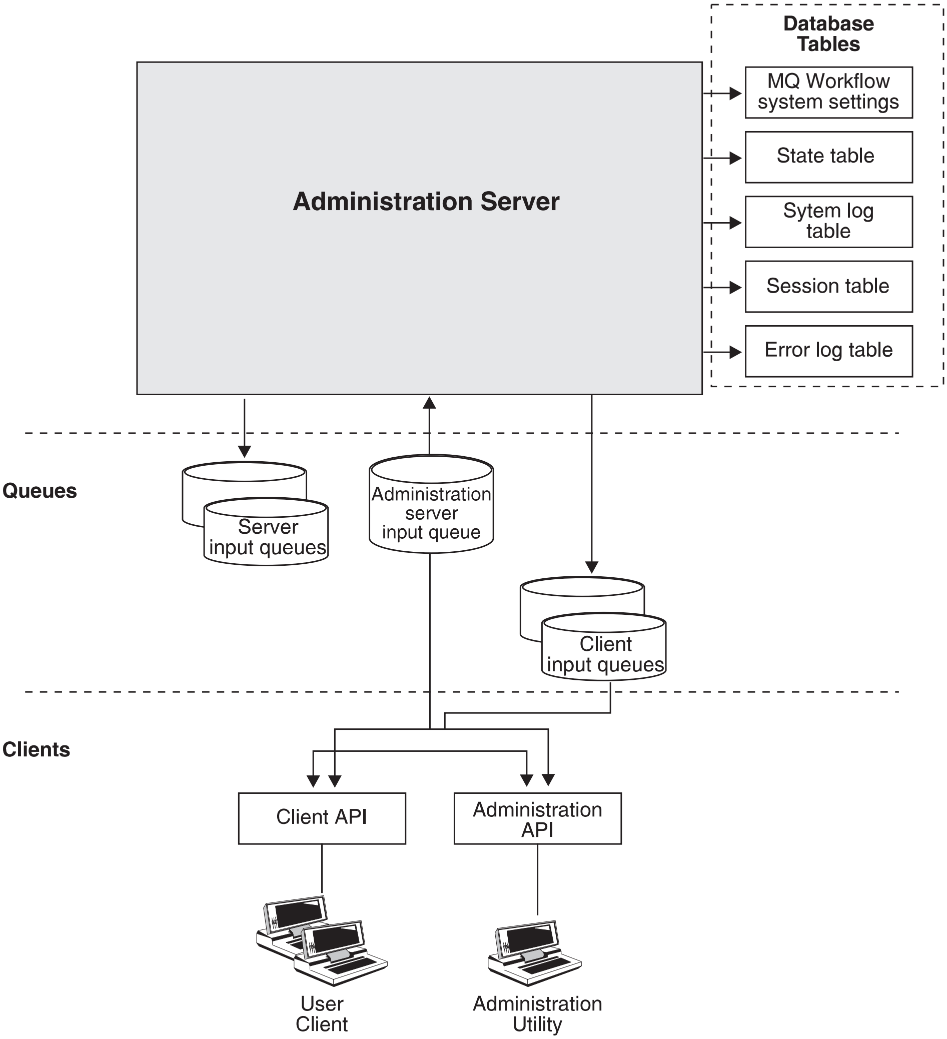


Figure 1. Implementation of the administration component in an MQ Workflow system

The administration server

The administration server is the working center of the administration component. It is responsible for the availability, operation, error recovery, and management of all components in an MQ Workflow system. It performs administrative functions in response to system administration requests, as well as automatic internal functions that are transparent to the system administrator. The administration server communicates with all other components in an MQ Workflow system and is responsible for session management in the MQ Workflow system. It handles all logon requests and checks user identification, password, and authorization for a requested session.

The administration server is always the first component in an MQ Workflow system that is started and it continues to run as long as any other server in the system is running. In fact, starting the administration server starts the system. It is also the last component that is shut down. Shutting down the administration server shuts down the complete system.

The administration server sends messages via queues that are managed by MQSeries, and has access to various system tables held in the system database.

Queues

All components in the system receive messages from input queues that are managed by MQSeries. MQSeries is IBM's message/queue communications product used to manage communications within an MQ Workflow system. The administration server uses such techniques to send messages to all system server and client input queues. The administration server maintains its own input queue from which all messages are received.

Database tables

The administration server accesses domain, system group, and system tables in the MQ Workflow database and makes their contents available to the administration utility. The following lists the tables that can be accessed by the administration server:

- The administration server state table, which lists the administration server state properties, and the operational status of system servers
- Property tables for all server and program execution agents (PEA), which list server and PEA properties
- The system properties table in which properties that determine the behavior of the system are contained
- The system group properties table in which properties that determine the behavior of the system group and some system properties are contained
- The domain properties table in which properties that determine the behavior of the domain and some system group and system properties are contained
- A system log table in which all significant system events are stored
- A session table in which a session record is created for each authorized user after logon
- An error log table in which all error information is stored

The administration utility

The administration utility is the system administrator’s user interface to the administration server. It communicates with the administration server using MQSeries. Using the administration utility you can ask the administration server to list system and server operational related properties, query, stop and start system components, as well as access system and error log records. The administration utility is a menu-based console application running in a dedicated MQ Workflow administration client shell and is explained in “Chapter 2. How to administer the MQ Workflow system” on page 11. Functions are performed by selecting items from menu lists. A record and playback facility is available that records commands previously entered and can then play them back, providing the complete automation of administration tasks.

Administration server/utility communications

The administration utility communicates with the administration server using the MQSeries message handling facility.

To illustrate how MQSeries works between the administration utility and server refer to Figure 2 on page 8. The administration utility requests a service from the administration server by putting a message on the administration server’s input queue (Queue 1). When the administration server takes the message from Queue 1, it uses the information in the message to process the request. If the administration server wants to return a response (an acknowledgment, for example), it does so by putting a message on the client’s input queue (Queue 2).

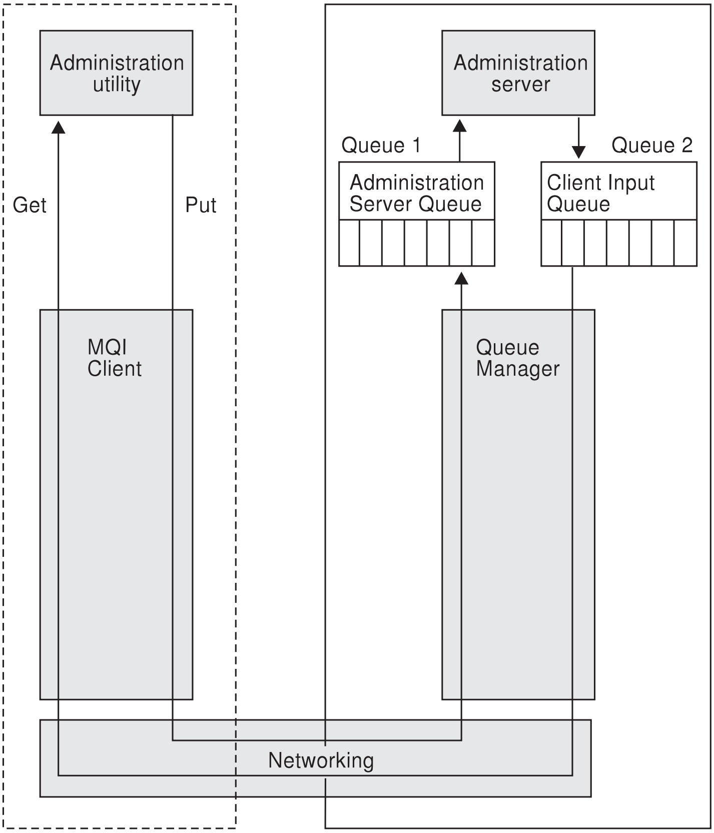


Figure 2. Client/server MQSeries communications

A queue manager is responsible for moving messages to and from both queues, and passing them to and retrieving them from the network on the server side. On the client side, messages are passed to the administration utility and retrieved from the network using the *Message Queue Interface (MQI)*. The MQI is a set of calls that programs use to ask for the services of a queue manager. The MQI and the queue managers are part of MQSeries.

The administration server and utility themselves operate in ignorance of the network and the methods of communication used, and therefore, bear no part of the networking load.

Administration messages

In ″Chapter 2. How to administer the MQ Workflow system″ the following terminology is used to describe two types of messages that are sent from either the administration server or utility:

- **Asynchronous requests**. Sent from the administration utility to the administration server.

- **Unsolicited messages**. Sent from the administration server to the administration utility.

These messages are sent using the MQSeries techniques previously explained and are described next.

Asynchronous requests

Messages can be both synchronous and asynchronous. Whereas a client that sends a synchronous request waits for a response from the server before doing anything else, a client that sends an asynchronous request does not. When you send an asynchronous request from the administration utility to the administration server you are not forced to wait for a response before being able to proceed with further requests. This means that you can continue to issue requests without interruption. It also means that you do not know if the server has honored and satisfied your request.

Unsolicited messages

An unsolicited message appears on the screen of your workstation running the administration utility whenever the administration server records an event in the system log. For details regarding the system log refer to “Executing system log commands” on page 26.

An unsolicited message is one that is not a direct response to a request, even though it may appear to be, but a message that indicates changes or activities that occur within the system. These may be information, warning, or error messages. An example of an unsolicited message is a ″server stopped″ message. This message may appear after a request to shut down a server is issued from the administration utility, or after an error situation occurred, in which case the administration server decided to shut down the server to avoid further problems. In either case the unsolicited message that appears is in response to the change in state of the server and not to the request from the administration utility or any other component of the system. The administration server writes the state change to the system log table in the MQ Workflow database and at the same time sends an unsolicited message to the administration utility.

Unsolicited messages are handled by a receive thread which is started as soon as a connection between an administration utility and server is made. The receive thread is responsible for writing unsolicited messages to the screen and into a log file.

Chapter 2. How to administer the MQ Workflow system

You administer the MQ Workflow system using the administration utility. The utility allows tasks to be performed by selecting objects from menu lists. However, before you can use the administration utility the administration server must be started. The first section of this chapter describes how to do this. The rest of the chapter explains how to start the administration utility and describes the menu lists available within it.

"Appendix C. Speeding up menu selection from the administration utility" on page 91 provides hints on automating and speeding up menu selections from the utility explained in this chapter. "Appendix D. The record and playback facility" on page 95 describes the record file and playback facility that provides further automation when performing system administration tasks and is useful for batch processing.

Starting the administration server

Before you can use the administration utility the administration server must be up and running. Starting the administration server starts the MQ Workflow system.

Each MQ Workflow configuration is identified by a configuration identifier that is set during configuration. If you do not specify a value for the configuration identifier switch (/y), the default value is read from the general configuration profile.

If it has not been started already, start the administration server as follows:

For AIX, HP-UX, and Sun Solaris:

1. Open a command prompt window and enter the command:

   fmcamain

For Windows NT, 2000:[1]

1. On the task bar, click the **Start** button and select **Settings**.
2. From **Settings** select **Control Panel**.
3. Within the **Control Panel** select the **Services** icon. A dialog box appears.

[1] For Windows NT, 2000, it is recommended to run the administration server as a *service*. However, you can also start the administration server as a command line program by entering **fmcamain -c** at a command prompt.

4. Within the service window of the dialog box locate the line that reads **MQSeries Workflow 3.2**. Highlight this line.
5. Click on the **Start** button to the right of the window to start the administration server.

Starting the administration utility

The following describes the **fmcautil** command used to start the administration utility and the command-line options that are available with it. To start and use the utility, establish a connection with an administration server as follows:

1. Open a command prompt window and enter the **fmcautil** command followed by the appropriate options. The syntax of the **fmcautil** command is as follows:

```
►►──fmcautil──────────────────────────────────────────────────────────────►

   ┌─default configuration identifier from general configuration profile─┐
►──┼─────────────────────────────────────────────────────────────────────┼──►
   └─-y──configurationidentifier─────────────────────────────────────────┘

   ┌─systemgroup from configuration profile─┐
►──┼────────────────────────────────────────┼─────────────────────────────►
   └─-g──systemgroup────────────────────────┘

   ┌─system from configuration profile─┐
►──┼───────────────────────────────────┼──┬──────────────┬──┬────────────────┬──►
   └─-s──system────────────────────────┘  └─-u──userid───┘  └─-p──password───┘

►──┬──────┬──┬───────────────────────────────┬──┬─────────────────────────────────┬──┬──────┬──►
   └─-f───┘  │       ┌─fmcautil.rec───────┐  │  │       ┌─fmcautil.rec─────────┐  │  └─-k───┘
             └─-r────┴─recordfilename─────┴──┘  └─-b────┴─playbackfilename─────┴──┘

►──┬─────────────────────────────┬────────────────────────────────────────►◄
   │       ┌─fmcautil.log──────┐ │
   └─-l────┴─logfilename───────┴─┘
```

For Windows NT, 2000: You can also use a slash (/) instead of a minus (-) as delimiter and lowercase letters.

The switches that precede values are used to indicate the meaning of the value that follows (for example, the **-p** switch indicates that the value that follows is the *password*). Switches that do not precede values are used to represent optional functions, such as **-f**. By using these switches you can specify values and options in any order you like.

-y *configurationidentifier*
If you do not specify a value for the *configurationidentifier*, the default value is read from the general configuration profile. Each MQ Workflow configuration is identified by a configuration identifier that is set during configuration.

-g *systemgroup*
If you do not specify a *systemgroup* name, the default value[2] is used. If no default value is specified in the MQ Workflow profile, you are prompted to enter a value. The value you enter should be the name of the system group containing the system that the administration server you wish to connect to is a member of.

-s *system*
If you do not specify a *system* name, the default value[2] is used. If no default value is specified in the MQ Workflow profile, you are prompted to enter a value. The value you enter should be the name of the system that the administration server you wish to connect to is a member of.

-u *userid*
For Windows NT, 2000, if you do not specify a *userid*, the value you used to log on to the operating system with is taken instead.

For operating systems other than Windows NT or Windows 2000, if you do not specify a *userid*, you are prompted to enter a value.

-p *password*
If you do not specify a *password*, you are prompted to enter a value unless you are running the administration utility under Windows NT, 2000 and *unified logon* has been set in the MQ Workflow database. Unified logon means that when NT or Windows 2000 client users have logged on to Windows NT, 2000 with their password, there is no need to further log on to individual applications. MQ Workflow supports unified logon when it is specified in the system properties table during MQ Workflow Buildtime. Refer to the *IBM MQSeries Workflow: Getting Started with Buildtime* book for details about setting unified logon.

[2] Default values for *systemgroup* and *system* are held in the MQ Workflow profile. For details regarding default values and the MQ Workflow profile refer to the *IBM MQSeries Workflow: Installation Guide*.

-f Enter the **-f** forcehere option when you are already connected to the selected administration server and you want to gain access to the same server from another location. By specifying the forcehere option you are effectively disconnecting yourself from the administration server leaving it open for you to reconnect again. You need to do this since you can only make one connection at a time to the administration server.

-r The **-r** record option is used to start the record facility. When used, this records all entries that you enter while using the administration utility into a record file. The name of the record file is specified by the *recordfilename* value you enter after the **-r** option. If you do not enter a file name, the default **fmcautil.rec** name is used. Refer to "Appendix D. The record and playback facility" on page 95 for details about the record file.

-b The **-b** playback option invokes the playback feature which reads and executes the entries within a file. The name of the file is specified by the *playbackfilename* value you enter after the **-b** option. If you do not enter a file name, the default **fmcautil.rec** name is used. Refer to "Appendix D. The record and playback facility" on page 95 for details about the playback feature.

If you specify the record and playback options together within the **fmcautil** syntax, make sure that the file names you specify after the record and playback options refer to different files.

-k The **-k** option is used after the playback option to stay within the administration utility once playback is complete. If it is not specified, the administration utility terminates after playback has completed.

-l The **-l** log option is used to create a log file that contains a record of all system events initiated by the administration server. All events are recorded into the file specified by the *logfilename* value you enter after the **-l** option. If you do not enter a file name, the default **fmcautil.log** name is used.

Once a connection with an administration server has been successfully made and the administration utility started, the following main menu window is displayed:

```
FMC15010I Main Menu:
     s ... System Commands Menu
     m ... Select Server Menu
     e ... Errorlog Commands Menu
     l ... Systemlog Commands Menu
     u ... User Commands Menu
     x ... Exit Main Menu
```

Figure 3. Main menu

2. Select one of the letters located in the left-hand column of the main menu to work on a listed object. The letter chosen is entered at the command prompt.

   To work on a listed object a connection must be established with the administration server. If no connection with an administration server has been established, only options **s** and **x** are displayed as follows:

   ```
   FMC15010I Main Menu:
        s ... System Commands Menu
        x ... Exit Main Menu
   ```

   To establish a connection with an administration server:

   a. Select **s** from the main menu. The following partial system commands menu is displayed:

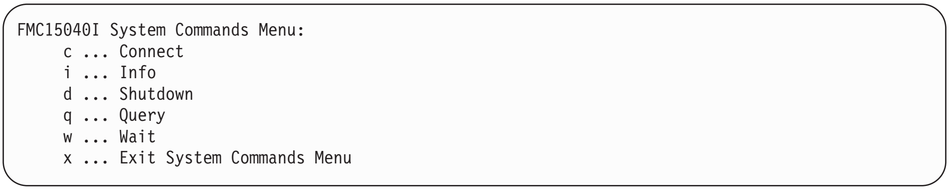
   ```
   FMC15040I System Commands Menu:
        c ... Connect
        w ... Wait
        x ... Exit System Commands Menu
   ```

   b. Proceed as explained in “Connecting to an administration server” on page 16.

   c. Select **x** from the system commands menu to return to the main menu screen. The full list of objects should be displayed as shown in Figure 3.

Executing system commands

If you select option **s** from the main menu, the following menu is displayed:



```
FMC15040I System Commands Menu:
     c ... Connect
     i ... Info
     d ... Shutdown
     q ... Query
     w ... Wait
     x ... Exit System Commands Menu
```

Figure 4. System commands menu

To exit the system commands menu, select option **x**. This returns you to the main menu screen.

Connecting to an administration server

Each MQ Workflow system within a system group contains an administration server. You can establish a connection with any administration server of any system by using the **Connect** subcommand. To establish a connection with a particular administration server:

1. Select option **c** from the system commands menu.
2. Specify the following information when prompted to do so:

   System group name
   The MQ Workflow system group in which the desired administration server resides.

   System name The MQ Workflow system in which the desired administration server resides.

   Userid The user ID of the authorized system administrator.

   Password The password for the user ID.

   If you press **Enter** when prompted for System group name, System name, Userid, or Password without specifying values for any of the fields, the initial values you entered in the **fmcautil** start command, described in step 1 on page 12, are used. If no values in the **fmcautil** start command were entered, no connection is made.

   Once a connection has been made, the workstation that the administration utility is running on is open to receive unsolicited messages from the administration server it is connected to. These messages appear on your screen whenever the administration server writes to the system log. (Refer to "Executing system log commands" on page 26 for details regarding the system log.) In addition, these messages are written to the log file you specified after the **-l** log option in the **fmcautil** start command.

Information about system settings

If you want to know information about the system to which you are connected, you can use the information option. This displays operational related settings held in the system properties table. To view information about the system:

- Select option **i** from the system commands menu.

  A list of properties that define the operation of the MQ Workflow system is displayed, such as *operation mode*, *message layer refresh interval*, and *session expiration time*. The meaning of each entry in the list is explained in Table 7 on page 72. Values for properties relating to time intervals, such as the *session expiration check interval*, are displayed in ISO format. This format is explained in "Operational properties" on page 71.

Shutting down the system

Shutting down the system refers to shutting down all server components of an MQ Workflow system, including the administration server. When you shut down the MQ Workflow system, all clients, including the API programs, are logged off by removing their session records from the MQ Workflow database. Then, all Program Execution Agents (PEAs) are shut down.

If you do not want to log off all clients and shut down all Program Execution Agents, you can add the *RTSystemShutdownMode* variable with the value *KeepSessions* to the MQ Workflow profile. If you set the *RTSystemShutdownMode* variable, clients are not logged off and running Program Execution Agents are not shut down. To set the variable use the configuration checker utility as follows:

```
fmczchk -y -c inst:m,RTSystemShutdownMode,KeepSessions
```

where *y* is the identifier for the MQ Workflow configuration. For details about the configuration checker utility and about how to set and change profile variables, refer to the *IBM MQSeries Workflow: Installation Guide*.

Make sure that the Administration server performs a regular cleanup of the sessions, because with this option the session records are no longer removed from the database during system shutdown. To do this, set an expiration time and an expiration check interval for the sessions in the settings of the MQ Workflow system.

To shut down the system:

- Select option **d** from the system commands menu. This sends an asynchronous request to the administration server requesting it to shut down all system servers including itself. A message appears to tell you that shutdown of the system has been requested.

  For every server that is shut down, an unsolicited message is sent by the administration server informing you about changes in the server's state.

The query command described in “Querying the system status” checks the current state of the servers and can be used to verify that servers have been shut down.

Querying the system status

The administration server regularly checks the state of all servers in the MQ Workflow system to verify operational status. These checks are performed according to the *check interval* values defined in the system, system group, and domain properties tables. By doing this, the administration server can monitor the system and take action in the case of problems. For example, if an execution server stops, after the administration server has checked the server’s state, it writes a record of the event in the system log, sends an unsolicited message to the system administrator, and restarts another execution instance.

To query the actual state of the system at any instance in time:

- Select option **q** from the system commands menu.

  A message code appears on the screen with message description. The description shows the status for each server, such as **’active’** or **’inactive’**, and the number of running instances for the execution server.

Waiting (system)

Waiting is used to wait for a specified period of time before proceeding to execute the next command. It is useful, for example, between starting a system and querying its status. The wait function, in this case, is used to delay querying of the system until all servers have started. It is most useful when performing multiple functions together as explained in “Shortcuts to menu selection” on page 91.

To hold execution for a given time interval:

1. Select option **w** from the system commands menu.
2. You are prompted for the wait time interval in seconds.

   Alternatively, you can enter the wait command and time interval on a single line as follows:

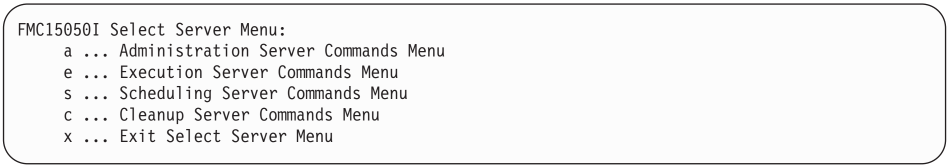
   ```
   w(t)
   ```

   where *t* is the period of time (in seconds) you want to wait.

   A message appears to tell you that the system is waiting for the length of time you specified. Once the wait is complete, another message appears to confirm this.

Executing server commands

If you select option **m** from the main menu, a list of all servers defined for the system is displayed in the following menu:

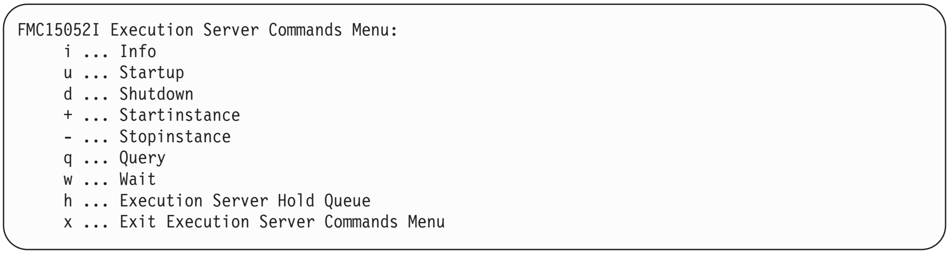


```
FMC15050I Select Server Menu:
     a ... Administration Server Commands Menu
     e ... Execution Server Commands Menu
     s ... Scheduling Server Commands Menu
     c ... Cleanup Server Commands Menu
     x ... Exit Select Server Menu
```

Figure 5. Select server menu

To exit the select server menu, select option **x**. This returns you to the main menu screen.

If you select a server, for example the execution server, from the select server menu, the following menu is displayed:



```
FMC15052I Execution Server Commands Menu:
     i ... Info
     u ... Startup
     d ... Shutdown
     + ... Startinstance
     - ... Stopinstance
     q ... Query
     w ... Wait
     h ... Execution Server Hold Queue
     x ... Exit Execution Server Commands Menu
```

Figure 6. Execution server commands menu

To exit this menu, select option **x**. This returns you to the select server menu screen.

Not all the entries shown in the above list are available for every server type selected.

Information about server properties

The behavior of all MQ Workflow system servers is determined by the server properties held in the system properties table.

To view server properties:

- Select option **i** from the server commands menu.

A list of properties that define the operation of the selected server is displayed, such as *number of initial instances* for the execution server, or *start mode* for the cleanup and scheduling server. The meaning of each entry in the list is explained in Table 8 on page 73. Values for properties relating to time intervals, such as the *check interval for execution server*, are displayed in ISO format. This format is explained in “Operational properties” on page 71.

Starting a server

You may wish to start a server if it has not already been started or if it has been previously shut down and you wish to restart it. The **startup** command in the server commands menu is available for starting all servers except the administration server.

To (re)startup any server **other than the administration server**:

- Select option **u** from the server commands menu. This sends an asynchronous request to the administration server requesting it to start the server. A message appears to tell you that startup of the server has been requested.

  Once the server has started, an unsolicited message is sent by the administration server to tell you that a change in the server’s state has occurred.

To (re)startup the **administration server** refer to “Starting the administration server” on page 11.

The query command described in “Querying the status of a server” on page 22 checks the current state of the server and can be used to verify that the server has started.

Shutting down a server

The shutdown option is available for all servers. However, shutting down the administration server has a different effect than the other servers. Shutting down the administration server shuts down all system servers as well as itself. Shutting down a server other than the administration server shuts down only the selected server.

To shut down a server:

1. Select option **d** from the server commands menu. This sends an asynchronous request to the administration server requesting it to shut down the server. A message appears to tell you that shutdown of the server has been requested.
2. If:
   a. **A server other than the administration server** has been selected, once the server has shut down, an unsolicited message is sent by the

administration server to tell you that a change in the server's state has occurred. A query as explained in "Querying the status of a server" on page 22 checks the current state of the server and can be used to verify that the selected server has been shut down.

b. **The administration server** has been selected, after the shutdown has been requested, only the **Wait** and **Exit** options are available for selection from the server commands menu. All other options are disabled until the administration server has been restarted and you have re-established a connection with it. To restart the administration server refer to "Starting the administration server" on page 11. To re-establish a connection refer to "Connecting to an administration server" on page 16.

   Before the administration server shuts itself down, it shuts down all other system servers. For every server that is shut down, an unsolicited message is sent by the administration server informing you about changes in the server's state.

Starting additional execution server instances

The execution server is a multiple instance server. This means that its server programs can be started multiple times. Each program started is called a server instance. During startup of the execution server the number of server instances as specified in the execution server properties table is started.

To start an additional execution server instance:

- Select option **+** from the server commands menu. This sends an asynchronous request to the administration server requesting it to start up a server instance. A message appears to tell you that startup of a server instance has been requested.

  Once the server instance has started, an unsolicited message is sent by the administration server to tell you that a new execution server instance has been started.

The query command described in "Querying the status of a server" on page 22 checks the current state of the server and can be used to verify the number of execution server instances running.

Stopping execution server instances

To stop an execution server instance:

- Select option **-** from the server commands menu. This sends an asynchronous request to the administration server requesting it to shut down a server instance. A message appears to tell you that a stop server instance has been requested.

  Once the server instance has stopped, an unsolicited message is sent by the administration server to tell you that an execution server instance has been stopped.

After the server instance is stopped, a query as explained in "Querying the status of a server" checks the current state of the server and can be used to check the remaining number of execution server instances running.

You cannot use the **-** option to shut down the execution server, even if you try to stop all running instances. This is because the administration server always leaves at least one instance running.

Querying the status of a server

The administration server regularly checks the state of all servers in the MQ Workflow system to verify operational status. These checks are performed according to the *check interval* values defined in the system properties table for each server.

To query the actual state of a server at any instance in time:

- Select option **q** from the server commands menu.

  A message code appears on the screen with message description. The description shows the status for the server and the number of running instances for the execution server.

Waiting (server)

As explained on page 18, the wait command is used to hold execution for a given time interval. To hold execution for a given time interval:

1. Select option **w** from the server commands menu.
2. You are prompted for the time interval in seconds.

   Alternatively, you can enter the wait command and time interval on a single line as follows:

   ```
   w(t)
   ```

   where *t* is the period of time (in seconds) you want to wait.

   A message appears to tell you that the system is waiting for the length of time you specified. Once the wait is complete, another message appears to confirm this.

Administering undelivered messages

Error events in the MQ Workflow system, such as database deadlocks, can cause transactions to fail. MQ Workflow retries such transactions until the customizable *message retry limit* is reached. When MQ Workflow is installed, the default retry limit for processing messages is set to 5. If there are many messages in the *hold queue*, you can increase the value temporarily by setting a new value for the profile variable *RTMessageRetryLimit*. It is recommended not to decrease the default retry limit. Use the configuration checker utility for setting the new value as shown in the following example:

```
fmczchk -y -c inst:m,RTMessageRetryLimit,10
```

where *y* is the configuration identifier of the MQ Workflow system.

For details about the configuration checker utility, refer to the *IBM MQSeries Workflow: Installation Guide*.

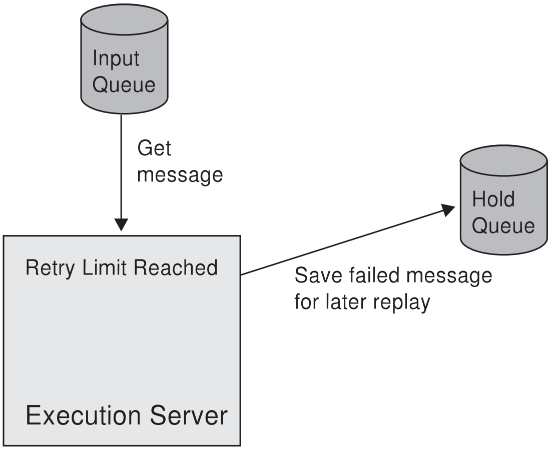


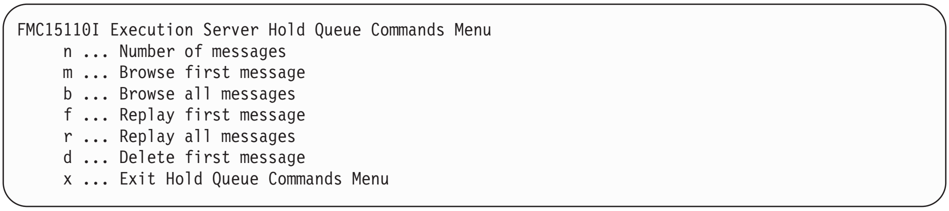
Figure 7. Failed messages are sent to the hold queue after the retry limit has been reached.

Once the retry limit is exceeded, failed messages sent to the execution server from other servers or the program execution agent are saved in a specific execution server *hold queue* for later processing.

Messages that exceed the retry limit indicate that the system needs attendance. For messages that exceed the retry limit, an entry is recorded in the system log. You should use the system log as your first source of information when trying to resolve message failures. The system log contains details about the failed message and normally indicates why it failed. After solving the problem, you can then use functions provided within the administration utility to restore the state of your processes.

Note that the *hold queue* contains failed server and program execution agent messages only. Client messages are discarded. In this respect, the *hold queue* can be considered to be the user interface from which failed server and program execution messages can be replayed. Failed client messages are immediately reported to the client where a user can recover from a failure by simply restarting the work item.

After finding and correcting the cause of the failure, the administration utility can be used to administer failed messages. By selecting option **h** from the execution server commands menu you can replay, browse, and delete messages contained in the *hold queue*. The following menu is displayed:



```
FMC15110I Execution Server Hold Queue Commands Menu
     n ... Number of messages
     m ... Browse first message
     b ... Browse all messages
     f ... Replay first message
     r ... Replay all messages
     d ... Delete first message
     x ... Exit Hold Queue Commands Menu
```

Figure 8. Execution server hold queue commands menu

To exit the hold queue commands menu, select option **x**. This returns you to the server commands menu.

Number of messages

By selecting option **n** from the execution server hold queue commands menu, the number of messages held in the *hold queue* is displayed.

Browsing messages

By selecting the appropriate browse option from the execution server hold queue commands menu, details for either the first message or all messages held in the *hold queue* are displayed:

Browse first message

Select option **m** to view the following information for the first message in the *hold queue*:

- Name of the system the message is associated with.
- Description of the failed message.
- Originator of the failing message (that is, which component caused the failure, for example the Program Execution Agent or the Server).
- Number of unsuccessful prior replays.
- Message content (that is, the message code that was sent).

Browse all messages

Select option **b** to view a list of all messages held in the *hold queue*. For each message, all the information listed for the **browse first message** option is listed except for the message content.

Replaying messages

By selecting the appropriate replay option from the execution server hold queue commands menu, either the first or all messages held in the *hold queue* are replayed:

Replay first message

Select option **f** to replay the first message in the *hold queue*.

Replay all messages

Select option **r** to replay all messages in the *hold queue*.

Deleting first message

By selecting option **d** from the execution server hold queue commands menu, the first message in the *hold queue* is deleted.

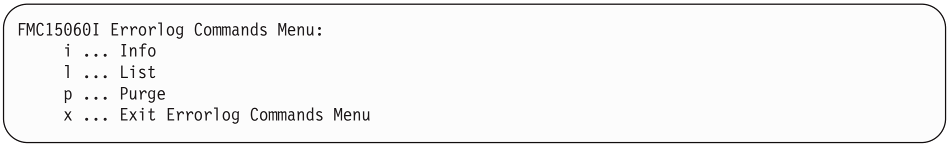
After the message is deleted, a message appears on the screen informing you that the message has been deleted.

Executing error log commands

The administration server is the focal point for error management within an MQ Workflow system. All errors and failures reported to the administration server are recorded in an error log table in the MQ Workflow database.

Error logging begins when the administration server is started and ends when the administration server is shut down. Further information on the error log and error management is given in “Chapter 3. How to use system and error logs” on page 35.

If you select option **e** from the main menu, the following menu is displayed:



```
FMC15060I Errorlog Commands Menu:
     i ... Info
     l ... List
     p ... Purge
     x ... Exit Errorlog Commands Menu
```

Figure 9. Error log commands menu

To exit the error log commands menu, select option **x**. This returns you to the main menu screen.

Information about error log properties

The system properties table held in the MQ Workflow database contains property values relevant to the error log. To view error log properties:

- Select option **i** from the error log commands menu.

  This displays the *error retain period*, which defines the length of time errors are to remain in the error log table. Once this period of time has elapsed, the administration server purges the contents of the error log table. The value displayed for the *error retain period* is in ISO format. This format is explained in “Operational properties” on page 71.

Listing error log records

For every error and failure reported to the administration server an error log record is created in the error log table in the MQ Workflow database.

To view the contents of the error log table:

- Select option **l** from the error log commands menu.

  All error log records contained in the error log table are displayed in a scrolling list. Table 11 on page 75 explains the meaning of the entries that make up each record contained in the error log table.

Purging error log records

If the error log table becomes too large to manage, you can delete all error log records using the **purge** subcommand.

To delete error log records from the database:

- Select option **p** from the error log commands menu. All error log records are deleted from the database.

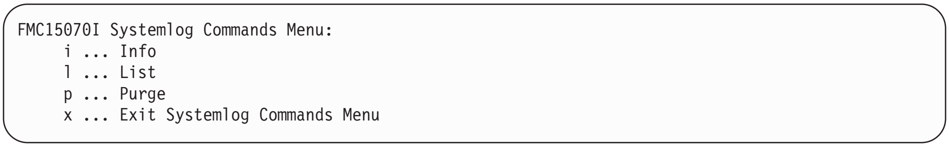
Executing system log commands

The administration server maintains a system log. The system log is a record of all system events sent or received by the administration server. For each event a system log record is created in the system log table, similar to error log records that are created in the error log table. Each record created contains a message number. This number is a coded description of the message sent or received by the administration server. Messages created by the administration server are explained in “Appendix B. System messages” on page 79. The administration server logs the following events in the system log table:

- Server state changes
- Message layer errors
- Database layer errors
- All MQ Workflow error events

Additionally, for Windows NT or Windows 2000, all system events are also recorded in the application log of the Windows NT, 2000 event log. Using the Windows NT, 2000 event viewer you can open the application log. System events are indicated by having **MQ Workflow** as the source entry. Double-clicking on the MQ Workflow system event opens a dialog box that contains a system message for that event. For further information regarding the application log refer to the Microsoft Windows NT, 2000 documentation.

If you select option **l** from the main menu, the following menu is displayed:



```
FMC15070I Systemlog Commands Menu:
     i ... Info
     l ... List
     p ... Purge
     x ... Exit Systemlog Commands Menu
```

Figure 10. System log commands menu

To exit the system log commands menu, select option **x**. This returns you to the main menu screen.

Information about system log properties

The system properties table held in the MQ Workflow database contains property values relevant to the system log. To view system log properties:

- Select option **i** from the system log commands menu.

  This displays the *message retain period*, which defines the length of time messages are to remain in the system log. Once this period of time has elapsed the administration server purges the contents of the system log. The value displayed for the *message retain period* is in ISO format. This format is explained in “Operational properties” on page 71.

Listing system log records

For every system event sent or received by the administration server a system log record is created in the system log in the MQ Workflow database.

To view the contents of the system log:

- Select option **l** from the system log commands menu.

  All system log records contained in the system log are displayed in a scrolling list. Table 12 on page 77 explains the meaning of the entries that make up each record contained in the system log. Within each system log record a message number is shown. This number is a coded description of the event that occurred. “Appendix B. System messages” on page 79 contains every message number that can occur and describes the meaning of each.

Purging system log records

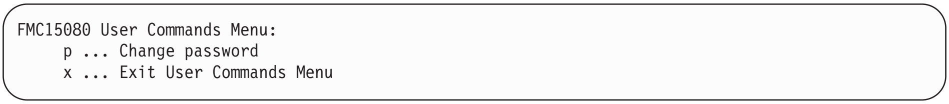
If the system log becomes too large to manage, you can delete all system log records using the **purge** subcommand.

To delete system log records from the database:

- Select option **p** from the system log commands menu. All system log records are deleted from the database.

Executing user commands

If you select option **u** from the main menu, the following menu is displayed:



```
FMC15080 User Commands Menu:
     p ... Change password
     x ... Exit User Commands Menu
```

Figure 11. User commands menu

To exit the user commands menu, select option **x**. This returns you back to the main menu screen.

Changing your password

By selecting the **Change password** option from the user commands menu you can change the password you use for logging on to MQ Workflow. To change your password:

1. Select option **p** from the user commands menu.
2. You are prompted to enter a new password.
3. You are prompted to reconfirm the new password.

For Windows NT and Windows 2000, if *unified logon* has been set in the MQ Workflow database, the **Change password** option is not available.

Starting a Program Execution Agent

The following describes the **fmcxspea** command used to start a Program Execution Agent (PEA) and the command-line options that are available with it. You can use the utility to start a Program Execution Agend manually. The following syntax diagram shows you how to start and use the utility:

```
►►──fmcxspea──────────────────────────────────────────────────────────────►


   ┌─default configuration identifier from general configuration profile─┐
►──┴─-y=──configurationidentifier────────────────────────────────────────┴──────────►


   ┌─systemgroup from configuration profile─┐
►──┴─-g=──systemgroup───────────────────────┴──────────────────────────────►
```

```
   ┌─system from configuration profile─┐
►──┼───────────────────────────────────┼──┬─────────────┬──┬─────────────────┬──────────►
   └─-s=──system───────────────────────┘  └─-u=──userid─┘  └─-p=──password───┘

►──┬────┬──────────────────────────────────────────────────────────────────────────────►◄
   └─-f─┘
```

The switches that precede values are used to indicate the meaning of the value that follows (for example, the **-p** switch indicates that the value that follows is the *password*). Switches that do not precede values are used to represent optional functions, such as **-f**. By using these switches you can specify values and options in any order you like.

-y= *configurationidentifier*
If you do not specify a value for the *configurationidentifier*, the default value is read from the general configuration profile. Each MQ Workflow configuration is identified by a configuration identifier that is set during configuration.

-g= *systemgroup*
If you do not specify a *systemgroup* name, the default value is used. If no default value is specified in the MQ Workflow profile, you are prompted to enter a value. The value you enter should be the name of the system group containing the system that the administration server you wish to connect to is a member of.

-s= *system*
If you do not specify a *system* name, the default value[3] is used. If no default value is specified in the MQ Workflow profile, you are prompted to enter a value. The value you enter should be the name of the system that the administration server you wish to connect to is a member of.

-u= *userid*
The **fmcxspea** utility logs on to MQ Workflow and starts the Program Execution Agent (PEA) for the *userid* you have specified. Then, the utillity is automatically logged off again.

-p= *password*
If you do not specify a *password*, you are prompted to enter a value for the password.

-f
Enter the **-f** forcehere option when you have already started a Program Execution Agent for the specified user ID and you want to start a Program Execution Agent for the same user ID on a different location. By specifying the forcehere option you

[3] Default values for *systemgroup* and *system* are held in the MQ Workflow profile. For details regarding default values and the MQ Workflow profile refer to the *IBM MQSeries Workflow: Installation Guide*.

can make sure to shut down the already started Program Execution Agents for the specified user ID before you start the new one. You need to do this since you can only start one Program Execution Agent at a time for a user ID.

Stopping a Program Execution Agent

The following section describes the **fmcxpsd** command and the command-line options that are available with it. You can use the utility to stop a Program Execution Agent (PEA) manually. The following syntax diagram shows you how to start and use the utility.

```
►►──fmcxpsd──────────────────────────────────────────────────────────────────►


   ┌─default configuration identifier from general configuration profile─┐
►──┼─────────────────────────────────────────────────────────────────────┼───►
   └─-y──configurationidentifier─────────────────────────────────────────┘


   ┌─systemgroup from configuration profile─┐
►──┼────────────────────────────────────────┼────────────────────────────────►
   └─-g──systemgroup────────────────────────┘


   ┌─system from configuration profile─┐
►──┼───────────────────────────────────┼──┬─────────────┬──┬──────────────┬──►◄
   └─-s──system────────────────────────┘  └─-u──userid──┘  └─-p──password─┘
```

The switches that precede values are used to indicate the meaning of the value that follows (for example, the **-p** switch indicates that the value that follows is the *password*). Switches that do not precede values are used to represent optional functions, such as **-f**. By using these switches you can specify values and options in any order you like.

-y *configurationidentifier*
If you do not specify a value for the *configurationidentifier*, the default value is read from the general configuration profile. Each MQ Workflow configuration is identified by a configuration identifier that is set during configuration.

-g *systemgroup*
If you do not specify a *systemgroup* name, the default value is used. If no default value is specified in the MQ Workflow profile, you are prompted to enter a value. If you specify a *systemgroup*, you must also specify a *system*.

-s *system*
If you do not specify a *system* name, the default value is used. If no default value is specified in the MQ Workflow profile,

you are prompted to enter a value. If you specify a *system* name, you must also specify a *systemgroup*.

-u *userid*
The **fmcxspsd** utility logs on to MQ Workflow and stops the Program Execution Agent (PEA) for the *userid* you have specified. Then, the utillity is automatically logged off again.

-p *password*
If you do not specify a *password*, you are prompted to enter a value for the password.

Part 2. Problem determination

Chapter 3. How to use system and error logs

The administration server maintains log files for viewing information about the system, controlling its status, and assisting in problem determination. The administration server is the focal point for collecting and recording all operation and error-related events and is responsible for maintaining the following logs:

The system log
The system log is a record of all operation-related events that take place in an MQ Workflow system. Examples of operation-related events are starting or stopping the system and its servers. Error events that take place within the system are also recorded in the system log.

The error log
The error log provides information that can be used for problem determination in addition to that contained in the system log. It contains a detailed error report and is updated whenever a component within an MQ Workflow system encounters a severe problem affecting its operation or causing it to fail.

Logging begins when the administration server is started and finishes when the administration server is shut down. It cannot be switched off.

In addition, MQ Workflow comes with a trace facility which traces MQ Workflow program instructions and can be used to diagnose reproducible problems.

This chapter talks about the main log files maintained by the administration server and applications used for error debugging.

Using the system log

In most cases, the cause of a problem in an MQ Workflow system can be determined from the system log. Therefore, the system log should be your first source of information when debugging errors. The type of messages recorded in the system log are listed in “Appendix B. System messages” on page 79.

The system log is maintained in the MQ Workflow database, the layout of which is shown in “System log record entries” on page 76. You can access the system log by using the administration utility. Since the administration utility and MQ Workflow Server can run on separate machines, it is possible to access the system log from a remote client machine.

As well as maintaining the system log in the MQ Workflow database, key contents of the system log are written to:

- The flat file **fmcsys.log**, which can be viewed with any ASCII editor. To view this file you need local access to the machine on which the administration server is running.
- The Windows NT, 2000 **application log**[4], which can be viewed with the Windows NT, 2000 event viewer.
- The screen of the **administration utilities** for all logged-on administration users. This means that the administration utility serves as a system console which is constantly updated, allowing you to monitor the state of a system.

By default, the **fmcsys.log** file is located in the subdirectory **log** of the MQ Workflow installation directory. The location and the name of this file is specified by the key **SystemLogFile** in the MQ Workflow profile. Changing the value of this key changes the location of the log file **fmcsys.log**. The value of this key must be a fully qualified file name (for example, c:\mylogs\fmcsys.log). See the *IBM MQSeries Workflow: Installation Guide* for details about the MQ Workflow profile.

The following shows how events recorded in the file **fmcsys.log** may appear:

```
02/15/99 02:48:12 PM FMC31030E  The queue manager FMCQM is not available. Reason code: 2059
02/15/99 02:50:42 PM FMC31020E  The server could not be started.
-------------------------------------------
02/15/99 02:50:42 PM FMC31000E  A database error occurred. SQL code: -998.
-------------------------------------------
02/15/99 03:21:20 PM FMC10110I  Administration server for system SYS started.
02/15/99 03:21:20 PM FMC10000I  System startup complete. System SYS in system group SYS_GRP
                                is now running.
```

Figure 12. Sample content of fmcsys.log

A *message retain period* is held in the system properties table. It defines the period of time that system log entries are kept by MQ Workflow. When this period has elapsed, the contents of the system log are deleted during the next startup of the system. You can get the value of the *message retain period* by using the **Information** command, as explained in “Information about system log properties” on page 27.

[4] In the **Event Viewer - Application Log** window, on the menu bar, select **Log**. Verify that you are viewing the application log and not the system log, by checking that **Application** is selected.

Using the error log

The error log provides further information that can be used for problem determination. Error information is written to the error log whenever a component within an MQ Workflow system encounters a problem that severely affects its operation or causes it to fail.

As with the system log, the error log is maintained in the MQ Workflow database, the layout of which is shown in “Error log record entries” on page 75. The administration utility can be used to access the error log from a remote machine if the administration utility and MQ Workflow Server are running on different machines.

In addition, key contents of the error log are written to the flat file **fmcerr.log**, which can be viewed with any ASCII editor. To view this file you need local access to the machine. By default, the **fmcerr.log** file is located in the subdirectory **log** of the MQ Workflow installation directory. The location and the name of this file is specified by the key **ErrorLogFile** in the MQ Workflow profile. Changing the value of this key changes the location of the log file **ErrorLogFile**. The value of this key must be a fully qualified file name (for example, c:\mylogs\fmcerr.log). See the *IBM MQSeries Workflow: Installation Guide* for details about the MQ Workflow profile.

An important concept of error reporting are exceptions. Exceptions are internal error events created by one of the components in an MQ Workflow system. Exceptions are mainly raised by the servers, the database layer, the message layer or by the kernel. These exceptions are reported to the administration server and are recorded in the error log.

The following shows how error events recorded in the file **fmcerr.log** may appear:

```
                  MQSeries Workflow 3.2 Error Report

Report creation = 02/15/99 02:50:42 PM
Related message = FMC31020E A database error occurred. SQL code: -998

Error location  = File=fmcdbsvs.sqx, Line=2208,Function=FmcDatabase::Server:...
Error data      = FmcSQLException, Sqlcode=-998, Sqlerrmc=9, Sqlerrp=SQL05000...
```

Figure 13. Sample content of fmcerr.log

An *error retain period* is held in the system properties table. It defines the period of time that error log entries are kept by MQ Workflow. When this period has elapsed, the contents of the error log are deleted during the next startup of the system. You can get the value of the *error retain period* by using the **Information** command, as explained in “Information about error log properties” on page 25.

Additional tools and logs

In addition to the system and error logs previously explained, additional log files and applications provided by MQ Workflow, DB2 (R), and MQSeries (R) can be used to control their status and assist in problem determination:

The MQ Workflow configuration checker
This utility can be used to gain detailed information about the MQ Workflow installation. For details about the configuration checker and how to use it see the *IBM MQSeries Workflow: Installation Guide*.

The DB2 diagnosis log
The DB2 diagnosis log file, **db2diag.log**, is normally located in the MQ Workflow DB2 database instance directory.

The Oracle alert file
The Oracle alert file, **<SID>ALRT.LOG**, is normally located in the bdump subdirectory of your database.

The MQSeries queue manager log
The MQSeries queue manager log files (**amqerr01.log**, **amqerr02.log**, **amqerr03.log**, ...) are normally located in the subdirectory **errors** of the MQSeries queue manager’s installation directory used by MQ Workflow.

Summary of logs

Having read this chapter, you should now be aware that MQ Workflow provides three main logs that can be used for error debugging and problem determination in your MQ Workflow system. The following table lists these logs and provides a summary of important points for each:

Table 1. Summary of logs

	System operational events	**Error information**	**Program instruction tracing**
Log name	System log	Error log	Trace log
Logged in database	System log table	Error log table	no
Logged in flat file (default name)	fmcsys.log	fmcerr.log	fmctrc.log
Logged in NT event log	Application log	no	no
Can be switched off	no	no	yes (default=off)

Table 1. Summary of logs (continued)

Logged on screen (system console)	yes	no	no
Accessible via the administration utility	yes	yes	no

Chapter 4. How to use the system trace facility

System trace is the process of recording statements and instructions that are executed within a system in the sequence in which they occur. The system trace facility records all system events in a trace file. The location of this file is always held locally to the system it is tracing. To view the trace file you must have local access to the traced system. The contents of the trace file can be used to diagnose reproducible problems. You can use any standard text editor to view the trace information that is continuously collected from the running system, and saved in the trace file.

When directed by IBM support personnel to do so, enable tracing by modifying the trace criteria on workstations where you experience reproducible problems.

About the different trace modes

The system trace facility of MQ Workflow provides different trace modes for recording system events. Depending on the trace mode of the MQ Workflow component, the trace records are written to different places. This section describes which trace modes are available, how the different trace modes work, and how you can enable them.

Basically, the MQ Workflow servers use the **extended trace mode** for tracing and the MQ Workflow clients use the **simple trace mode**. The trace records for the MQ Workflow clients are written to a file on a disk, according to the variables you have set. The trace records for the servers are written to memory buffers. The following section describes how the extended trace mode works:

Using the extended trace mode

The extended trace mode is used for all MQ Workflow servers on all platforms. It provides several functions you do not have when using the simple trace mode, for example:

- Level 0 trace entries are always written, even if tracing is switched off
- In case of a program failure, you can find the trace entries in the system dump
- Trace records are written to memory buffers

The MQ Workflow system provides three memory buffers. Each MQ Workflow server that has the trace facility enabled writes its trace records to these memory buffers. You can externalize the memory buffers so that the buffer information is written to the input queue of the **external trace writer**. The external trace writer then writes the content of the queue to a disk as described in the following section.

Using the external trace writer

The external trace writer always writes complete buffer information to a trace file in **wrap-around** mode. **Wrap-around mode** means that the external trace writer writes the buffer information to the specified trace file (also called output file) until it is full, then it switches to the next trace file. You can define up to 16 trace files. When the last trace file is full, the external trace writer starts writing to the first trace file again.

If you do not want to lose the content of a buffer during tracing, you can request to write the content of a memory buffer synchronously to an MQ queue before the system trace switches to the next buffer. This function is called **externalization**. To enable the externalization, you must perform the following steps:

- Start the external trace writer as follows:

  ```
  fmcktrcw [-y Configuration]
  ```

  where *[-y Configuration]* is the identifier of the configuration you want to use.
- Specify the variable *FMC_EXTERNALIZE_TRACE_BUFFERS* by setting its value to YES.

Figure 14 gives you an overview of how the external trace writer works within an MQ Workflow system.

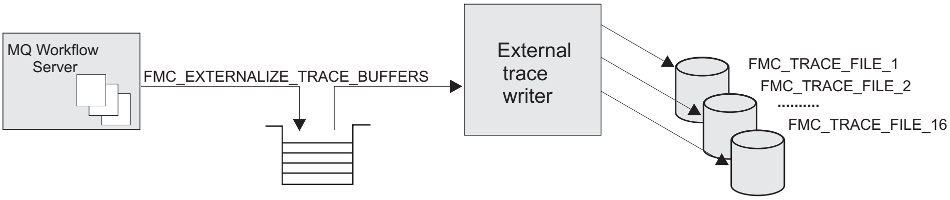


Figure 14. External trace writer in an MQ Workflow system

The MQ Workflow Server writes trace entries to its internal trace buffers. If the externalization of memory buffers is enabled, the MQ Workflow Server writes the trace buffer information to the input queue of the external trace writer. The MQ Workflow server does this every time its internal trace buffer is full.

Then, the external trace writer takes the trace buffer information from the input queue and writes it to the trace files that you have specified with the

FMC_TRACE_FILE_1 to *FMC_TRACE _FILE_16* variable. For details about the available trace variables, refer to "Setting environment variables" on page 48.

The name of the external writer's input queue is defined during the system configuration. The file names, the number, and the size of the external writer's output files can be specified by setting the appropriate environment variables.

In case of a system failure, the buffer information is written to a file, even if externalization is not switched on. The MQ Workflow system determines the name of the trace file by using the values that are set for the simple trace variables *FMC_TRACE_FILE*, *FMC_SPLIT_TRACES*, and *FMC_FLIP_FLOP_TRACES*. See "Enabling traces" on page 44 for details about trace parameter settings.

To stop the external trace writer, enter the following command:

```
fmckstrw [-y Configuration]
```

where [-y Configuration] is the identifier of the configuration you have used.

Using the flip-flop and the split trace mode

The flip-flop and the split trace mode are only active in the simple trace mode. But if an MQ Workflow server fails, the values that you have set for the corresponding variables are also used by the MQ Workflow servers, which operate in the extended trace mode. In case of a server failure, the variables determine the files to which the internal trace buffers are written.

In the simple trace mode, you can use the flip-flop and the split trace mode to reduce the trace file size. In the following section it is described how these trace modes work. If you set the split trace mode, each executable writes into an own trace file. If you set the flip-flop trace mode, each MQ Workflow process writes into an own trace file. With the flip-flop trace mode you can define a file size for the trace file by specifying a value for the parameter *[filesize]*.

If you do not set this parameter, the default trace file size is 5000 KB. When the maximum file size is reached, MQ Workflow automatically closes the trace file and saves it by renaming the file extension to **.bak**. Before renaming the trace file, MQ Workflow checks whether a backup trace file already exists. If this is the case, this old file is deleted first. Then a new trace file is opened so that tracing can continue.

This method ensures that you always have one backup trace file that contains the latest trace information together with the current trace file. It also means that older trace files are lost as soon as a backup file is created.

To enable the split trace mode set the parameter *[split]* to 1. If you also want to set the flip-flop mode, set the parameter *[flipflop]* to 1. The variables can be set in the MQ Workflow configuration or in the operating system environment.

You can only set the flip-flop trace mode when the split trace mode is enabled.

Enabling traces

To enable the appropriate trace mode, you must set the corresponding variables in the configuration or in the system environment.

You can enable tracing by setting the appropriate variables on the machine that is to be traced. You can set the variables in the MQ Workflow configuration or in the operating system environment. When a process is started, it checks whether there are trace variables set in the system environment. If the process does not find any variables in the system environment, it checks the MQ Workflow configuration.

The following table lists the available environment variables showing the default value for each variable and the mode in which you can use the respective variable.

Table 2. Available environment variables for simple and extended tracing

Environment variable	Default value	Comments
FMC_TRACE_CRITERIA	Not set	Used in **simple** or in **extended trace mode**.
FMC_TRACE_FILE	fmctrc	Used in **simple trace mode**. This variable determines the trace file name for the simple trace mode. In case of a server failure, it is also used in the **extended trace mode** because the servers write their internal trace buffer information to this trace file.
FMC_TRACE_FILE_1 to FMC_TRACE_FILE_16	system dependent	Used in **extended trace mode** to control the operation of the external trace writer.
FMC_SPLIT_TRACES	Not set	Used in **simple trace mode**. This variable determines the trace file name for the simple trace mode. In case of a server failure, it is also used in the **extended trace mode** because the servers write their internal trace buffer information to this trace file.

Table 2. Available environment variables for simple and extended tracing (continued)

FMC_FLIP_FLOP_TRACES	Not set	Used in **simple trace mode**. It requires split trace mode. If this variable is set, it determines the trace file name for the simple trace mode. In case of a server failure, it is also used in the **extended trace mode** because the servers write their internal trace buffer information to this trace file.
FMC_EXTERNALIZE_ TRACE_BUFFERS	NO	Used in **extended trace mode**. This variable must be set to ″YES″ to enable the external trace writer.
FMC_TRACE_ BUFFER_SIZE	256 KB	Used in **extended trace mode** to specify the size of the trace buffers.
FMC_TRACE_FILE_ SIZE	5 MB	Used in **simple trace mode** and by the external trace writer.
FMC_REFRESH_COUNT _FOR_TRACE_CRITERIA	50	Used in **extended trace** mode, that is, for the servers only. This variable is used to control the refresh rate of the following variables: *FMC_TRACE_CRITERIA*, *FMC_EXTERNALIZE_ TRACE_BUFFERS*, and *FMC_REFRESH_COUNT_ FOR _TRACE_CRITERIA*.

Using the configuration checker utility for the simple trace mode

To set the appropriate trace parameters for the simple trace mode in the MQ Workflow configuration, you can use the configuration checker utility by specifying the command `fmczchk` and parameters as follows:

```
fmczchk -c trc:level[,file][,split][,flipflop][,filesize]
```

This command invokes the configuration checker which sets the trace parameters in the MQ Workflow configuration with the values you specify.

With this command, you can only specify the *FMC_TRACE_CRITERIA* variable and those that are used in the simple trace mode. Table 2 on page 44 shows which variables are used in the simple or the extended trace mode.

Name of the parameter	**Description**

level	Specifies the tracing level. The configuration checker utility automatically specifies FFFF for categories and for components FFFFFFFF. This setting cannot be modified.
file	Specifies the *FMC_TRACE_FILE* variable that is used to control the trace file name in the simple trace mode.
split	Specifies whether the split trace mode is enabled or not. The mode is enabled by setting the value to 1. Setting any other value deactivates it.
flipflop	Specifies whether the flip-flop mode is enabled or not. To enable this mode, the split trace mode must also be active and the value must be set to 1.
filesize	Specifies the size of the trace file in megabytes (MB). This parameter is only active if the flip-flop mode is set. It is used in the simple trace mode and by the external trace writer in the extended trace mode. If the flip-flop mode is set and the file size is not specified, the default file size is 5 MB.

This command and its options are also explained in the *IBM MQSeries Workflow: Installation Guide*.

To enable tracing, specify a trace level in the range of one to three. If required, you can also specify a drive and path for the trace file, a trace file name, the trace mode, and the trace file size (when using the flip-flop trace mode). The following examples show the syntax:

The drive and path name given in the examples are optional. Use the drive and path name according to your installation. If you do not specify a drive and path name for the trace file, it is written to the current drive of the traced executable.

```
fmczchk -c trc:1
```

enables tracing with the trace level one.

```
fmczchk -c trc:1,x:\tmp\traces\mytrace
```

additionally specifies a path and file name different from the default.

If you specify the file name information, make sure not to add an extension to the trace file name. MQ Workflow appends the appropriate file extension, depending on the trace mode: If the split trace mode is set, the name of the traced executable is appended. If the flip-flop trace mode is active, too, the number of the traced process is added to the name of the executable. Without a trace mode, the file extension is .LOG.

```
fmczchk -c trc:1,x:\tmp\traces\mytrace,1,1,5000
```

additionally enables the split and the flip-flop trace mode and specifies the size of the trace file to 5000 KB.

To start the trace, do the following:

1. Restart MQ Workflow.
2. Reproduce the problem.
3. To view the result of the trace, do the following:
   a. Shut down MQ Workflow.
   b. Use a standard text editor to open locally the trace file that you specified for the variable *FMC_TRACE_FILE*. If you did not specify a value for the variable *FMC_TRACE_FILE*, the default FMCTRC.LOG file name is used.
   c. Turn the trace off again by setting the trace level to zero as follows:
      ```
      fmczchk -c trc:0
      ```
      You should do this because MQ Workflow operation is slower when tracing is active.

If necessary, send the trace file to the appropriate IBM support personnel.

Make sure that there is enough disk space available for the trace file. Delete the file when the problem is solved and you do not need it any longer.

Using the configuration checker utility for the extended trace mode

To set the appropriate trace parameters for the extended trace mode in the MQ Workflow configuration, you can use the configuration checker utility by specifying the command `fmczchk` and parameters as follows:

```
fmczchk -c inst:m,VariableName,VariableValue
```

This command invokes the configuration checker which sets the trace parameters in the MQ Workflow configuration with the values you specify.

With this command, you can only specify the *FMC_TRACE_CRITERIA* variable and those that are used in the extended trace mode. Table 2 on page 44 shows which variables are used in the simple or the extended trace mode.

The following examples show you how to set the variables that are used for the extended trace mode. If you want to set the *FMC_TRACE_BUFFER_SIZE* variable to a value other than the default, enter the following command:

```
fmczchk -c inst:m,FMC_TRACE_BUFFER_SIZE,512
```

This command sets the trace buffer size in the MQ Workflow profile to 512 KB.

If you want to set the *FMC_EXTERNALIZE_TRACE_BUFFERS* variable, enter the following command:

```
fmczchk -c inst:m,FMC_EXTERNALIZE_TRACE_BUFFERS,YES
```

This command enables the externalization of trace buffers so that the trace buffer information is written to the input queue of the external trace writer.

If you want to specify a name for the trace files, set the *FMC_TRACE_FILE_1 ... FMC_TRACE_FILE_16* variable as shown in the following example:

```
fmczchk -c inst:m,FMC_TRACE_FILE_7,d:\tracefile7.log
```

With this command, you set the variable *FMC_TRACE_FILE_7* and the trace information is written to the *tracefile7.log* file, which is stored in *d:*

Setting environment variables

Instead of using the configuration checker utility to enable tracing, you can also set variables in the operating system environment of MQ Workflow to control tracing. The following section provides detailed information about each variable that is listed in Table 2 on page 44.

The *FMC_TRACE_CRITERIA* variable is necessary to enable traces. It contains three parameters that define what is to be traced:

FMC_TRACE_CRITERIA

Level The parameter *level* defines the importance of the generated trace points. There are three levels of importance you can set. Level *1* traces only the most important events. Level *2* is an intermediate trace level, and level *3* traces all important events.

Category
The parameter *category* is a bit field where each bit specifies whether a specific category should be traced or not. The bits are numbered from 0 to 15, where bit 0 is the rightmost least significant bit and bit 15 is the leftmost most significant bit. The following table describes the bits 0 to 6. These are the only bits that are currently used.

Bit	Category	Description

0	Control flow	Function entries and exits, and major conditional branches are traced.
1	Error conditions	Error conditions are traced when detected, for example, when a file is not found or the queue manager is not available.
2	System call	Traces before and after a system call is issued.
3	Messages	Traces when messages are created, sent, or received.
4	Context	Traces various context settings used to identify the flow of requests and responses through the system.
5	Performance	Performance-specific events are traced, for example, before and after DB2 is started.
6	General	All trace points that are not covered by one of the other categories are traced.

You can set any combination of these bits. For example, if you want to set the categories control flow, context, and general, you must switch on the bits 0, 4, and 6. Specifying FFFF as category ensures that all bits are switched on, inluding those that are currently not used.

Component

The parameter *component* also is a bit field. Each bit specifies whether a specific MQ Workflow internal component should issue trace entries. The bits are numbered from 0 to 31, where bit 0 is the rightmost least significant bit, and bit 31 is the leftmost most significant bit. Currently only the bits 0 to 26 are used. The following table lists the component names of each used bit.

Bit	Internal component name
	Kernel
1	Communications
2	Messages
3	Database access
4	Tiny object manager
5	Containers
6	Staff resolution
7	Server framework
8	Administration server
9	Execution server

10	Modeling server
11	Scheduling server
12	Cleanup server
13	Distribution server
14	API functions
15	Administration client
16	Modeling client
17	Runtime client
18	Other clients
19	Program execution agent
20	Installation
21	ActiveX controls
22	Test
23	Gateway server
24	Migration
25	Cleanup audit trail
26	Configuration

As for the category, you can set any combination of these bits. Specifying FFFFFFFF as component ensures that all components issue trace entries. To specify the parameters *category* and *component*, enter the following: set FMC_TRACE_CRITERIA = 3, FFFF, FFFFFFFF, where 3 represents the trace level[5].

The parameters *category* and *component* can only be specified in the operating system environment, you cannot specify them with the configuration checker utility.

FMC_TRACE_FILE

If you use the simple trace mode, this optional variable specifies the name of the file to which the trace information is written. You only need to set this variable in the MQ Workflow configuration on the server machine if you want to use a trace file name other than the default.

[5] If you have a UNIX platform, enter the command ″export″ instead of ″set″, or ″setenv″, depending on the command shell you use.

[6] Do not use the file names **fmcerr.log** or **fmcsys.log** for the environment variable *FMC_TRACE_FILE*. These names are reserved for the error and system log, respectively.

If you specify only a file name[6], the file is saved in the current directory where MQ Workflow is started. You can also specify a different directory and path. For example, if you want to save trace information to the file specified by e:\fmc\trace\mytrace, enter the following command:

```
set FMC_TRACE_FILE=e:\fmc\trace\mytrace
```

In the example, the trace file **mytrace.log** is written to the directory e:\fmc\trace. If you do not specify a file name, the default file name is FMCTRC.LOG. If you do not specify a drive and path name for the trace file, it is written to the current drive of the traced executable.

FMC_TRACE_FILE_1 ... FMC_TRACE_FILE_16

If you use the extended trace mode, this variable specifies the file name to which the trace information is written. You can specify up to 16 trace file names. Note: The trace file names must be consecutively numbered. If you specify trace file 1 to 4, and 7 to 12, for example, the trace information is only written to the files 1, 2, 3, and 4. The names 7 to 12 will be ignored.

In a file system that supports long file names, such as HPFS or NTFS, MQ Workflow can write traces to different trace files. MQ Workflow provides the following environment variables to specify the trace mode:

FMC_SPLIT_TRACES

If you set this optional variable, each executable writes into an own trace file so that, for example, the execution server writes into a different trace file than the administration server or the scheduling server. To have each executable write its own trace file, set the variable *FMC_SPLIT_TRACES* to 1 by entering the following:

```
set FMC_SPLIT_TRACES=1
```

In this case, the *FMC_TRACE_FILE* variable specifies the first part of the name of the file to which the trace information is written. The second part of the trace file name consists of the name of the executable that writes the trace. If you specify only the file name information, the trace file is saved in the directory where MQ Workflow is started. You can also specify a different directory and path, for example:

```
set FMC_TRACE_FILE=FMC\LOGDIRECTORY\TRACE
```

If you specify the file name information, make sure not to add an extension to the trace file name. MQ Workflow appends the appropriate file extension, depending on the trace mode: If the split trace mode is set, the name of the traced executable is appended. If the flip-flop trace mode is active, too, the number of the traced process is added to the name of the executable. Without a trace mode, the file extension is .LOG. If you do not specify the file-name information, the default is FMCTRC (plus the name of the executable that writes the trace file). So, for example, the default trace file name for the executable **FMCEMAIN** is `FMCTRC.FMCEMAIN`.

FMC_FLIP_FLOP_TRACES

When you set the flip-flop trace mode, each execution server process writes into an own trace file. The flip-flop trace mode can only be set when the split trace mode is set, too. To have each process write its own trace file, set the variable *FMC_FLIP_FLOP_TRACES* to 1 by entering the following:

```
set FMC_FLIP_FLOP_TRACES=1
```

For detailed information about the flip-flop and the split trace mode, refer to "Using the flip-flop and the split trace mode" on page 43.

FMC_EXTERNALIZE_TRACE_BUFFERS

If the extended trace mode is active (the trace records are written to memory buffers) and you want to write the content of a full buffer to an MQ queue, you must set the variable *FMC_TRACE_BUFFERS* to ″YES″ by entering:

```
set FMC_TRACE_BUFFERS=YES
```

As a default, this variable is set to ″NO″.

FMC_TRACE_BUFFER_SIZE

This variable specifies the size of the trace buffer in kilobytes (KB) when using the extended trace mode. If you want to change the default buffer size to 378 KB, for example, enter the following command:

```
set FMC_TRACE_BUFFER_SIZE=378.
```

Default size:
256 KB

Minimum size:
64 KB

Maximum size:
4 MB

FMC_TRACE_FILE_SIZE

This variable specifies the size of the trace file in megabytes (MB). It is only active if the *FMC_FLIP_FLOP_TRACE* variable is set. It is used in the simple trace mode and by the external trace writer in the extended trace mode. If the specified file size is exceeded, the external trace writer switches to the next trace file, according to the values you have set for the trace file. The default value for the maximum trace file size is 5 MB.

FMC_REFRESH_COUNT_FOR_TRACE_CRITERIA

This variable specifies the number of transactions that the server performs before any changes of the trace criteria settings become active. The default value for this variable is 50. If you use the default value, the MQ Workflow servers read the profile settings and update any changes after every 50 transactions. If you want to disable the refresh function, set the variable to 0.

Chapter 5. How to use the audit trail

This chapter describes the audit trail and the command-line application used to clean up records from the audit trail held in the MQ Workflow database.

What is the audit trail?

When a process instance is executed, MQ Workflow writes information about each significant event into an audit trail. The audit trail is managed in the MQ Workflow relational database.

Whether an audit trail is written at all, and if so, how much is written into the audit trail, is controlled by the ***AUDIT_TO_DB*** or the **AUDIT_TO_MQ** option of the process instance. The **AUDIT_TO_DB** option replaces the previously used **AUDIT** option. With the **AUDIT_TO_MQ** option, you can write audit events in XML message format to an MQ queue.

For details about the XML message interface, see the *IBM MQSeries Workflow: Programming Guide* and refer to the part on *The XML message interface*.

Note: The previously used **AUDIT** option is still valid.

The audit options are set using Buildtime or an Flow Definition Language (FDL) and can take any one of the following values:

Audit trail No
Specifies that an audit trail is not created when an instance of the process runs.

Audit trail Condensed
Specifies that only important event information, for example, who issued terminate and resume requests for an activity instance, is written to the audit trail.

Audit trail Full
Specifies that all audit information for an event is written to the audit trail.

Audit trail Filter
Specifies which audit event information is written. You can specify these events by entering a string of audit event numbers as listed in Table 4 on page 59.

For further information regarding the ***AUDIT*** options and how to set them, refer to the *IBM MQSeries Workflow: Getting Started with Buildtime* book.

Audit trail specifications are inherited from the domain level of MQ Workflow through the system group to the system and down to the process template. Each specification can be overwritten on a lower level.

There is a deviation from FlowMark Version 2: An audit trail will only be written for processes imported from Version 2.3 and previous versions, if AUDIT is set within the MQ Workflow Definition Language (FDL).

The events are written into the audit trail of the MQ Workflow system on which the process instance was started.

Process instances are identified by the process instance name or the process instance identifier. Both are written into the audit trail. Object identifiers are stored in their external character format.

With MQ Workflow you can use the DB2 or Oracle relational database. Applications that use standard SQL can be written to access the audit trail. Care must be taken to avoid any unintentional changes to the audit trail.

Each audit trail record is associated with a timestamp. This timestamp reflects the date and time the audit trail record was written. As such, it is filled by the underlying relational database management system. Since it is not guaranteed that all timestamps are unique, the sequence in which audit trail records with the same timestamp are retrieved is random.

Depending on the database you use, the audit trail table has different names.

For DB2:
If you use DB2, the name of the audit trail table is **fmc.audit_trail**.

For Oracle:
If you use Oracle, the name of the audit trail table is **<schema name>.audit_trail**.

Table 3 on page 57 shows the structure of the audit trail in the relational database.

Table 3. Audit trail record layout

Field Name	Column name of database table	Type	Explanation
Timestamp (DB2)	CREATED	TIMESTAMP Mandatory	Date and time the audit trail record is written.
Timestamp (Oracle)	CREATED	TIMESTAMP _WF Mandatory	Date and time the audit trail record is written.
Event	EVENT	INTEGER Mandatory	Type of event as indicated in Table 4 on page 59.
Process Name	PROCESS_NAME	VARCHAR (63) Mandatory	Name of the process instance.
Process Identifier	PROCESS_ID	IDENTIFIER Mandatory	Object identifier of the process instance.
Top-level Name	TOP_LVL_PROC_ NAME	VARCHAR (63) Mandatory	Name of the top-level process instance if the process instance is executing as subprocess, or the same as in process name if the process instance is a top-level process instance.
Top-level Identifier	TOP_LVL_PROC_ID	IDENTIFIER Mandatory	Object identifier of the top-level process instance if the process is executing as subprocess, or the same as in process identifier if the process instance is a top-level process instance.
Parent Process Name	PARENT_PROC_NAME	VARCHAR (63) Optional	Name of the parent process instance if the process instance is executing as a subprocess.
Parent Process Identifier	PARENT_PROC_ID	IDENTIFIER Optional	Object identifier of the parent process instance if the process instance is executing as a subprocess.
Process Model Name	PROC_TEMPL_NAME	VARCHAR(32) Mandatory	Name of the process model.
Process Model Valid from Date	TEMPL_VALID_FROM	TIMESTAMP Optional	The date the associated process model becomes valid.
Block Names	BLOCK_NAMES	VARCHAR(254) Optional	The concatenated names of all blocks in which the activity is contained in. The various names are separated by a dot.
User ID	USER_NAME	VARCHAR(32) Optional	ID of the user associated with the event that caused the audit trail to be written. If the audit trail record is written by the MQ Workflow system, this field is not filled.

Table 3. Audit trail record layout (continued)

Field Name	Column name of database table	Type	Explanation
Second user ID	SECOND_USER_NAME	VARCHAR(32) Optional	ID of the second user associated with the event that caused the audit trail to be written.
Activity Name	ACTIVITY_NAME	VARCHAR(32) Optional	If the audit trail entry is associated with an activity, the field contains the name of the activity. If the audit trail entry is associated with a control connector, the field contains the name of the activity that is the source of the control connector.
Activity Type	ACTIVITY_TYPE	INTEGER Optional	If the audit trail record is written for an activity, the field contains the type of the activity as defined in Table 5 on page 62.
Activity Status	ACTIVITY_STATE	INTEGER Optional	If the audit trail record is written for an event associated with an activity, the field contains the status of the activity encoded as shown in Table 6 on page 63.
Second Activity Name	SECOND_ACT_NAME	VARCHAR(32) Optional	If the audit trail is written for an event associated with a control connector, the field contains the name of the target activity.
Command Parameters	COMMAND_ PARAMETERS	VARCHAR(1024) Optional	If the event is the start of a program activity, the field contains the actual parameters passed when invoking the program.
Associated Object	ASSOCIATED_OBJECT	IDENTIFIER Optional	Contains the identifier of the object associated with the event. Can be used to locate the object in the MQ Workflow database.
Object Description	OBJECT_DESCRIPTION	VARCHAR(254) Optional	Contains the description of the object associated with the event.
Program Name	PROGRAM_NAME	VARCHAR(32) Optional	If the event is the start of a program activity, the field contains the name of the program.
Activity Return Code	ACTIVITY_RC	LONG Optional	Return code of the activity.

The contents of each audit trail record depends on the event. Table 4 on page 59 shows the contents of each field.

The audit trail level field indicates which events are written to the audit trail when either full or condensed audit trailing is active. If full audit trailing is active, all audit trail records are written. If condensed audit trailing is active, only the events listed with C are written.

Table 4. Audit trail record contents

Code	Audit trail level	User ID	Second user ID	Associated object	Event
21000	C	Process starter		Process instance	Process started.
21001		Issuer of suspend command		Process instance	Process suspended.
21002		Issuer of resume command		Process instance	Process resumed.
21003				Process instance	Process notification sent.
21004	C			Process instance	Process ended normally.
21005	C			Process instance	Process terminated.
21006	C			Activity instance	Activity ready.
21007	C	User on whose behalf the activity is started		Activity instance	Activity started.
21008				Activity instance	First activity notification sent.
21009		Target of transfer	Source of transfer	Work item	Work item transferred.
21010			User for whom work item is created	Work item	Work item created.
21011	C	User on whose behalf the activity was executed		Activity instance	Activity ended normally.

Table 4. Audit trail record contents (continued)

Code	Audit trail level	User ID	Second user ID	Associated object	Event
21012	C	Issuer of force-finish command		Activity instance	Activity force-finished.
21013		Issuer of restart command		Activity instance	Activity restarted.
21014	C	Issuer of finish command		Activity instance	Activity exited manually.
21015					Block started.
21016					Block ended.
21017		Issuer of create command		Process instance	Process created.
21018	C	Issuer of create and start command		Process instance	Process created and started.
21019		Issuer of restart command		Process instance	Process restarted.
21020		Issuer of delete command		Process instance	Process deleted.
21021	C	Issuer of cancel checkout command	.	Activity instance	Checkout of activity canceled.
21022	C	Issuer of checkout command		Activity instance	Checkout activity.
21023		Issuer of checkin command		Activity instance	Checkin activity.
21024				Activity instance	Second notification for activity sent.
21025	C			Process instance	Process ended normally and deleted.

Table 4. Audit trail record contents (continued)

Code	Audit trail level	User ID	Second user ID	Associated object	Event
21026	C	Issuer of terminate command		Process instance	Process terminated and deleted.
21027	C	Issuer of terminate command		Activity Instance	Activity terminated.
21028	C	Issuer of create with start time command		Process instance	Process created with future start time.
21030		Issuer of delete work item command	Owner of work item	Work item	Work item deleted.
21031	C	Issuer of force restart work item command		Activity instance	Activity force restarted.
21032		User on whose behalf the activity was executed		Activity instance	Activity implementation completed.
21034					Control connector evaluated to true.
21037	C	Issuer of suspend command		Process instance	The specified user has issued a suspend process command.
21038	C	Issuer of terminate process command		Process instance	The specified user has issued a terminate process command.
21039	C	Issuer of execute command		Process instance	The specified user has issued an execute command.
21040	C	Issuer of resume command		Process instance	The specified user has issued a resume process command.

Table 4. Audit trail record contents (continued)

Code	Audit trail level	User ID	Second user ID	Associated object	Event
21041		User who has processed the activity		Activity instance	Activity automatically restarted as exit condition evaluated to false.
21044	C	Issuer of terminate activity request		Activity instance	The specified user has issued a terminate process activity command.
21052	C	Issuer of import request		Process instance	Process instance imported.
21053	C	Issuer of import request		Activity instance	Activity instance imported.
21056				Process instance	Block ended and loop back to the beginning because the exit condition failed.
21080	C			Activity instance	Activity state is set to inError as the activity implementation failed.
21081	C			Activity instance	Activity expired.

The following table shows the encoding for activity types.

Table 5. Audit trail activity type encoding

Code	Activity Type
21100	Program activity
21101	Process activity
21102	Block activity
21103	Information activity
21104	Bundle activity

The following table shows the encoding for activity states. If an event is associated with a state change, the target state is recorded in the audit trail record.

Table 6. Audit trail activity state encoding

Code	Activity State
21200	Ready
21201	Running
21202	Finished
21203	CheckedOut
21204	Force-Finished
21205	Terminated
21206	Suspended
21207	InError
21208	Executed
21209	Skipped
21210	Deleted
21211	Suspending
21212	Terminating
21213	Expired

Analyzing the audit trail

All information about the audit trail is stored in the relational database table. For DB2, this table is called **fmc.audit_trail**. If you use Oracle, the database table is called **<schema name>.audit_trail**. For a description of the audit trail structure, refer to Table 3 on page 57.

There are several SQL queries you can perform to analyze the audit trail. The following section provides examples for both databases, DB2 and Oracle. To perform these queries, you need a database client that can access the MQ Workflow database as, for example, the system administrator.

Using DB2:

To establish a connection with the MQ Workflow database, open a database command prompt and enter the following command:

```
connect to name
```

where *name* is the name of your database.

Using Oracle:

To establish a connection with the MQ Workflow database, open a database command prompt and enter the following command:

```
sqlplus userid/password @ name
```

where *name* is the name of your database and *userid/password* the user ID and its corresponding password of a database user with the necessary access authority to the database.

When you have established the database connection, you can run any query for the specified database to get the information you need. Following you find several examples for queries that can be helpful for analyzing the audit trail.

The following query provides information about which process models have been started and how often they have been started:

For DB2:

```
select proc_templ_name, count(*) from fmc.audit_trail where
event in (21000,21018) group by proc_templ_name order by
count(*)
```

For Oracle:

```
select proc_templ_name, count(*) from <schema name>.audit_trail
where event in (21000,21018) group by proc_templ_name order by
count(*)
```

The following query provides information about which programs have been executed and how often they have been executed.

For DB2:

```
select  program_name, count(*) from fmc.audit_trail where event
= 21007 and program_name is not null group by program_name order
by count(*)
```

For Oracle:

```
select  program_name, count(*) from <schema name>.audit_trail
where event = 21007 and program_name is not null group by
program_name order by count(*)
```

The following query provides information about which activities have been terminated and how often they have been terminated.

For DB2:

```
select activity_name, count(*) from fmc.audit_trail where event
= 21044 group by activity_name order by count(*)
```

For Oracle:

```
select activity_name, count(*) from <schema name>.audit_trail
where event = 21044 group by activity_name order by count(*)
```

The following query provides information about the number of work items per activity created within a specific time frame. In the example, the query returns all work items that were created in January 2000.

For DB2:

```
select activity_name, count(*) from fmc.audit_trail where
event=21010 and created between '2000-01-01-00.00.00.000000' and
'2000-01-31-23.59.59.999999' group by activity_name order by
count(*)
```

For Oracle:

```
select activity_name, count(*) from <schema name>.audit_trail
where event=21010 and timestamp_wf.as_string(created) between
'2000-01-01-00.00.00.000000' and '2000-01-31-23.59.59.999999'
group by activity_name order by count(*)
```

The following query requires full audit trail. It provides information about which blocks have been restarted because their exit condition failed. It also informs you about how often the blocks have been restarted.

For DB2:

```
select proc_templ_name, block_names, count(*) from
fmc.audit_trail where event = 21056 group by proc_templ_name,
block_names order by count(*)
```

For Oracle:

```
select proc_templ_name, block_names, count(*) from <schema
name>.audit_trail where event = 21056 group by proc_templ_name,
block_names order by count(*)
```

Using the cleanup audit trail utility

The cleanup audit trail utility is a command-line tool that you can use to delete records from the audit trail table that you do not need any longer. You can delete all records that:

- Are older than a specified date
- Belong to processes that finished prior to a specified date

It is recommended that you run the cleanup audit trail utility regularly, according to your business needs. Running the utility regularly helps you to keep the size of the audit trail table at a manageable size.

The following describes the **fmcsclad** command used to start the cleanup audit trail utility and the command-line options that are available with it. To delete records from the audit trail table, open a command prompt window and enter the **fmcsclad** command followed by the appropriate options as follows:

```
►►—fmcsclad——————————————————————————————————————————————————————►
```

```
   ┌─default configuration identifier from general configuration profile─┐
►──┼─────────────────────────────────────────────────────────────────────┼──────────►
   └─/y──configurationidentifier─────────────────────────────────────────┘

►──┬──────────────┬──┬────────────────┬──┬─────────────────┬──┬───────────────────┬──►◄
   └─/u──userid───┘  └─/p──password───┘  └─/d──timestamp───┘  └─/x──deleteoption──┘
```

For AIX (R), HP-UX, and Sun Solaris: Use minus (-) instead of a slash (/) as delimiter and lowercase letters.

The switches that precede values are used to indicate the meaning of the value that follows (for example, the /p switch indicates that the value that follows is the *password*). By using these switches you can specify values and options in any order you like.

/y *configurationidentifier*

If you do not specify a value for *configurationidentifier*, the default value is read from the general configuration profile. Each MQ Workflow configuration is identified by a configuration identifier that is set during configuration. The *configurationidentifier* is used to access the MQ Workflow profile which holds the values for system group, system, and database. For details regarding default values and the MQ Workflow profile refer to the *IBM MQSeries Workflow: Installation Guide*.

/u *userid*

For Windows NT, 2000, if you do not specify a user ID, the value you used to log on to the operating system with is taken instead.

For operating systems other than Windows NT, 2000, if you do not specify a user ID, you are prompted to enter a value.

The user ID that you specify must have the following privileges:

For DB2:

- **CONNECT** privilege on the MQ Workflow database.
- **EXECUTE** privilege on the package FMC.FMCDBAD2.

For Oracle:

The user ID that you specify must have the following privileges:

- **CREATE SESSION** on the MQ Workflow database.
- **GRANT ALL ON** <schema name>.audit_trail for this table.

For details about obtaining these privileges, refer to the documentation of the underlying database management system.

/p *password*

If you do not specify a *password*, you are prompted to enter a value unless you are running the cleanup audit trail utility under Windows NT and unified logon has been set in the MQ Workflow database for the selected system. Unified logon means that when NT client users have logged on to Windows NT with their password, there is no need to further log on to individual applications. MQ Workflow supports unified logon when it is specified in the system properties table during MQ Workflow Buildtime. Refer to the *IBM MQSeries Workflow: Getting Started with Buildtime* book for details about setting unified logon.

/d *timestamp*

If you do not specify a value for *timestamp*, you are prompted to enter a value in the following format:

```
YYYY-MM-DD[-HH.MM.SS]
```

That is, you have to enter a 4-digit year followed by a minus sign, then a 2-digit month followed by a minus sign, and then a 2-digit day.

Optionally the date can be followed by a time. If you want to specify a time, you have to enter a minus sign, then a 2-digit hour followed by a period (.), then a 2-digit minute followed by a period (.), and then a 2-digit second.

/x *deleteoption*

If you do not specify a value for *deleteoption*, you are prompted to enter one of the following values:

all If you want to delete all records from the audit trail table that are older than the specified time stamp.

finished

If you want to delete all records from the audit trail table that belong to processes that have been finished prior to the specified time stamp. If you use the finished option, it is recommended that you create an index on the audit trail table as follows:

For DB2:

Create an index on the audit trail table **fmc.audit_trail** with the column TOP_LVL_PROC_ID as index key. To enable the index, you must run the DB2 runstats command on the table **fmc.audit_trail** and rebind the package fmc.fmcdbad2.pkg. For details about how to create an index, refer to the DB2 documentation.

For Oracle:

Create an index on the audit trail table **<schema**

name>.audit_trail with the column TOP_LVL_PROC_ID as index key. For details about how to create an index, refer to the documentation of the underlying database management system.

Part 3. Appendixes

Appendix A. Properties and tables

This appendix describes operational system, server, error log, and system log properties that can be listed using the **Information** subcommand as explained in ″Chapter 2. How to administer the MQ Workflow system″. The layout of error log and system log records is also described.

Operational properties

The behavior of an MQ Workflow system is determined by its properties. Property values are defined during Buildtime and are held in various property tables in the MQ Workflow database. To view operational-related settings held in the property tables for the system, servers, error log, and system log, use the **Information** subcommand from the administration utility. Tables 7 to 10 show the operational-related properties that are listed when this subcommand is invoked from the:

- System commands menu
- Cleanup server commands menu
- Scheduling server commands menu
- Execution server commands menu
- Administration server commands menu
- Error log commands menu
- System log commands menu

When the **Information** subcommand is used to list values related to time interval properties, such as the *session expiration check interval*, values displayed have the following string format:

```
P n Y n M n W n D T n H n M n S
```

Where:

P Is a prefix denoting period and is the first character displayed in the time interval string.

n Is an integer variable representing the number of years, months, weeks, days, hours, minutes, and seconds.

Y Indicates that the value preceding it represents the number of years.

M Indicates that the value preceding it represents the number of months.

W Indicates that the value preceding it represents the number of weeks.

D Indicates that the value preceding it represents the number of days.

T Is a prefix that denotes time and precedes hours, minutes, and seconds entries in the time interval string.

H Indicates that the value preceding it represents the number of hours.

M Indicates that the value preceding it represents the number of minutes.

S Indicates that the value preceding it represents the number of seconds.

For example, a value of one week is displayed as `P1W`, one hour is displayed as `PT1H`, and fifteen minutes is displayed as `PT15M`.

Further information regarding system properties is available in the *IBM MQSeries Workflow: Getting Started with Buildtime* book or in the online help for the Buildtime component.

Table 7. Operational system properties

Name	Description
System Qualifier	Identifies the main qualifier for all objects maintained by the system. In particular it is used as the high-level qualifier for the message queues. It must be unique within all MQ Workflow systems that share the same queue manager and all MQ Workflow systems that share the same node.
Operation Mode	Specifies the execution behavior of the servers of an MQ Workflow system.
Session expiration check interval	Specifies the time after which the administration server checks whether sessions have expired or not.
Session expiration time	Specifies how long a session can last until it is terminated by the system.
Unified logon for Windows NT, 2000, clients	Specifies if Windows NT, 2000 clients should be trusted and logon should be performed without having the user supply a user identification and a password. If set to true, the Windows NT, 2000 client is trusted. If set to false, even Windows NT, 2000 clients are non-trusted.
Message layer refresh interval	Specifies the refresh interval for the message layer. This triggers internal processing that helps to clean up internal buffers. Cleaning up the buffers helps to improve performance.
Client message expiration time	Specifies the time period after which a message sent from a server to a client is deleted. It is the responsibility of the message queuing system to perform the deletion.

Table 7. Operational system properties (continued)

Name	Description
Maximum message size	Specifies the maximum message size within the domain that can be handled by the underlying message queuing network. This property is respected (a) when messages are sent between different MQ Workflow systems and (b) when a client is working in passthrough mode. If the message to be processed exceeds the defined limit, appropriate error processing takes place.
Maximum query message size	Specifies the maximum size the message may have that is returned as the result of a user query. If the query result exceeds the specified size, an error message is returned to the user asking for refinement of the query. The specified size must not exceed the size specified via ’maximum message size’ if the query is not answered by the local system.
Message retry limit	Specifies the number of times a failed message is retried.

Table 8. Operational server properties

Cleanup server

Name	Description
Check interval for cleanup server	Specifies the time intervals that the administration server checks whether the cleanup server is still active or not.
Cleanup server start option	Specifies whether the cleanup server should be started during startup of the MQ Workflow system.
Cleanup start time	Specifies when the cleanup server should start automatically.
Cleanup stop time	Specifies when the cleanup server should stop.
Cleanup time	Specifies how long the cleanup server should be deleting process instances. The specified amount of time that the cleanup server deletes process instances is the minimum amount. The cleanup server may be deleting longer than the specified amount of time.
Cleanup delay time	Specifies how long the cleanup server should be idle before checking again whether process instances are to be deleted. This only applies to the situation where the server has deleted all process instances.

Scheduling server

Name	Description

Table 8. Operational server properties (continued)

Check interval for scheduling server	Specifies the time intervals that the administration server checks whether the scheduling server is still active or not.
Scheduling server start option	Specifies whether the scheduling server should be started during startup of the MQ Workflow system.
Scheduling start time	Specifies when the scheduling server should start automatically.
Scheduling stop time	Specifies when the scheduling server should stop. A value must be provided when a value has been specified for the scheduling start time.
Notification check interval	Specifies the time interval after which the scheduling server checks again for notifications.
Delete notification items	Specifies the number of notification items deleted by the scheduling server in one transaction.
Create notification items	Specifies the number of requests to create notification items sent to the execution server in one transaction.

Execution server

Name	Description
Number of initial execution server instances	Defines the number of execution server instances created when the execution server is started.
Check interval for execution server	Specifies the time intervals that the administration server checks whether the execution server is still active or not. If 0 is specified, no checking is performed. If the administration server determines that an execution server instance has stopped, it performs error management, and starts a new execution server instance.
Immediate cleanup	Specifies the cleanup mode for the physical deletion of processes and their associated objects. If immediate cleanup is specified, the execution server deletes the relevant processes and objects. If no immediate cleanup is specified, the deletion is executed by the cleanup server.
Navigation transaction threshold	Specifies a theshold value for navigation transactions. If 0 is specified, the work item updates are performed in a second transaction. Otherwise, the updates are performed synchronously in the first transaction.

Administration server

Name	Description
No entries	No entries

Table 9. Operational error log properties

Name	Description
Error retain period	Specifies how long error information should be kept in the error log.

Table 10. Operational system log properties

Name	Description
Message retain period	Specifies how long message information should be kept in the message log.

Error log record entries

The error log in the MQ Workflow database contains records that are created for every occurrence of an error.

The layout of error log records that are held in the error log is shown in Table 11. You can list all error log records recorded in the error log byusing the **List** subcommand explained in "Listing error log records" on page 26.

If an error log record cannot be created in the error log, for example if the database crashes, the error log record is written to the file fmcerr.log.

Table 11. Error log record entries

Name	Description
Object ID (mandatory)	Uniquely identifies an error report created by the administration server.
Creation time (mandatory)	Contains the time and date the error log entry was created in the database. Set by the administration server.
System name (mandatory)	Contains the system name. Taken from system settings by the administration server.
System group (mandatory)	Contains the system group name of which the system is a member. Taken from system settings by the administration server.
Version number (mandatory)	Identifies the MQ Workflow version number.
Release number (mandatory)	Identifies the MQ Workflow release number.
Modification level number (mandatory)	Identifies the MQ Workflow modification level number.
Service pack (optional)	Identifies the MQ Workflow service pack level.
Component ID (mandatory)	Identifies the MQ Workflow component that reports the error.

Table 11. Error log record entries (continued)

Component name (mandatory)	Identifies the MQ Workflow component that is related to, or the cause of this error.
Platform (mandatory)	Operating system platform[7] on which the reporting MQ Workflow component is executing: **1** OS/2 (R) **2** AIX **3** Win 3.1 **4** Win NT, 2000 **5** Win 98, Me **6** HP **7** SUN **8** MVS (TM) **9** AS/400 (R)
Exception type (optional)	The value of the thrown MQ Workflow exception.
System log ID (mandatory)	Identifier of the corresponding system log message entry. Refer to "Appendix B. System messages" on page 79.
Message type (optional)	Identifier of the MQ Workflow message related to this event.
Detection time (mandatory)	Contains the date and time the error was detected.
Error file (optional)	Identifies the file within an MQ Workflow component where the error was detected.
Error line (optional)	Identifies the line in the file within an MQ Workflow component where the error was detected.
Error description (mandatory)	Summary of the error in the form of an exception notice.

System log record entries

The system log contains records created for all system events sent or received by the administration server.

The layout of system log records held in the system log is shown in Table 12 on page 77. You can list all system log records recorded in the system log using the **List** subcommand explained in "Listing system log records" on page 27

[7] Not all these platforms are currently supported.

page 27.

Table 12. System log record entries

Name	Description
Object ID (mandatory)	Uniquely identifies each system log entry.
Creation time (mandatory)	Specifies the time when the system log entry was created in the database.
System name (mandatory)	Contains the system name.
System group (mandatory)	Contains the system group name of which the system is a member.
Message number (mandatory)	Identifier of the MQSeries Workflow system log message. It is coded description of the event that occurred. A list of these messages is provided in "Appendix B. System messages" on page 79.
Severity (mandatory)	Identifies the severity associated with the message: **0 (I)** Information **4 (W)** Warning **8 (E)** Error For error events, the administration server also creates an entry in the error log.
Message class (mandatory)	Identifies the class to which this message belongs: **SERVER** Server events **USER** User events **SYSTEM** System events **PEA** Program execution agent events **MQ** Message queuing events **DB** Database events
Event identifier (optional)	A string used to group the entries into related events. This could be a server name, a session ID, or a user defined string.
Message type (optional)	Identifier of the MQSeries Workflow message related to this event.
Error flag (mandatory)	If true, it indicates that error information has been written to the error log.
Server kind (optional)	Identifies the server type for which the system log entry was created if the message belongs to the class SERVER. Otherwise it contains NULL.

Table 12. System log record entries (continued)

Server state (optional)	Identifies the server's state when the system log entry was created if the message belongs to the class SERVER. Otherwise it contains NULL.
UserID (optional)	If set, the name of the user whose request caused the syslog entry.
Exception (optional)	Contains the enumeration value for the exception thrown by an MQ Workflow component. See IBM for details.
Message parameters (optional)	Contains additional parameters which are used to replace real message text. See IBM for details.

Appendix B. System messages

The system log is a record of all relevant system events. It records all events reported by servers, the message layer, the database layer, and the operating system. The administration server logs the following events in the system log:

- Server state changes
- Message layer report messages
- Database layer report messages
- All error events

You can view the contents of the system log by selecting the list function as described in “Listing system log records” on page 27.

For every system event, a system log record is created. The meaning of the entries in each system log record is described in Table 12 on page 77. For each record within the system log a message number exists. This number is a coded description of the event that occurred. These message numbers also appear on the screen of your workstation running the administration utility as part of an unsolicited message.

Additionally, for Windows NT, the administration server writes important system events to:

- The **application log** of Windows NT
- The flat file **fmcsys.log**
- The **administration utilities** for all logged-on administration users

Refer to “Using the system log” on page 35 for details.

Message list

The following explains the meaning of each message number recorded in the system log. For each, the message number, text, severity, ID, and an explanation is given. Where appropriate, the user action and the destination are also given.

10000 **System startup complete. System %1$s in system group %2$s is now running.**

Severity: Information

MsgID: FMC_INFO_SYSTEM_STARTED

Explanation: The administration server completed the system startup processing.

User Response: None.

Destination: System Log

10010 System starting.

Severity: Information

MsgID: FMC_INFO_SYSTEM_STARTING

Explanation: The administration server is performing startup processing as defined in the domain, system group, and system settings for this system.

User Response: None.

Destination: System Log

10020 System %1$s in system group %2$s stopped.

Severity: Information

MsgID: FMC_INFO_SYSTEM_STOPPED

Explanation: The administration server has completely and successfully shut down the system.

User Response: None.

Destination: System Log

10030 System is being shutdown.

Severity: Warning

MsgID: FMC_INFO_SYSTEM_STOPPING

Explanation: The administration server is about to shut down the system. This includes stopping all running servers and program execution agents, cleaning up internal tables, and deleting all sessions records.

User Response: None.

Destination: System Log

10040 System is being shut down. The system shutdown will complete once all running servers have stopped.

Severity: Warning

MsgID: FMC_INFO_SYSTEM_STOPPING_DEFERRED

Explanation: The administration server is about to shut down the system. This includes stopping all running servers and program execution agents, cleaning up internal tables, and deleting all session records. The system shutdown is completed when all running servers have confirmed that they stopped. The administration server waits the time specified as check interval for each server to complete (FDL keyword CHECK_INTERVAL).

User Response: You should observe the system as it shuts down and take corrective actions if the shutdown is unsuccessful. If you are sure that no more servers are running, you can enforce the shutdown by requesting the system shutdown again. Also, in this case, stop requests are send to each program execution agent and all session records are deleted.

Destination: System Log

10100 Administration server starting.

Severity: Information

MsgID: FMC_INFO_ADMIN_SVR_STARTING

Explanation: The administration server for this system is starting.

User Response: None.

Destination: System Log

10110 Administration server for system %1$s started.

Severity: Information

MsgID: FMC_INFO_ADMIN_SVR_STARTED

Explanation: The administration server for this system has started and is operational.

User Response: None.

Destination: System Log

10120 Administration server stopping.

Severity: Information

MsgID: FMC_INFO_ADMIN_SVR_STOPPING

Explanation: The administration server is stopping. It now begins with the system shutdown.

User Response: None.

Destination: System Log

10130 **Administration server for system %1$s stopped.**

Severity: Information

MsgID: FMC_INFO_ADMIN_SVR_STOPPED

Explanation: The administration server stopped.

Destination: System Log

User Response: None.

10200 **Execution server for system %1$s started.**

Severity: Information

MsgID: FMC_INFO_EXE_SVR_STARTED

Explanation: The execution server is started.

User Response: None.

Destination: System Log

10210 **Execution server for system %1$s stopped.**

Severity: Information

MsgID: FMC_INFO_EXE_SVR_STOPPED

Explanation: All instances of the execution server have been stopped by the administration server.

User Response: None.

Destination: System Log

10220 **The execution server for system %1$s could not be started.**

Severity: Error

MsgID: FMC_ERROR_EXE_SVR_START

Explanation: The execution server could not be started by the administration server. The administration server has sent a start request to the MQSeries trigger monitor and received no confirmation from the execution server that it started.

User Response: Check if the MQSeries trigger monitor is running.

Destination: System Log

10300 **Cleanup server for system group %2$s started in system %1$s.**

Severity: Information

MsgID: FMC_INFO_CLEANUP_SVR_STARTED

Explanation: The administration server has started the cleanup server.

User Response: None.

Destination: System Log

10310 **Cleanup server for system group %1$s stopped.**

Severity: Information

MsgID: FMC_INFO_CLEANUP_SVR_STOPPED

Explanation: The administration server has stopped the cleanup server.

User Response: None.

Destination: System Log

10320 **The cleanup server for system group %2$s could not be started in system %1$s.**

Severity: Error

MsgID: FMC_ERROR_CLEANUP_SVR_START

Explanation: The cleanup server could not be started by the administration server. The administration server has sent a start request to the MQSeries trigger monitor and received no confirmation from the execution server that it started.

User Response: None.

Destination: System Log

10400 **Scheduling server for system group %2$s started in system %1$s.**

Severity: Information

MsgID: FMC_INFO_SCHED_SVR_STARTED

Explanation: The administration server has started the scheduling server.

User Response: None.

Destination: System Log

10410 **Scheduling server for system group %1$s stopped.**

Severity: Information

MsgID: FMC_INFO_SCHED_SVR_STOPPED

Explanation: The administration server has stopped the scheduling server.

User Response: None.

Destination: System Log

10420 **The scheduling server for system group %2$s could not be started in system %1$s.**

Severity: Error

MsgID: FMC_ERROR_SCHED_SVR_START

Explanation: The scheduling server could not be started by the administration server. The administration server has sent a start request to the MQSeries trigger monitor and received no confirmation from the execution server that it started.

User Response: None.

Destination: System Log

10500 **Execution server instance started.**

Severity: Information

MsgID: FMC_INFO_EXE_SVR_INST_STARTED

Explanation: The administration server has started another instance of the execution server.

User Response: None.

Destination: System Log

10510 **Execution server instance stopped.**

Severity: Information

MsgID: FMC_INFO_EXE_SVR_INST_STOPPED

Explanation: The administration server has stopped an instances of the execution server.

User Response: None.

Destination: System Log

10520 **Execution server instance could not be started.**

Severity: Error

MsgID: FMC_ERROR_EXE_SVR_INST_START

Explanation: The additional instance of the execution server could not be started. The administration server has sent a start request to the MQSeries trigger monitor and received no confirmation from the execution server that it started.

User Response: None.

Destination: System Log

11000 **Administration server in system %1$s recovering from a previous failure.**

Severity: Warning

MsgID: FMC_WNG_ADMIN_SVR_PREVIOUS_ABEND

Explanation: The administration server has determined from its internal state table that it did not properly shut down before. This could have been caused by a failure of the operating system, the database, the message layer, or by the administration server itself. The administration server now performs a different startup processing by bringing back the system state to the last recorded state.

User Response: You should check the system and error log for more information about the causes of the error.

Destination: System Log

11010 **System %1$s in system group %2$s stopped with the force option.**

Severity: Warning

MsgID: FMC_WNG_FORCED_SYSTEM_STOP

Explanation: The system was stopped with the force option which allows for an emergency shutdown. The administration server has sent shutdown requests to all servers and has then immediately terminated itself after cleanup processing but without control whether outstanding requests were successful.

Note: The administration server also uses a forced shutdown when it receives a shutdown request from the Windows NT operating system.

User Response: You need to check yourself whether all servers stopped successfully.

Destination: System Log

11020 **The trigger monitor for queue manager %1$s is not available.**

Severity: Warning

MsgID: FMC_WNG_TRIGGER_MONITOR_NOT_

AVAILABLE

Explanation: The trigger monitor is required to start MQSeries Workflow servers, a) during system startup, b) with the adminstration utility, c) by exploiting the automatic restart facility of MQSeries Workflow.

User Response: You should start the trigger monitor for the queue manager you have started for the current MQSeries Workflow system. The trigger monitor for the queue manager as defined by the MQSeries Workflow installation can be started with the command from the command line: runmqtrm /M FMCQM

Destination: System Log

12000 **The requested shutdown of system %1$s could not be completed.**

Severity: Error

MsgID: FMC_ERROR_SYSTEM_STOP

Explanation: The administration server could not completely shut down the system. In this case either one or more of the controlled servers have not responded to a shutdown request or the administration server encountered an error while cleaning up its internal tables.

User Response: You should check the system and error log for further information about the causes of the incomplete shutdown. You can force a shutdown by stopping the administration server from the administration utility again.

Destination: System Log

12010 **One or more servers have stopped.**

Severity: Error

MsgID: FMC_ERROR_INCONSISTENT_STATE

Explanation: The administration server has detected during its regular checks that the number of servers that should be available is higher than the number of servers actually running. The administration server is restarting the required servers if this is still possible.

User Response: You need to analyze the error and take corrective actions.

Destination: System Log

12020 **No execution server defined for system %2$ in database.**

Severity: Error

MsgID: FMC_ERROR_EXE_SVR_UNDEFINED

Explanation: The execution server is not defined in the database for this system.

User Response: Define the server in the database. A default execution server can be defined by loading the reference FDL into the database.

Destination: System Log

12030 **No scheduling server defined for system %2$ in database.**

Severity: Error

MsgID: FMC_ERROR_SCHED_SVR_UNDEFINED

Explanation: The scheduling server is not defined in the database.

User Response: Define the server in the database. A default scheduling server can be defined by loading the reference FDL into the database.

Destination: System Log

12040 **No modeling server defined for system %2$ in database.**

Severity: Error

MsgID: FMC_ERROR_MODEL_SVR_UNDEFINED

Explanation: The modeling server is not defined in the database.

User Response: Define the server in the database. A default modeling server can be defined by loading the reference FDL into the database.

Destination: System Log

12050 **No cleanup server defined for system %2$ in database.**

Severity: Error

MsgID: FMC_ERROR_CLNUP_SVR_UNDEFINED

Explanation: The cleanup server is not defined in the database.

User Response: Define the server in the database. A default cleanup server can be defined by loading the reference FDL into the database.

Destination: System Log

12060 **No distribution server defined for system %2$ in database.**

Severity: Error

MsgID: FMC_ERROR_DIST_SVR_UNDEFINED

Explanation: The distribution server is not defined in the database.

User Response: Define the server in the database. A default distribution server can be defined by loading the reference FDL into the database.

Destination: System Log

12070 **No administration server defined for system %2$ in database.**

Severity: Error

MsgID: FMC_ERROR_ADMIN_SVR_UNDEFINED

Explanation: The administration server is not defined in the database.

User Response: The administration server represents the system. Therefore an administration server must be defined for each system. A default administration server can be defined by loading the reference FDL into the database. If this is the first time you access this database, you need to run the bootstrap program fmxzbstr to initialize the database with an administration server definition.

Destination: System Log

12100 **The scheduling server is already started for system group %3$s in system %2$.**

Severity: Error

MsgID: FMC_ERROR_SCHED_SVR_ALREADY_STARTED

Explanation: The scheduling server is already starting or running. Only one instance of the scheduling server can run for one system group.

User Response: None.

Destination: System Log

12110 **The cleanup server is already started for system group %3$s in system %2$.**

Severity: Error

MsgID: FMC_ERROR_CLNUP_SVR_ALREADY_STARTED

Explanation: The cleanup server is already starting or running. Only one instance of the cleanup server can run for one system group.

User Response: None.

Destination: System Log

12120 **The administration server is already started for system %2$s.**

Severity: Error

MsgID: FMC_ERROR_ADMIN_SVR_ALREADY_STARTED

Explanation: The administration server is already starting or running. Only one instance of the administration server can run for one system.

User Response: None.

Destination: System Log

12150 **%1$s messages have been moved to the execution server input queue.**

Severity: Information

MsgID: FMC_INFO_HOLD_QUEUE_MSG_MOVED

Explanation: The number of messages indicated have been moved from the hold queue to the input queue of the execution server and can now be replayed. For this you need, however, a running execution server instance.

User Response: None.

Destination: System log

12160 **Execution server hold queue message %1$s deleted upon request of user %2$s.**

Severity: Information

MsgID: FMC_INFO_HOLD_QUEUE_MSG_DELETED

Explanation: The hold queue messages has been removed from the hold queue.

User Response: None.

Destination: System log

12200 **The completion of the program for activity '%1$s' was ignored.**

Severity: Information

MsgID: FMC_ERROR_PROG_FINISH_IGNORED

Explanation: The execution server was notified that execution of a specific program was finished. However, the activity associated with the program was either not in the running state or not found any more. This can happen when an activity is force-restarted or force-finished while the implementing program is still running. This message is also generated when program execution encountered an error, and the associated activity was not found any more.

User Response: None.

Destination: System log

12210 **Could not execute program of activity %1$s ('%4$s') in process %2$s. The error was: %3$s.**

Severity: Error

MsgID: FMC_ERROR_PROG_ERROR_RECEIVED

Explanation: The execution server was notified that program execution encountered an error when trying to invoke the specified program, which is associated with the named activity in the named process. The associated activity is set to state in error.

User Response: Correct the problem that prevented program execution from working. Then, locate the affected activity and force-restart it. Alternatively, force-finish the activity.

Destination: System Log

12230 **An invalid passthrough was attempted for activity '%1$s'.**

Severity: Information

MsgID: FMC_ERROR_INVALID_PASSTHROUGH

Explanation: An API program implementing an activity tried to issue a passthrough, but the activity associated with the program was either not in the running state or not found any more. This can happen when an activity is force-restarted or force-finished while the implementing program is still running.

User Response: None.

Destination: System Log

12240 **Execution server instance(s) stopped abnormally.**

Severity: Error

MsgID: FMC_ERROR_EXE_SVR_INST_STOPPED

Explanation: One or more instances of the execution server stopped abnormally. The administration server detected during its regular checks that the number of instances that should be available is higher than the number of instances actually running. The administration server is restarting the required instances if this is still possible.

User Response: Analyze the error information provided in the system and error logs. Verify that the instances were successfully restarted.

Destination: System Log

12250 **The execution server for system %1$s stopped abnormally.**

Severity: Error

MsgID: FMC_ERROR_EXE_SVR_STOPPED

Explanation: The execution server stopped abnormally. The administration server detected during its regular checks that the server is no longer running. The administration server is restarting the server if this is still possible.

User Response: Analyze the error information provided in the system and error logs. Verify that the server was succesfully restarted.

Destination: System Log

12260 **The cleanup server for system group %1$s stopped abnormally.**

Severity: Error

MsgID: FMC_ERROR_CLEANUP_SVR_STOPPED

Explanation: The cleanup server stopped abnormally. The administration server detected during its regular checks that the server is not running anymore even though it should run. The administration server is restarting the server if this is still possible.

User Response: Analyze the error information provided in the system and error logs. Verify that the server was successfully restarted.

Destination: System Log

12270 **The scheduling server for system group %1$s stopped abnormally.**

Severity: Error

MsgID: FMC_ERROR_SCHED_SVR_STOPPED

Explanation: The scheduling server stopped abnormally. The administration server has detected during its regular checks that the server is not running any more even though it should run. The administration server is restarting the server if this is still possible.

User Response: Analyze the error information provided in the system and error logs. Verify that the server was successfully restarted.

Destination: System Log

12280 **One or more servers are not responding to control requests.**

Severity: Error

MsgID: FMC_ERROR_SVR_NOT_RESPONDING

Explanation: The administration tried to start or stop a server or server instance and got no response.

User Response: You need to analyze why servers are not responding. Check whether there are problems with the message layer or whether servers do run correctly.

Destination: System Log

12290 **The execution server for system %1$s is not responding to control requests.**

Severity: Error

MsgID: FMC_ERROR_EXE_SVR_NOT_RESPONDING

Explanation: The administration tried to start or stop the execution server or an execution server instance and got no response.

User Response: You need to analyze why servers are not responding. Check whether there are problems with the message layer or with servers that do not run correctly.

Destination: System Log

12300 **The scheduling server for system %1$s is not responding to control requests.**

Severity: Error

MsgID: FMC_ERROR_SCHED_SVR_NOT_RESPONDING

Explanation: The administration server tried to start or stop the scheduling server and got no response.

User Response: You need to analyze why servers are not responding. Check whether there are problems with the message layer or with servers that do not run correctly.

Destination: System Log

12310 **The cleanup server for system %1$s is not responding to control requests.**

Severity: Error

MsgID: FMC_ERROR_CLEANUP_SVR_NOT_RESPONDING

Explanation: The administration tried to start or stop the cleanup server and got no response.

User Response: You need to analyze why servers are not responding. Check whether there are problems with the message layer or with servers that do not run correctly.

Destination: System Log

31000 **A database error occurred. SQL code: %1$s.**

Severity: Error

MsgID: FMC_ERROR_SQL_EXCEPTION

Explanation: The datebase reported an error when trying to access it.

User Response: Analyze and correct the error and restart if required. The content of the sqlda can be found in the error log of MQSeries Workflow. You should check the SQL code, for example, in the information center that comes with DB2.

Destination: System Log

31010 **An error in the message layer occurred. Error code: %1$s, Reason code: %2$s.**

Severity: Error

MsgID: FMC_ERROR_MQ_EXCEPTION

Explanation: The message layer reported an error.

User Response: Analyze and correct the error and restart if required. Additional information can be found in the error log.

Destination: System Log

31020 **The server could not be started.**

Severity: Error

MsgID: FMC_ERROR_INITIALIZATION

Explanation: The server could not start because a nonrecoverable error occurred in its inititialization procedure. This can be caused by one of the following reasons: a) The profile cannot be found; b) the message catalogue cannot be found; c) the database is not available or not set up correctly; d) MQSeries is not started or is not set up correctly; e) the server or the system is not correctly defined in the database or the profile.

User Response: Analyze and correct the error and restart if required. Additional information can be found in the error log.

Destination: System Log

31030 **The queue manager %1$s is not available.**

Severity: Error

MsgID: FMC_ERROR_QMGR_NOT_AVAILABLE

Explanation: The server could not start because it could not connect to the queue manager. In most of the cases the queue manager was not started if this error occurs. If starting the queue manager does not solve the problem, you should check whether you have started the queue manager that is defined in the MQSeries Workflow profile in key FMLConnectName (example: SYS_GRP. SYSTEM,FMCQM). Further hints are provided in the NT event log if you are running under NT and/or the log files written by MQSeries in the installation directory of MQSeries on all platforms.

User Response: None.

Destination: System Log

31040 **The database %1$s is not available for connection.**

Severity: Error

MsgID: FMC_ERROR_DB_NOT_AVAILABLE

Explanation: The server could not start because the database is not available or not set up correctly. MQSeries Workflow connects to the database via MQSeries, which acts as transaction manager. When using Windows NT, 2000, you need to check the event log for both MQSeries and database instance entries. Additionally, you should check the log files written by MQ and the database management system. For all platforms, the log files are placed in the respective installation directory.

User Response: None.

Destination: System Log

31050 **An error has occurred which has terminated processing.**

Severity: Error

MsgID: FMC_ERROR_UNRECOVERABLE_ERROR

Explanation: MQSeries Workflow encountered a problem which caused a server to end processing. Analyze the error log and take corrective actions. Depending on the cause of the problem it can be required to shut down the complete system (for example, table space exhausted in the database) or to correct the problem and restart the affected server.

User Response: None.

Destination: System Log

31060 **The retry limit for message %1$s has exceeded. The message is stored in the execution server hold queue for later processing.**

Severity: Error

MsgID: FMC_ERROR_RETRY_LIMIT_REACHED_PUT_HOLD_QUEUE

Explanation: If a message cannot be processed, MQSeries Workflow retries to process the message up to its internal retry limit. If this limit is exceeded, the message is stored in the hold queue. This capability is only provided for the execution server.

User Response: None.

Destination: System Log

31065 **The retry limit for message %1$s has exceeded. The message cannot be processed and is ignored.**

Severity: Error

MsgID: FMC_ERROR_RETRY_LIMIT_REACHED

Explanation: If a message cannot be processed, MQSeries Workflow retries to process the message up to its internal retry limit. If this limit is exceeded, the message is ignored.

User Response: None.

Destination: System Log

31070 **The current version %1$s of MQSeries Workflow is incompatible with version %2$s as defined in the database for system %3$s.**

Severity: Error

MsgID: FMC_ERROR_INCOMPATIBLE_VERSION

Explanation: The version of MQSeries Workflow defined by version, release, modification level, and service pack number must match the table definitions in the database. If you receive this error you need to upgrade the tables of the database up to the level supported as described in the *IBM MQSeries Workflow: Installation Guide*.

User Response: None.

Destination: System Log

31080 **Database %1$s is not available for distributed transaction processing with queue manager %2$s acting as resource coordinator. SQL code: %3$s.**

Severity: Error

MsgID: FMC_ERROR_SQL_TX_EXCEPTION

Explanation: The database could not be opened for distributed transaction processing.

User Response: Check if MQSeries could not establish an XA protocol with the database. Therefore, check

- The event log (NT only)
- The log files for the database instance in which this database is running
- The MQSeries error logs for this queue manager

This problem can occur if:

- The queue manager was started before the database manager
- The user ID with which MQSeries tried to establish the XA protocol with the database is incorrect or not authorized

If you start MQSeries Workflow as service under Windows NT, 2000, make sure that both MQSeries Workflow and MQSeries run with a user account that has a valid authorization ID for the database. Additionally make sure that you have defined the user ID and password in the queue manager's qm.ini file.

Destination: System Log

31090 The message prefix defined in the profile does not match with the MQ prefix defined in the database for system %1$s.

Severity: Error

MsgID: FMC_ERROR_MISMATCH_MQ_PREFIX

Explanation: The MQ prefix is used for addressing and must be the same in the database as well as in the profile.

User Response: None.

Destination: System Log

31100 The message %1 could not be processed because of a database deadlock or a timeout. The message will be retried.

Severity: Warning

MsgID: FMC_WARNING_DB_DEADLOCK

Explanation: The transaction failed because of a database deadlock or a timeout. MQ Workflow rolls back the transaction and tries to process it again. If the transaction fails more often than determined by the retry limit for messages, the message is saved in the hold queue to be processed again later.

User Response: None.

Destination: System Log

FMC37005E Option 'd' cannot be accepted. Enter date and time using the format YYYY-MM-DD[-HH.MM.SS].

Explanation:

User Response: None.

FMC37006E Option 'x' cannot be accepted. Valid delete options are '%1' and '%2'.

Explanation:

User Response: None.

FMC37009I Deleting all records older than '%1'...

Explanation:

User Response: None.

FMC37010I Deleting all finished processes older than '%1'...

Explanation:

User Response: None.

FMC37011I %1 record(s) deleted from audit trail table.

Explanation:

User Response: None.

Appendix C. Speeding up menu selection from the administration utility

“Chapter 2. How to administer the MQ Workflow system” on page 11 explains how to perform system administration tasks by selecting items from menus within the administration utility. In some cases, many selections from several menu screens have to be made and entered one at a time in order to perform a single task. When performing multiple tasks this can lead to a lengthy process. This lengthy process can be avoided, however, by a simple and quick entry of multiple commands on a single command line. The following describes how to do this.

Shortcuts to menu selection

Multiple commands can be entered sequentially on a single line at any stage within the administration utility. The following example explains this.

Suppose you want to perform the following set of administration routines:

1. Start the cleanup server.
2. Start an additional execution server instance.
3. Wait 50 seconds for the cleanup server and execution server instance to start.
4. Query the system status to verify operation.
5. Exit from the administration utility and return control to the command prompt.

To perform these routines by selecting items from the menus within the administration utility as described in ″Chapter 2. How to administer the MQ Workflow system″, 13 separate entries have to be made and 5 different menu screens scrolled through. By entering all commands together on a single line, these tasks can be done in one step, avoiding the lengthy process of scrolling through multiple menu screens and entering commands one at a time. The list of commands you sequentially enter to perform these routines is:

```
m c u x e + x x s w(50) q x x
```

Enter the commands together in the same sequence as you would if selecting from the menus. Note that the commands must be entered without blanks between each other. The meaning of each entry in this example is:

m Selects the select server menu from the main menu.

c Selects the cleanup server commands menu from the select server menu.

`u` Starts the cleanup server.

`x` Exits from the cleanup server menu and returns to the select server menu.

`e` Selects the execution server commands menu from the select server menu.

`+` Starts an additional executional server instance.

`x` Exits from the execution server menu and returns to the select server menu.

`x` Exits from the select server menu and returns to the main menu.

`s` Selects the system commands menu from the main menu.

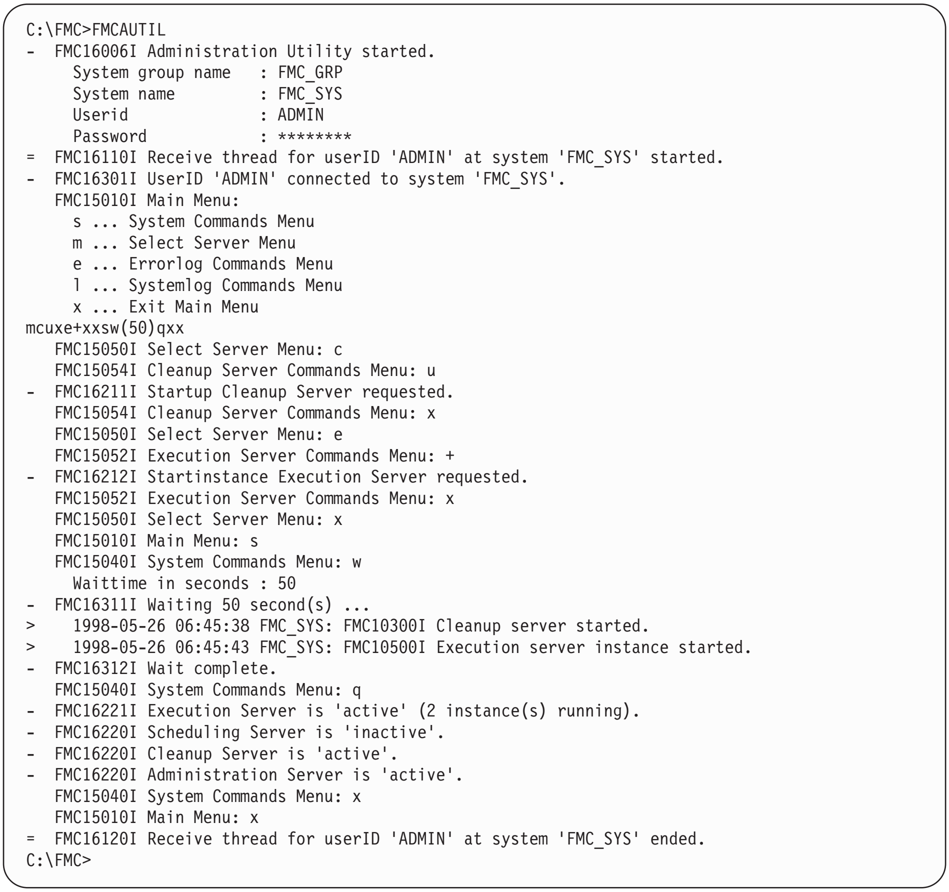
`w(50)` Waits 50 seconds for the cleanup server and execution server instance to start.

`q` Queries the system status to verify the operation of all system servers.

`x` Exits the system commands menu and returns to the main menu.

`x` Exits the administration utility and returns control to the command prompt.

Figure 15 on page 93 shows the actual output you see when you input this sequential list of commands. Within Figure 15 on page 93 the administration utility has been started by entering the FMCAUTIL start command followed by the system group and system names, user ID and password. The sequential list of commands has been entered after the main menu.



```
C:\FMC>FMCAUTIL
-  FMC16006I Administration Utility started.
     System group name   : FMC_GRP
     System name         : FMC_SYS
     Userid              : ADMIN
     Password            : ********
=  FMC16110I Receive thread for userID 'ADMIN' at system 'FMC_SYS' started.
-  FMC16301I UserID 'ADMIN' connected to system 'FMC_SYS'.
   FMC15010I Main Menu:
     s ... System Commands Menu
     m ... Select Server Menu
     e ... Errorlog Commands Menu
     l ... Systemlog Commands Menu
     x ... Exit Main Menu
mcuxe+xxsw(50)qxx
   FMC15050I Select Server Menu: c
   FMC15054I Cleanup Server Commands Menu: u
-  FMC16211I Startup Cleanup Server requested.
   FMC15054I Cleanup Server Commands Menu: x
   FMC15050I Select Server Menu: e
   FMC15052I Execution Server Commands Menu: +
-  FMC16212I Startinstance Execution Server requested.
   FMC15052I Execution Server Commands Menu: x
   FMC15050I Select Server Menu: x
   FMC15010I Main Menu: s
   FMC15040I System Commands Menu: w
     Waittime in seconds : 50
-  FMC16311I Waiting 50 second(s) ...
>    1998-05-26 06:45:38 FMC_SYS: FMC10300I Cleanup server started.
>    1998-05-26 06:45:43 FMC_SYS: FMC10500I Execution server instance started.
-  FMC16312I Wait complete.
   FMC15040I System Commands Menu: q
-  FMC16221I Execution Server is 'active' (2 instance(s) running).
-  FMC16220I Scheduling Server is 'inactive'.
-  FMC16220I Cleanup Server is 'active'.
-  FMC16220I Administration Server is 'active'.
   FMC15040I System Commands Menu: x
   FMC15010I Main Menu: x
=  FMC16120I Receive thread for userID 'ADMIN' at system 'FMC_SYS' ended.
C:\FMC>
```

Figure 15. Output after you enter the sequential list of commands given in the example

When processing has finished, what appears depends on the last command entered. In this case **x**, exit the main menu, is the last command entered. Therefore, when finished, the administration utility is closed and control passes back to the command line.

To distinguish between the different messages that appear and for better readability purposes the messages generated within the administration utility start with a **-** symbol, unsolicited messages sent by the administration server start with a **>** symbol, messages generated by the receive thread start with a **=** symbol, and the menu headings that appear are indented by three characters.

Multiple commands can be entered together at any stage within the administration utility and not only after the main menu, as shown in the previous example. You should note, however, that the order in which you enter commands is important. The correct order is the order in which commands are selected from the administration utility menus.

Appendix D. The record and playback facility

This appendix is divided into three sections that describe the playback and record facilities available for use in the administration utility. These three sections describe the following:

- Creating a record file
- Viewing a record file
- Playing back commands recorded in a record or self-edited file

The playback and record facilities intend to provide a means of automating task processing and avoid re-entering the same tasks again and again.

Creating a record file

A record file is a file that records all the commands you enter and messages and reponses that appear while you are using the administration utility. It also contains the names and codes of the menus you have selected, message and command codes, character abbreviations, and comments. However, unsolicited messages sent by the administration server are not written to the record file.

When the record file is called by the playback option, described later, commands that are recorded in the record file are used to automate task processing. This avoids having to re-enter the same tasks again and again.

A record file is created in the current directory by specifying the **/r** option in the **fmcautil** command line syntax. You can specify a name for the the record file by entering a file name directly after the **/r** option. If no file name is specified, the default **fmcautil.rec** name is used. To create a record file:

1. Open a command prompt window and start the administration utility by entering the **fmcautil** start command followed by the appropriate options as shown in the following syntax diagram. Make sure to specify the **/r** option and include a file name, unless the default **fmcautil.rec** name is to be used.

```
►►──fmcautil───────────────────────────────────────────────────────────────►

   ┌─default configuration identifier from general configuration profile─┐
►──┼─────────────────────────────────────────────────────────────────────┼──►
   └─/y──configurationidentifier─────────────────────────────────────────┘

   ┌─systemgroup from configuration profile─┐
►──┼────────────────────────────────────────┼───────────────────────────────►
   └─/g──systemgroup────────────────────────┘

   ┌─system from configuration profile─┐
►──┼───────────────────────────────────┼──┬─────────────┬──┬───────────────┬──►
   └─/s──system────────────────────────┘  └─/u──userid──┘  └─/p──password──┘

►──┬────────────────────────┬──────────────────────────────────────────────►◄
   │      ┌─fmcautil.rec─┐  │
   └─/r───┼──────────────┼──┘
          └─filename─────┘
```

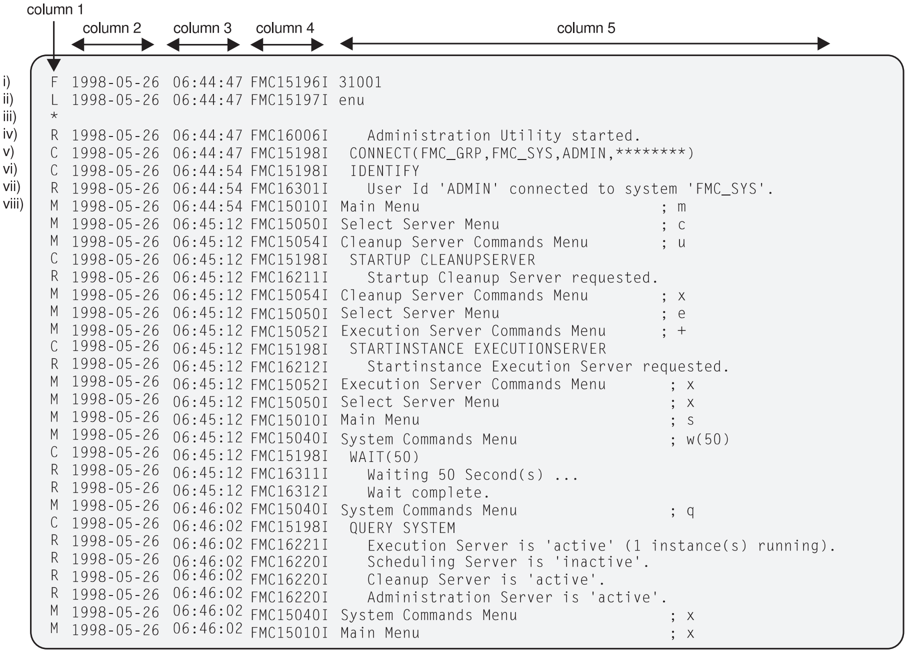
For AIX, HP-UX, and Sun Solaris: Use minus (-) instead of slash (/) as delimiter and lowercase letters.

For details regarding the **fmcautil** start command and the variables that follow it refer to “Starting the administration utility” on page 12.

2. Use the administration utility as explained in “Chapter 2. How to administer the MQ Workflow system” on page 11 to perform the tasks you desire. Each task you perform and the title of every menu you select is recorded in the record file.

Viewing the record file

To view the record file a standard text editor can be used. The layout of the file has a specific format. An example record file is shown in Figure 16 on page 97.



```
i)    F 1998-05-26 06:44:47 FMC15196I 31001
ii)   L 1998-05-26 06:44:47 FMC15197I enu
iii)  *
iv)   R 1998-05-26 06:44:47 FMC16006I    Administration Utility started.
v)    C 1998-05-26 06:44:47 FMC15198I  CONNECT(FMC_GRP,FMC_SYS,ADMIN,********)
vi)   C 1998-05-26 06:44:54 FMC15198I  IDENTIFY
vii)  R 1998-05-26 06:44:54 FMC16301I    User Id 'ADMIN' connected to system 'FMC_SYS'.
viii) M 1998-05-26 06:44:54 FMC15010I Main Menu                           ; m
      M 1998-05-26 06:45:12 FMC15050I Select Server Menu                  ; c
      M 1998-05-26 06:45:12 FMC15054I Cleanup Server Commands Menu        ; u
      C 1998-05-26 06:45:12 FMC15198I  STARTUP CLEANUPSERVER
      R 1998-05-26 06:45:12 FMC16211I    Startup Cleanup Server requested.
      M 1998-05-26 06:45:12 FMC15054I Cleanup Server Commands Menu        ; x
      M 1998-05-26 06:45:12 FMC15050I Select Server Menu                  ; e
      M 1998-05-26 06:45:12 FMC15052I Execution Server Commands Menu      ; +
      C 1998-05-26 06:45:12 FMC15198I  STARTINSTANCE EXECUTIONSERVER
      R 1998-05-26 06:45:12 FMC16212I    Startinstance Execution Server requested.
      M 1998-05-26 06:45:12 FMC15052I Execution Server Commands Menu       ; x
      M 1998-05-26 06:45:12 FMC15050I Select Server Menu                   ; x
      M 1998-05-26 06:45:12 FMC15010I Main Menu                            ; s
      M 1998-05-26 06:45:12 FMC15040I System Commands Menu                 ; w(50)
      C 1998-05-26 06:45:12 FMC15198I  WAIT(50)
      R 1998-05-26 06:45:12 FMC16311I    Waiting 50 Second(s) ...
      R 1998-05-26 06:45:12 FMC16312I    Wait complete.
      M 1998-05-26 06:46:02 FMC15040I System Commands Menu                 ; q
      C 1998-05-26 06:46:02 FMC15198I  QUERY SYSTEM
      R 1998-05-26 06:46:02 FMC16221I    Execution Server is 'active' (1 instance(s) running).
      R 1998-05-26 06:46:02 FMC16220I    Scheduling Server is 'inactive'.
      R 1998-05-26 06:46:02 FMC16220I    Cleanup Server is 'active'.
      R 1998-05-26 06:46:02 FMC16220I    Administration Server is 'active'.
      M 1998-05-26 06:46:02 FMC15040I System Commands Menu                 ; x
      M 1998-05-26 06:46:02 FMC15010I Main Menu                            ; x
```

Figure 16. An example record file

In order to explain the format of the record file the example in Figure 16 has been divided into five columns, 1 to 5. The first eight rows have also been numbered, from i) to viii).

1. The first column in the file (column 1), located on the left-hand side of the screen, contains a list of characters. Each character is an abbreviation describing the type of syntax that follows it. These characters are used to signify comments, format entries, language entries, menu entries, command entries, and responses. The following shows the character convention used and the meaning of each:

   F **Format**. An example of this character is shown at column 1, row i). This character always occurs at the beginning of the file and determines the layout of the file and its recorded contents.

   L An example of this character is shown at column 1, row ii). This character points to the type of language that is used, which is stated in column 5. In this example the language used is English U.S. (enu).

   * **Comment**. An example of this character is shown at column 1, row iii). This character is used to space rows for readability.

   M **Menu entry**. An example of this character is shown at column 1, row viii). This character is used to indicate that a menu screen appeared on the screen while you were using the administration utility. The name of the menu is written into the record file at the start of the fifth column, (column 5, row viii)). In this example the Main Menu is the menu that appeared.

C **Executed command**. An example of this character is shown at column 1, row v). When you enter a command in the administration utility a routine is executed. This character indicates that a routine has been executed. The routine is described by the command message that is written into the fifth column (column 5, row v)). This message is indented by two spaces from the beginning of the fifth column for better readability. In this example the **CONNECT** command is the command that was entered.

R **A response or return value**. An example of this character is shown at column 1, row iv). This character indicates a response displayed or return value received by the administration utility. The response or return value is written into the fifth column (column 5, row iv)) and is indented by four spaces from the beginning of that column for better readability. In this example an ″Administration Utility started″ message is the response that was displayed.

2. The second column contains the date stamp which indicates the date on which each entry in the record file was made. The date stamp occurs on every line. The first example is written in column 2, row i).
3. The third column contains the time stamp which indicates the time of day on which each entry in the record file was made. The time stamp occurs on every line. The first example is written in column 3, row i).
4. The fourth column contains a list of message codes that are coded descriptions of the response or return messages that appeared, commands that were entered, and menu screens that were selected while the administration utility was being used. A format code also exists. There are many of these message numbers, far too many to list here. However, it is sufficient to know that each number is the message code for the command routines, return values, menus, messages, and comments that appear in column 5.
5. The fifth column contains the most important information. The content of the fifth column is made up of practically everything you see during your administration utility session, with the exception of the menu screens and unsolicited messages sent by the administration server. Although the menu screens are not contained in the record file, the title of each menu you select is. While the administration utility is used, it writes the following into the fifth column:

Names of the menu screens The names of menu screens that appeared.

These are written at the beginning of the fifth column (column 5, row viii)).

Command messages
The command messages that appeared after you entered a command. These are indented by two spaces from the beginning of the fifth column (column 5, row v)).

Response or return messages
The response or return messages that appeared. These could be messages informing you about the start or completion of a command, warnings or errors that occurred. These are indented by four spaces from the beginning of the fifth column (column 5, row iv)).

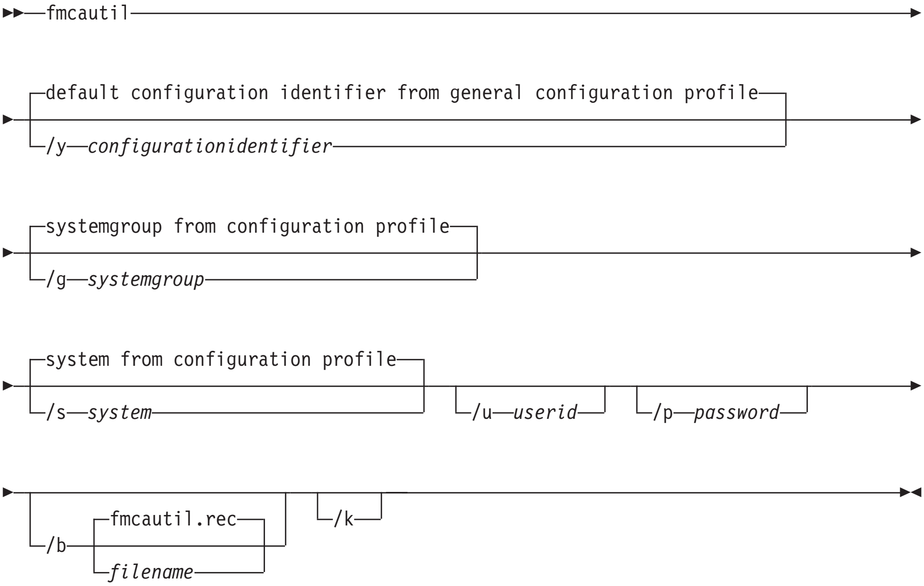
Input data
The command that you entered or menu you selected while using the administration utility. Input data is preceded by a ″;″ (column 5, row viii)).

Playing back a record file

A playback facility is used to automate task processing and avoid re-entering the same tasks again and again.

The available playback option allows you to play back the commands and menu selections previously recorded in the record file. The input data in the record file that is preceded by a semicolon (see column 5 in Figure 16 on page 97), is called upon during playback and executed in sequence. This feature saves you from re-entering commands when you have to repeat the same system administration tasks. To initiate the playback feature, the **fmcautil** start command is entered with the **/b** playback option specified. To invoke the playback option:

- Open a command prompt window and start the administration utility by entering the **fmcautil** start command followed by the appropriate options as shown in the following syntax diagram. Make sure to specify the **/b** playback option and include the filename of the record file that you want to use, unless the file uses the default name, **fmcautil.rec**.


fmcautil
default configuration identifier from general configuration profile
/y
configurationidentifier
systemgroup from configuration profile
/g
systemgroup
system from configuration profile
/s
system
/u
userid
/p
password
/b
fmcautil.rec
filename
/k

For AIX, HP-UX, and Sun Solaris: Use minus (-) instead of slash (/) as delimiter and lowercase letters.

For details regarding the **fmcautil** start command and the variables that follow it refer to “Starting the administration utility” on page 12.

Specifying the **/b** option invokes the playback feature which reads the menu entries recorded in the record file and replays the input data preceded by a "**;**". Therefore, the same menus are selected and the same commands are executed as done while recording the file.

The **/k** option can be used after the playback option to stay within the administration utility once playback is complete. If it is not specified, after invoking the playback option the administration utility terminates.

If you specify the record and playback options together within the **fmcautil** syntax, make sure that the file names you specify after the record and playback options refer to different files. This is because reading and writing to the same file at the same time is not possible.

Notices

This information was developed for products and services offered in the U.S.A. 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:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, 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.

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 Deutschland
Informationssysteme GmbH
Department 3982
Pascalstrasse 100
70569 Stuttgart
Germany

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 or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

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 illustrates 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. 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 IBM's application programming interfaces.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp.1993, 2001. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

- IBM
- AIX
- DB2
- FlowMark
- MQSeries
- MVS
- OS/2
- AS/400

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.

Glossary

This glossary defines important terms and abbreviations used in this publication. If you do not find the term you are looking for, refer to the index or the *IBM Dictionary of Computing*, New York: McGraw-Hill, 1994.

A

administration server. The MQSeries Workflow component that performs administration functions within an MQSeries Workflow system. Functions include starting and stopping of the MQSeries Workflow system, performing error management, and participating in administrative functions for a system group.

activity. One of the steps that make up a process model. This can be a program activity, process activity, or block activity.

activity information member. A predefined data structure member associated with the operating characteristics of an activity.

API. Application Programming Interface.

application programming interface. An interface provided by the MQSeries Workflow workflow manager that enables programs to request services from the MQSeries Workflow workflow manager. The services are provided synchronously.

audit trail. A relational table in the database that contains an entry for each major event during execution of a process instance.

authorization. The attributes of a user's staff definition that determine the user's level of authority in MQSeries Workflow. The system administrator is allowed to perform all functions.

B

bend point. A point at which a connector starts, ends, or changes direction.

block activity. A composite activity that consists of a group of activities, which can be connected with control and data connectors. A block activity is used to implement a Do-Until loop; all activities within the block activity are processed until the exit condition of the block activity evaluates to true. See also *composite activity.*

Buildtime. An MQSeries Workflow component with a graphical user interface for creating and maintaining workflow models, administering resources, and the system network definitions.

C

cardinality. (1) An attribute of a relationship that describes the membership quantity. There are four types of cardinality: One-to-one, one-to-many, many-to-many, and many-to-one. (2) The number of rows in a database table or the number of different values in a column of a database table.

child organization. An organization within the hierarchy of administrative units of an enterprise that has a parent organization. Each child organization can have one parent organization and several child organizations. The parent is one level above in the hierarchy. Contrast with *parent organization*.

cleanup server. The MQSeries Workflow component that physically deletes information in the MQSeries Workflow Runtime database, which had only been deleted logically.

composite activity. An activity which is composed of other activities. Composite activities are block activities and bundle activities.

container API. An MQSeries Workflow API that allows programs executing under the control of MQSeries Workflow to obtain data from the input and output container of the activity and to store data in the output container of the activity.

control connector. Defines the potential flow of control between two nodes in the process. The actual flow of control is determined at run time based on the truth value of the transition conditions associated with the control connector.

coordinator. A predefined role that is automatically assigned to the person designated to coordinate a role.

D

data connector. Defines the flow of data between containers.

data container. Storage for the input and output data of an activity or process. See *input container* and *output container*.

data mapping. Specifies, for a data connector, which fields from the associated source container are mapped to which fields in the associated target container.

data structure. A named entity that consists of a set of data structure members. Input and output containers are defined by reference to a data structure and adopt the layout of the referenced data structure type.

data structure member. One of the variables of which a data structure is composed.

default control connector. The graphical representation of a standard control connector, shown in the process diagram. Control flows along this connector if no other control path is valid.

domain. A set of MQSeries Workflow system groups which have the same meta-model, share the same staff information, and topology information. Communication between the components in the domain is via message queuing.

dynamic staff assignment. A method of assigning staff to an activity by specifying criteria such as role, organization, or level. When an activity is ready, the users who meet the selection criteria receive the activity to be worked on. See also *level*, *organization*, *process administrator*, and *role*.

E

end activity. An activity that has no outgoing control connector.

execution server. The MQSeries Workflow component that performs the processing of process instances at runtime.

exit condition. A logical expression that specifies whether an activity is complete.

export. An MQSeries Workflow utility program for retrieving information from the MQSeries Workflow database and making it available in MQSeries Workflow Definition Language (FDL) or HTML format. Contrast with *import*.

F

fixed member. A predefined data structure member that provides information about the current activity. The value of a fixed member is set by the MQSeries Workflow workflow manager.

(FDL) MQSeries Workflow Definition Language. The language used to exchange MQSeries Workflow information between MQSeries Workflow system groups. The language is used by the import and export function of MQSeries Workflow and contains the workflow definitions for staff, programs, data structures, and topology. This allows non-MQSeries Workflow components to interact with MQSeries Workflow. See also *export* and *import*.

fork activity. An activity that is the source of multiple control connectors.

form. In Lotus Notes, a form controls how you enter information into Lotus Notes and how that information is displayed and printed.

formula. In Lotus Notes, a mathematical expression that is used, for example, to select documents from a database or to calculate values for display.

fully-qualified name. A qualified name that is complete; that is, one that includes all names in the hierarchical sequence above the structure member to which the name refers, as well as the name of the member itself.

I

import. An MQSeries Workflow utility program that accepts information in the MQSeries Workflow definition language (FDL) format and places it in an MQSeries Workflow database. Contrast with *export*.

input container. Storage for data used as input to an activity or process. See also *source* and *data mapping*.

L

level. A number from 0 through 9 that is assigned to each person in an MQSeries Workflow database. The person who defines staff in Buildtime can assign a meaning to these numbers such as rank and experience. Level is one of the criteria that can be used to dynamically assign activities to people.

local user. Identifies a user during staff resolution whose home server is in the same system group as the originating process.

local subprocess. A subprocess that is processed in the same MQSeries Workflow system group as the originating process.

logical expression. An expression composed of operators and operands that, when evaluated, gives a result of true, false, or an integer. (Nonzero integers are equivalent to false.) See also *exit condition* and *transition condition*.

M

manager. A predefined role that is automatically assigned to the person who is defined as head of an organization.

message queuing. A communication technique that uses asynchronous messages for communication between software components.

N

navigation. Movement from a completed activity to subsequent activities in a process. The paths followed are determined by control connectors, their associated transition conditions, and by the start conditions of activities. See also *control connector*, *exit condition*, *transition condition*, and *start condition*.

node. (1) The generic name for activities within a process diagram. (2) The operating system image that hosts MQSeries Workflow systems.

notification. An MQSeries Workflow facility that can notify a designated person when a process or activity is not completed within the specified time.

notification work item. A work item that represents an activity or process notification.

O

organization. An administrative unit of an enterprise. Organization is one of the criteria that can be used to dynamically assign activities to people. See *child organization* and *parent organization*.

output container. Storage for data produced by an activity or process for use by other activities or for evaluation of conditions. See also *sink*.

P

parent organization. An organization within the hierarchy of administrative units of an enterprise that has one or more child organizations. A child is one level below its parent in the hierarchy. Contrast with child *child organization*.

parent process. A process instance that contains the process activity which started the process as a subprocess.

pattern activity. A single and simple activity in a bundle activity from which multiple instances, called pattern activity instances, are created at run time.

person (pl. people). A member of staff in an enterprise who has been defined in the MQSeries Workflow database.

predefined data structure member. A data structure member predefined by MQSeries Workflow and used for communication between user applications and MQSeries Workflow Client.

process. Synonymously used for a process model and a process instance. The actual meaning is typically derived from the context.

process activity. An activity that is part of a process model. When a process activity is executed, an instance of the process model is created and executed.

process administrator. A person who is the administrator for a particular process instance. The administrator is authorized to perform all operations on a process instance. The administrator is also the target for staff resolution and notification.

process category. An attribute that a process modeler can specify for a process model to limit the set of users who are authorized to perform functions on the appropriate process instances.

process definition. Synonym for *process model*.

process diagram. A graphical representation of a process that shows the properties of a process model.

process instance. An instance of a process to be executed in MQSeries Workflow Runtime.

process instance list. A set of process instances that are selected and sorted according to user-defined criteria.

process instance monitor. An MQSeries Workflow client component that shows the state of a particular process instance graphically.

process management. The MQSeries Workflow Runtime tasks associated with process instances. These consist of creating, starting, suspending, resuming, terminating, restarting, and deleting process instances.

process model. A set of processes represented in a process model. The processes are represented in graphical form in the process diagram. The process model contains the definitions for staff, programs, and data structures associated with the activities of the process. After having translated the process model into a process template, the process template can be executed over and over again. *Workflow model* and *process definition* are synonyms.

process monitor API. An application programming interface that allows applications to implement the functions of a process instance monitor.

process-relevant data. Data that is used to control the sequence of activities in a process instance.

process status. The status of a process instance.

process template. A fixed form of a process model from which process instances can be created. It is the translated form in MQSeries Workflow Runtime. See also *process instance*.

process template list. A set of process templates that have been selected and sorted according to user-defined criteria.

program. A computer-based application that serves as the implementation of a program activity or as a support tool. Program activities reference executable programs using the logical names associated with the programs in MQSeries Workflow program registrations. See also *program registration*.

program activity. An activity that is executed by a registered program. Starting this activity invokes the program. Contrast with *process activity*.

program execution agent. The MQSeries Workflow component that manages the implementations of program activities, such as .EXE and .DLL files.

program registration. Registering a program in MQSeries Workflow so that sufficient information is available for managing the program when it is executed by MQSeries Workflow.

R

role. A responsibility that is defined for staff members. Role is one of the criteria that can be used to dynamically assign activities to people.

S

scheduling server. The MQSeries Workflow component that schedules actions based on time events, such as resuming suspended work items, or detecting overdue processes.

server. The servers that make up an MQSeries Workflow system are called execution server, Administration server, Scheduling server, and Cleanup server.

sink. The symbol that represents the output container of a process or a block activity.

source. The symbol that represents the input container of a process or a block activity.

specific resource assignment. A method of assigning resources to processes or activities by specifying their user IDs.

standard client. The MQSeries Workflow component, which enables creation and control of process instances, working with worklists and work items, and manipulation of personal data of the logged-on user.

start activity. An activity that has no incoming control connector.

start condition. The condition that determines whether an activity with incoming control connectors can start after all of the incoming control connectors are evaluated.

subprocess. A process instance that is started by a process activity.

substitute. The person to whom an activity is automatically transferred when the person to whom the activity was originally assigned is declared as absent.

support tool. A program that end users can start from their worklists in the MQSeries Workflow Client to help complete an activity.

symbolic reference. A reference to a specific data item, the process name, or activity name in the description text of activities or in the command-line parameters of program registrations. Symbolic references are expressed as pairs of percent signs (%) that enclose the fully-qualified name of a data item, or either of the keywords _PROCESS or _ACTIVITY.

system. The smallest MQSeries Workflow unit within an MQSeries Workflow domain. It consists of a set of the MQSeries Workflow servers.

system group. A set of MQSeries Workflow systems that share the same database.

system administrator. (1) A predefined role that conveys all authorizations and that can be assigned to exactly one person in an MQSeries Workflow system. (2) The person at a computer installation who designs, controls, and manages the use of the computer system.

T

top-level process. A process instance that is not a subprocess and is started from a user’s process instance list or from an application program.

transition condition. A logical expression associated with a conditional control connector. If specified, it must be true for control to flow along the associated control connector. See also *control connector*.

translate. The action that converts a process model into a Runtime process template.

U

user ID. An alphanumeric string that uniquely identifies an MQSeries Workflow user.

V

verify. The action that checks a process model for completeness.

W

workflow. The sequence of activities performed in accordance with the business processes of an enterprise.

Workflow Management Coalition (WfMC). A non-profit organization of vendors and users of workflow management systems. The Coalition’s mission is to promote workflow standards for workflow management systems to allow interoperability between different implementations.

workflow model. Synonym for *process model*.

work item. Representation of work to be done in the context of an activity in a process instance.

work item set of a user. All work items assigned to a user.

worklist. A list of work items assigned to a user and retrieved from a workflow management system.

worklist view. List of work items and notifications selected from a work item set of a user according to filter criteria which are an attribute of a worklist. It can be sorted according to sort criteria if specified for this worklist.

Bibliography

To order any of the following publications, contact your IBM representative or IBM branch office.

MQSeries Workflow publications

This section lists the publications included in the MQSeries Workflow library.

- *IBM MQSeries Workflow: List of Workstation Server Processor Groups*, GH12-6357, lists the processor groups for MQSeries Workflow.
- *IBM MQSeries Workflow: Concepts and Architecture*, GH12-6285, explains the basic concepts of MQSeries Workflow. It also describes the architecture of MQSeries Workflow and how the components fit together.
- *IBM MQSeries Workflow: Getting Started with Buildtime*, SH12-6286, describes how to use Buildtime of MQSeries Workflow.
- *IBM MQSeries Workflow: Getting Started with Runtime*, SH12-6287, describes how to get started with the Client.
- *IBM MQSeries Workflow: Programming Guide*, SH12-6291, explains the application programming interfaces (APIs).
- *IBM MQSeries Workflow: Installation Guide*, SH12-6288, contains information and procedures for installing and customizing MQSeries Workflow.
- *IBM MQSeries Workflow: Administration Guide*, SH12-6289, explains how to administer an MQSeries Workflow system.

Related publications

- *Frank Leymann, Dieter Roller, "Workflow-based Applications", IBM Systems Journal 36, no. 1(1997): 102–123*, you can also refer to the Internet: http://www.almaden.ibm.com/journal/sj361/leymann.html
- *Workflow Handbook 1997, published in association with WfMC*, edited by Peter Lawrence

Index

Readers' Comments — We'd Like to Hear from You

IBM MQSeries Workflow
Administration Guide
Version 3.3

Publication No. SH12-6289-04

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	□	□	□	□	□

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	□	□	□	□	□
Complete	□	□	□	□	□
Easy to find	□	□	□	□	□
Easy to understand	□	□	□	□	□
Well organized	□	□	□	□	□
Applicable to your tasks	□	□	□	□	□

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? □ Yes □ No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Name

Address

Company or Organization

Phone No.

Readers' Comments — We'd Like to Hear from You

SH12-6289-04

Cut or Fold
Along Line

Fold and Tape

Please do not staple

Fold and Tape

PLACE
POSTAGE
STAMP
HERE

IBM Deutschland Entwicklung GmbH
Information Development, Dept. 0446
Schoenaicher Str. 220
71032 Boeblingen
Germany

Fold and Tape

Please do not staple

Fold and Tape

Cut or Fold
Along Line

Part Number: CT897IE
Program Number: 5697-FM3

Printed in Denmark by IBM Danmark A/S

SH12-6289-04

(1P) P/N: CT897IE