

WebSphere Message Broker



Record and Replay

Version 8 Release 0

WebSphere Message Broker



Record and Replay

Version 8 Release 0

Note

Before using this information and the product it supports, read the information in "Notices" on page 25.

This edition applies to version 8, release 0, modification 0 of IBM WebSphere Message Broker and to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright IBM Corporation 2011.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

About this topic collection. v

**Chapter 1. Recording, viewing, and
replaying data 1**

Record and replay 1

Enabling security for record and replay 1

Chapter 2. Recording data 5

DataCaptureStore configurable service properties . . . 8

DataCaptureSource configurable service properties . . . 10

Disabling and enabling data capture 10

Tuning data capture 11

Recording and replaying data with a CMP
application 13

Chapter 3. Viewing recorded data 15

Configuring a web administration server for data
capture 16

 webadmin component parameter values. 18

Chapter 4. Replaying data 21

Replaying data by using the Message Broker API. . . 21

Replaying data by using a web browser 21

DataDestination configurable service properties . . . 23

Notices 25

Trademarks 27

About this topic collection

This topic collection explains how to record, view, and replay messages in WebSphere® Message Broker Version 8.0.

This PDF contains information that is extracted from the WebSphere Message Broker Version 8.0 information center. The content of these topics is created for viewing online; you might find that the formatting and presentation of some figures, tables, examples, and so on are not optimized for the printed page. Text highlighting might also have a different appearance.

The topic content included in the PDF does not include the "Related Links" sections provided in the online topics. Links within the topic content itself are included, but are active only if they link to another topic in the same PDF collection. Use the online information center to view related topics that are not included in this PDF.

Chapter 1. Recording, viewing, and replaying data

You can record data that is being processed by a message flow, then view it, and replay it.

For concept information about recording and replaying data, see “Record and replay.”

To record, view, and replay data, see the following topics.

- Chapter 2, “Recording data,” on page 5
- Chapter 3, “Viewing recorded data,” on page 15
- Chapter 4, “Replaying data,” on page 21

Record and replay

For audit or problem determination purposes, you can record data to a database, then view it, and replay it.

If you need an audit record of messages that pass through the broker, you can record those messages. You might also want to record messages if you need to keep a history of messages for development and test purposes, or to help in problem determination.

To determine which data is recorded, you must configure monitoring on the message flow. Data is stored in a database. You must create a data source, then use a configurable service to define the data source name to use when recording data.

After you have recorded data, you can view it in several ways. By using the web interface, you can view a list of recorded messages, or you can view details of a specific message. You can also view recorded data by using the Message Broker API or the WebSphere Message Broker Representational State Transfer (REST) API.

You can replay a message to a WebSphere MQ queue by using the web interface or Message Broker API. You can also replay the message to another message flow for testing or problem determination.

Enabling security for record and replay

You can restrict the number of users who can view and replay data for a broker by enabling security.

Before you start:

For information about how to configure your system to record data, see Chapter 2, “Recording data,” on page 5.

If you do not enable security, all users can complete all actions against a broker and all execution groups. To enable administrative security, you must set the **-s** parameter on the **mqsicreatebroker** or **mqsichangebroker** command. The **-s** parameter specifies the administrative security status for the broker; by default, this parameter is set to `inactive`. If you set the **-s** parameter to `active`, administrative security is enabled and only user IDs that you authorize are

permitted to complete actions on the broker. When you create an execution group on a broker for which administrative security is enabled, the queue `SYSTEM.BROKER.AUTH.executionGroupName` is created. Populate the queue with the appropriate user authorization.

To enable security for a broker, complete the following steps.

1. To use a web administration server to view and replay data, you must create a user called `ibmuser` on your system. The user `ibmuser` cannot be a member of the `mqm` or `mqbrkr` groups. You do not need to create this user if you are not enabling security. You will use the user ID `ibmuser` to log in to the web administration server.

2. Enable administrative security by setting the `-s` parameter to active.

- To enable administrative security when you create the broker, run the `mqsicreatebroker`, as shown in the following example:

```
mqsicreatebroker brokerName -q brokerQueueManagerName -s active
```

(If you run this command on Windows, you must also set the `-i` parameter. For details, see `mqsicreatebroker command` in the information center.)

- To enable administrative security for a broker that you have already created, stop the broker, then run the `mqsichangebroker`, as shown in the following example:

```
mqsichangebroker brokerName -s active
```

For more information, see *Activating broker administration security* in the information center.

3. To allow the user `ibmuser` to run record and replay queries on the execution group, ensure that they have inquire (`inq`) administration authority on the queues `SYSTEM.BROKER.AUTH` and `SYSTEM.BROKER.AUTH.executionGroupName`.

For more information, see *Granting and revoking authority for broker administration security* in the information center.

4. In addition to setting administrative security, you must also set security for data capture. The queue `SYSTEM.BROKER.DC.AUTH` controls the record and replay actions that user `ibmuser` can complete on the broker. Ensure that the user has the appropriate authorization to complete the following actions on this queue:

Action	Authority required
To view data	Inquire (+inq)
To view bit streams and exception lists	Write (+put)
To replay data	Execute (+set)

To change these authorizations, you can use WebSphere MQ commands or the WebSphere Message Broker Explorer.

To use WebSphere MQ commands, see *Granting and revoking authority on Linux, UNIX, and Windows systems* in the information center.

To use the WebSphere Message Broker Explorer, complete the following steps.

- a. In the MQ Explorer - Navigator view, navigate to **IBM WebSphere MQBrokers > Queene Managers**, expand your queue manager, and select **Queues**.
- b. Right-click the queue `SYSTEM.BROKER.DC.AUTH`, then click **Object Authorities > Manage Authority Records**.

- c. Expand **Specific Profiles** and click SYSTEM.BROKER.DC.AUTH.
 - d. On the **Users** tab, select **ibmuser** and click **Edit**.
 - e. Set the appropriate authorizations and click **OK**, then close the Manage Authority Records dialog box.
5. If you change authorizations, restart your broker and broker queue manager to ensure that changes take effect.

To disable security, set the **-s** parameter to `inactive` on the **mqsi changebroker** command.

Next:

To view data that has been recorded, see Chapter 3, “Viewing recorded data,” on page 15.

To replay data that has been recorded, see Chapter 4, “Replaying data,” on page 21.

Chapter 2. Recording data

To record data that is flowing through a message flow, create a database, configure security settings, create appropriate configurable services, and configure monitoring for the message flow.

Before you start:

Ensure that the message flow for which you want to record data has been deployed. For more information, see *Deploying resources* in the information center.

You can record data to a database for audit purposes, or to help with problem determination. A script has been provided with WebSphere Message Broker that you can use to create the DB2® database and database tables. You can run this script unmodified, or you can customize it. The script creates a database called MBRECORD with a default schema.

To configure WebSphere Message Broker to record data, complete the following steps.

1. Use the script that is provided with WebSphere Message Broker to create and configure a database to store your recorded data.

- a. Locate the script for your operating system:

- Windows: *install_dir*\ddl\db2\DataCaptureSchema.sql
- UNIX: *install_dir*/ddl/db2/DataCaptureSchema.sql
- z/OS® : the JCL script is located in the data set and member SBIPPROC(BIPRRDB)

install_dir is the location of your WebSphere Message Broker installation.

- b. Optional: To specify your own database or schema, customize the provided DataCaptureSchema script, and save your changes.

If you modify the SQL to specify a particular schema, you must also set the same schema name in the DataCaptureStore configurable service.

You might also want to edit the script for the following reasons:

- If you have run the script once, and want to run it again, you must drop the database MBRECORD first. Insert the command drop database MBRECORD before the line that reads create database MBRECORD.
- The maximum message body size that you can record (after encoding has taken place) is 5 MB (which is also the default size). You can increase this size by editing the script to make the value in the WMB_BINARY_DATA.DATA column larger.

- c. At a command line, navigate to the script location and run it.

On Windows, use a DB2 Command Window to ensure that the command environment is set up correctly. Click **Start > IBM DB2 > databaseInstance > Command Line Tools**, and select **Command Window**, where *databaseInstance* is the DB2 installation name.

On UNIX, a script called *db2profile* is provided for setting up the environment; see *Command environment: Linux and UNIX systems* in the information center.

When the command environment is set up, you can run the script.

For example, on Windows or UNIX, enter the following command:

```
db2 -tvf DataCaptureSchema.sql
```

2. Create an ODBC definition for the database.

If you have used the supplied script to create your database without modifications, create an ODBC definition for the database called MBRECORD, with MBRECORD as the data source name (DSN). For more information, see *Enabling ODBC connections to the database* in the information center.

3. Use the **mqsisetdbparms** command to set a user identifier and password for the broker to use when connecting to the database; for example:

```
mqsisetdbparms brokerName -n dataSourceName -u userID -p password
```

- *brokerName* is the name of your broker.
- *dataSourceName* identifies the database to which you want to record data.
- *userID* and *password* specify the user identifier and the password that the broker uses to connect to the database.

4. To ensure that the changes to the **mqsisetdbparms** command take effect, restart the broker. For more information, see *Starting and stopping a broker* in the information center.
5. Test the connection to your database by using the **mqsicvp** command. For more information, see *mqsicvp command* in the information center.
6. Specify the runtime properties for data capture by creating the appropriate configurable services.

To specify data processing information and connection information for the database, create a DataCaptureStore configurable service.

To identify the monitoring topic and the data capture store to use for processing event data for this topic, create a DataCaptureSource configurable service. Multiple instances of the DataCaptureSource configurable service can use the same DataCaptureStore configurable service.

- a. Configure the DataCaptureStore configurable service.

You can use the provided DefaultCaptureStore configurable service or create your own configurable service of type DataCaptureStore. You can use the WebSphere Message Broker Explorer to create the configurable service; for more information, see *Using the WebSphere Message Broker Explorer to work with configurable services* in the information center. Alternatively, use the **mqsicreateconfigurableservice** command; for more information, see *mqsicreateconfigurableservice command* in the information center. For descriptions of properties of this configurable service, see “DataCaptureStore configurable service properties” on page 8. For example, enter the following command on a command line:

```
mqsicreateconfigurableservice brokerName -c DataCaptureStore -o dataCaptureStoreName  
-n dataSourceName, egForRecord -v dataSource, executionGroup
```

- *brokerName* is the name of your broker.
- *dataCaptureStoreName* is the name of your configurable service object.
- *dataSource* is the name of your data source.
- *executionGroup* is the name of the execution group that processes data for recording.

- b. Configure the DataCaptureSource configurable service.

Create a configurable service of type DataCaptureSource. You can use the WebSphere Message Broker Explorer or the **mqsicreateconfigurableservice** command to create the configurable service. If you use the DataCaptureSourceTemplate in WebSphere Message Broker Explorer, you must create a new configurable service based on the template. If you edit the template without creating a new configurable service, an error is issued

at run time. For information about the properties of this configurable service, see “DataCaptureSource configurable service properties” on page 10. For example, on UNIX systems, enter the following command on a command line:

```
mqsicreateconfigurableservice brokerName -c DataCaptureSource -o dataCaptureSourceName
-n dataCaptureStore,topic
-v dataCaptureStoreName, '$SYS/Broker/myBroker/Monitoring/executionGroupName/msgFlowName'
```

- *brokerName* is the name of your broker.
- *dataCaptureSourceName* is the name of the configurable service object.
- *dataCaptureStoreName* is the name of the DataCaptureStore configurable service that you want to use for this subscription.
- *myBroker*, *executionGroupName*, and *msgFlowName* are the names of the broker, execution group, and message flow from which you want to capture data. These values are part of a topic string, which is used to subscribe to events that you set up by using business monitoring. You can use topic wildcards in this topic string.

On UNIX systems, enclose the topic string in single quotation marks when you enter it on a command line. On Windows systems, use double quotation marks. No quotation marks are required if you create the configurable service by using the WebSphere Message Broker Explorer.

Test that you have successfully subscribed to the topic specified for the topic property by displaying the subscriptions on the queue manager for *brokerName*. Use WebSphere Message Broker Explorer or the **runmqsc** command.

To check the subscriptions by using the WebSphere Message Broker Explorer, complete the following steps:

- 1) Expand the queue manager under the **Queue Managers** folder
- 2) To open the Subscriptions pane, click **Subscriptions**.
- 3) Click **Refresh** and check that a subscription with a topic string of `$SYS/Broker/myBroker/Monitoring/executionGroupName/msgFlowName` exists

To check the subscription by using **runmqsc**, complete the following steps:

- 1) At a command prompt, type `runmqsc qmName`, where *qmName* is your queue manager name.
 - 2) To display all the queue manager subscriptions, type `dis sub(*)`
 - 3) Check that the topic string `$SYS/Broker/myBroker/Monitoring/executionGroupName/msgFlowName` is returned in the list of subscription topics
 - 4) To exit the **runmqsc** environment, type `end`
7. To configure monitoring on a message flow to emit events for capture, use one of the following methods.
- Configure and enable event sources, and activate monitoring for the message flow, by completing the steps in *Business-level monitoring* in the information center.
You do not need to subscribe to the monitoring topic; the broker manages the subscription to the topic specified in the DataCaptureSource configurable service.
 - Configure monitoring event sources by using the sample monitoring profile that is provided at `install_dir/sample/RecordReplay/basicMonitoringProfile.xml`.

The sample monitoring profile works for a single input node. You must update the profile to replace *NODENAME* with the name of your input node. To enable this monitoring profile, run the **mqscreateconfigurable** service, **mqschangeproperties**, and **mqschange** flow monitoring commands against the broker where you have deployed the flow. For details about running these commands, see *Configuring monitoring event sources using a monitoring profile* in the information center.

The subscription to the monitoring events is durable, which results in messages being put to the queue specified in the `queueName` property of the `DataCaptureStore` configurable service. Messages can be put to this queue even when the broker is not running. For more information about how you can monitor the queue depth and tune data capture, see “Tuning data capture” on page 11.

Next:

- You can enable security for recording data by following the instructions in “Enabling security for record and replay” on page 1.
- After you have recorded data, you can view it. For more information, see Chapter 3, “Viewing recorded data,” on page 15.
- You might want to turn off recording for a particular execution group temporarily (for performance reasons, for example). For more information, see “Disabling and enabling data capture” on page 10.
- For more information about tuning data capture, see “Tuning data capture” on page 11.

DataCaptureStore configurable service properties

The following table describes the properties that you can set for the `DataCaptureStore` configurable service.

Supplied configurable services that are created for each broker	Properties for each configurable service that is defined	Description of properties
DefaultCaptureStore	backoutQueue	This property identifies the name of the backout queue used for data capture. Messages that cannot be processed (for example, because the specified database does not exist) are sent to this queue. The default value for the property is <code>SYSTEM.BROKER.DC.BACKOUT</code> . The queue <code>SYSTEM.BROKER.DC.BACKOUT</code> is created when the broker is created. If a data capture source refers to this data capture store, and no <code>backoutQueue</code> has been specified, an error occurs. The same <code>backoutQueue</code> can be specified for multiple data capture stores.
	commitCount	This property identifies the number of input messages that are processed on a data capture thread before a sync point is taken.
	commitIntervalSecs	This property identifies the time interval at which a commit is taken when the <code>commitCount</code> property is greater than 1 but the number of messages processed has not reached the value of the <code>commitCount</code> property.

Supplied configurable services that are created for each broker	Properties for each configurable service that is defined	Description of properties
	dataSourceName	This property represents the ODBC connection definition for the database to which you want to record data. This property does not have a default value, but you get an error if you do not set it.
	egForRecord	This property identifies the execution group to use for processing the data that is being recorded. By default, no value is set. If you do not set a value, and a DataCaptureSource configurable service refers to this DataCaptureStore configurable service, you get a runtime error.
	egForView	This property identifies the execution group to use for processing the data that is retrieved from a database for viewing. By default, no value is set. If you do not set a value, this data capture store is not accessible from a web administration server. You must start the execution group in order to view data.
	queueName	This property identifies the name of the queue that holds the data before it is recorded to the database. The default value for this property is SYSTEM.BROKER.DC.RECORD. The queue SYSTEM.BROKER.DC.RECORD is created when the broker is created. Change the value of this property to distribute the data from multiple data capture sources across multiple queues. If you do not use the default value, you must create the corresponding local queue.
	schema	This property identifies the database schema name to use when recording to a database. This property does not have a default value. If you do not set the property, depending on the database, the default database schema is used, if there is one, or no schema is used.
	threadPoolSize	This property identifies the number of threads that are used by the egForRecord execution group to process the monitoring topic subscriptions. The default value is 10.
	useCoordinatedTransaction	This property determines whether transactions are globally coordinated across WebSphere MQ and database resources. The default value is false. For more information, see <i>Configuring global coordination of transactions (two-phase commit)</i> in the information center.

DataCaptureSource configurable service properties

The following table describes the properties that you can set for the DataCaptureSource configurable service.

Supplied configurable services that are created for each broker	Properties for each configurable service that is defined	Description of properties
None	dataCaptureStore	This property identifies the DataCaptureStore configurable service that is used to configure data capture for the message flows specified in the topic property. Multiple instances of DataCaptureSource configurable services can refer to one instance of DataCaptureStore configurable service. Use the topic property to determine which message flows are processed by the DataCaptureStore configurable service identified in this property.
	topic	This property identifies a subscription topic that is used for business-level monitoring of a message flow. The value of the property takes the form: <code>\$SYS/Broker/brokerName/Monitoring/executionGroup/msgFlow</code> where <i>brokerName</i> , <i>executionGroup</i> , and <i>msgFlow</i> represent the names of your broker, execution group, and message flow. You can use wildcards in the topic string; for more information about using wildcards in topics, see <i>Special characters in topics</i> in the information center. On Windows platforms, enclose the topic string in double quotation marks; on UNIX platforms, use single quotation marks.

Disabling and enabling data capture

Recording of data is enabled by default, but takes effect only if you define a DataCaptureSource configurable service. You can disable recording for all of the data capture sources on an execution group by modifying the broker runtime properties for that execution group.

You might want to temporarily stop recording for an execution group for performance reasons, or to modify the execution group. You can also use these instructions to start recording again.

You can use the WebSphere Message Broker Explorer or WebSphere Message Broker runtime commands to disable recording for your execution group. Complete the following steps.

Review and edit the current setting on the ComIbmDataCaptureManager resource manager.

- If you are using the WebSphere Message Broker Explorer, complete the following steps:
 1. In the MQ Explorer - Navigator view, navigate to **IBM WebSphere MQ > Brokers > brokerName**, where *brokerName* is the name of your broker. Right-click your execution group name, and select **Properties**.
 2. Select the **DataCapture** tab.
 3. Review and set the **DataCapture Enabled** property by selecting either true or false from the menu.
 4. Click **Apply** to review your changes and click **OK** to close the Properties window.

Your change takes effect immediately.

- If you are using the runtime commands, complete the following steps:
 1. Enter the following command on a WebSphere Message Broker command line, where *brokerName* is the name of your broker and *executionGroup* is the name of the execution group:

```
mqsireportproperties brokerName  
-e executionGroup -o ComIbmDataCaptureManager -a
```

The command produces a response similar to this example:

```
ComIbmDataCaptureManager  
  uuid='ComIbmDataCaptureManager'  
  userTraceLevel='none'  
  traceLevel='none'  
  userTraceFilter='none'  
  traceFilter='none'  
  enabled='true'
```

2. Change the enabled property to false, by entering the following command:

```
mqsichangeproperties brokerName  
-e executionGroup -o ComIbmDataCaptureManager -n enabled -v false
```

Your change takes effect immediately.

Next:

When you are ready to start recording again for your execution group, use the **mqsichangeproperties** command to change the enabled property to true, and restart the broker.

Tuning data capture

You can change the broker configuration to improve performance and message throughput when you are recording data. You can configure the number of execution groups and queues used to process the data, the size of the thread pool that is used by each execution group, and how often data is committed. You can also use more than one database for storing data.

Consider each of the following factors when designing your broker configuration for record and replay. For further information about the DataCaptureStore configurable service properties, see “DataCaptureStore configurable service properties” on page 8.

threadPoolSize

This property is set on the DataCaptureStore configurable service. The property determines the number of threads that are used by the execution group specified in the *egForRecord* property to process subscriptions to the monitoring topic.

commitCount and commitIntervalSecs

These properties are set on the DataCaptureStore configurable service. The *commitCount* property identifies the number of input messages that are processed on each thread before a sync point is taken. Decreasing the value of *commitCount* can improve performance. The *commitIntervalSecs* identifies the time interval at which a commit is taken when the *commitCount* property is greater than 1 but the number of messages processed has not reached the value of the *commitCount* property. Increasing the value of *commitIntervalSecs* can improve performance.

egForRecord and egForView

These properties are set on the DataCaptureStore configurable service. You

can spread workload by specifying multiple data capture stores with different execution groups for recording and viewing data specified for each one.

DataCaptureStore and DataCaptureSource configurable service instances

The relationship of the DataCaptureStore configurable service to the DataCaptureSource configurable service is many to one. To use multiple data sources for storing recorded data, you can define several DataCaptureSource configurable services. To increase the numbers of executions groups and queues used to process data for recording and viewing, you can change the ratio of DataCaptureStore to DataCaptureSource configurable services.

queueName and backoutQueue

These properties are set on the DataCaptureStore configurable service. The queueName property specifies the queue that is used for holding data before it is recorded. The backoutQueue property specifies the queue that is used for backing out messages that cannot be processed. You can use the default queues or you can create new queues so that the data is spread across multiple queues.

You can configure WebSphere MQ to warn when queue depths are close to their maximum size. For more information, see the WebSphere MQ documentation about “Event monitoring” on the WebSphere MQ Library web page. You can also increase the maximum queue depth and the maximum message length for queues by using either the **runmqsc** command or the WebSphere Message Broker Explorer.

To use the **runmqsc** command, complete the following steps.

1. At a command prompt, enter the following command, where *qmName* is the name of your queue manager:

```
runmqsc qmName
```
2. Confirm the current settings for your queue by running the following command, where *qName* is the name of the relevant queue; for example, SYSTEM.BROKER.DC.RECORD. If your queue name contains lowercase characters, you must enclose the queue name in single quotation marks.

```
dis qlocal(qName) maxdepth,maxmsgl
```
3. Enter the following command, where *qName* is the name of your queue, *newMaxDepth* is the new setting for maximum queue depth, and *newMaxMsgL* is the new setting for maximum message length:

```
alter qlocal(qName) maxdepth(newMaxDepth) maxmsgl(newMaxMsgL)
```
4. To exit the **runmqsc** environment, type **end**.

To use the WebSphere Message Broker Explorer, complete the following steps.

1. Under the **Queue Managers** folder, right-click the queue manager and select **Properties**.
2. In the menu tree, select **Extended**.
3. Edit the values for the **Max uncommitted messages** or **Max message length** properties.

Recording and replaying data with a CMP application

You can create CMP applications to examine and replay the data that you have recorded and stored by using record and replay.

Before you start:

Read the concept topic “Record and replay” on page 1.

The `DataCaptureProxy` class programmatically represents the recorded data. For more information about where the `DataCaptureProxy` is located in the CMP class hierarchy, see *Navigating brokers and broker resources in a CMP application* in the information center. Retrieved data is contained in `DataCaptureEntry` objects.

For further information about these objects and how to use them, see the *CMP API* in the information center.

The following example shows how to retrieve data from an execution group by using the `DataCaptureProxy` API.

```
BrokerProxy bp = null;
try {
    BrokerProxy.enableAdministrationAPITracing("C:\\AdminTrace.txt");
    bp = BrokerProxy.getLocalInstance("WEB");
    ExecutionGroupProxy eg = bp.getExecutionGroupByName("default");

    Properties newFilterProps = new Properties();
    newFilterProps.setProperty(DataCaptureEntry.PROPERTY_BROKER_NAME, "WEB");

    DataCaptureEntry filterMessage = new DataCaptureEntry(newFilterProps);
    DataCaptureProxy dcp = eg.getDataCapture("MYDATA", filterMessage);

    String output = dcp.getDataCaptureEntryAsXml(1);
    System.out.println("XML output: "+output);

    int messNo = 0;

    Enumeration <DataCaptureEntry> dceE = dcp.elements();
    while (dceE.hasMoreElements()) {
        DataCaptureEntry dce = dceE.nextElement();
        System.out.print("\nMessage: "+messNo++);

        Properties props = dce.getAllProperties();
        String[] columns = dce.getPropertyNames();
        for (int i = 0; i < columns.length; i++) {
            System.out.print("\nProperty: "+columns[i]+ " Value: "+props.getProperty(columns[i]));
        }
    }
} catch (ConfigManagerProxyLoggedException e) {
    e.printStackTrace();
} catch (ConfigManagerProxyPropertyNotInitializedException e) {
    e.printStackTrace();
} finally {
    if(bp != null) {
        bp.disconnect();
    }
}
```

The following example shows how to replay data.

```
BrokerProxy brokerProxy = null;
try {
    // DataCaptureStore containing message to replay
```

```

String dataCaptureStore = "MyStore";

// ID of message to replay
String replayMsgId = "0123456789ABCDEF";

// DataDestination to replay message to
String dataDestination = "ReplayDestination";

// EG to replay
String replayEG = "MyExecutionGroupName";

Properties dataCaptureProps = new Properties();

// Set the message to replay
dataCaptureProps.setProperty(DataCaptureEntry.PROPERTY_WMBMSG_KEY, replayMsgId);

// Set the destination to replay to
dataCaptureProps.setProperty(AttributeConstants.DATACAPTURE_REPLAY, dataDestination);

DataCaptureEntry dataCaptureEntry = new DataCaptureEntry(dataCaptureProps );

brokerProxy = WebAdminBrokerProxy.getLocalInstance("MB8BROKER");
ExecutionGroupProxy egProxy = brokerProxy.getExecutionGroupByName(replayEG);

// Submit request to EG to actually do replay
DataCaptureProxy dataCaptureProxy = egProxy.getDataCapture(dataCaptureStore, dataCaptureEntry);
dataCaptureProxy.hasBeenPopulatedByBroker(true);

String responseBody = dataCaptureProxy.getDataCaptureEntryAsXml(1);

} catch (Exception e) {
    // TODO: Handle Exception
} finally {
    if (brokerProxy != null) {
        brokerProxy.disconnect();
    }
}

```

Chapter 3. Viewing recorded data

You can view a list of recorded messages, or details of individual messages.

Before you start:

Ensure that you have completed all the steps in Chapter 2, “Recording data,” on page 5. You must have configured a database and message flow to record messages. To connect to the database that holds the recorded data, you must have set up a user identifier and password by using the `mqsisetdbparms` command.

You can view recorded data from a web interface or programmatically.

To configure WebSphere Message Broker to view recorded data, complete the following steps.

1. Specify the execution group that is to be used for processing data for viewing by setting a value for the `egForView` property of your `DataCaptureStore` configurable service.

If you are using the configurable services that you used for recording data, use the `mqsichangeproperties` command to modify the `egForView` property, as shown in the following example:

```
mqsichangeproperties brokerName -c DataCaptureStore -o dataCaptureStoreName  
-n egForView -v executionGroupName
```

- *brokerName* is the name of your broker.
- *dataCaptureStoreName* is the name of your `DataCaptureStore` configurable object.
- *executionGroupName* is the name of the execution group that processes the data for viewing.

If you do not want to use the `DataCaptureStore` and `DataCaptureSource` configurable services that you set up for recording data, create new ones as described in Chapter 2, “Recording data,” on page 5. Specify a value for the `egForView` property. You can specify the same execution group in the `egForRecord` and the `egForView` properties, or you might want to spread the workload across two execution groups.

For more information about the `egForRecord` and `egForView` properties, see “`DataCaptureStore` configurable service properties” on page 8.

2. Use one of the following methods to view your data.
 - Use the Message Broker API. For more information, see “Recording and replaying data with a CMP application” on page 13.
 - View recorded data and administer your brokers from a web interface, by configuring a WebSphere Message Broker web administration server. For more information, see “Configuring a web administration server for data capture” on page 16.
 - Use the WebSphere Message Broker Representational State Transfer (REST) application programming interface. Documentation for this application programming interface can be found in the WebSphere Message Broker installation directory under the subfolder `docs/REST/`, for example: `/opt/mqsi/8.0.0.0/docs/REST/`.

Configuring a web administration server for data capture

To view and replay captured data from a web interface, configure the WebSphere Message Broker web administration server.

Before you start:

1. Configure a message flow to record messages by following the instructions in Chapter 2, “Recording data,” on page 5.
2. Complete the steps for viewing recorded data, as described in Chapter 3, “Viewing recorded data,” on page 15.

To configure a web administration server for record and replay, complete the following steps.

1. Configure the properties relating to the HTTP or HTTPS port to be used, and enable the web administration server, by using the WebSphere Message Broker Explorer or the `mqsichangeproperties` command.
 - If you are using the WebSphere Message Broker Explorer, complete the following steps:
 - a. In the MQ Explorer - Navigator view, navigate to **IBM WebSphere MQ > Brokers**, right-click your broker name, and select **Properties**.
 - b. Select the **WebAdmin** tab.
 - c. Enable web administration and set the port to use for your protocol of choice.

If you are using HTTP, set **Enabled** to true, set **Enabled SSL** to false, and enter a value for the **HTTP Port**.

If you are using HTTPS, set **Enabled** to true, set **Enabled SSL** to true, and enter a value for the **HTTPS Port**.

To set additional parameters for the HTTPS protocol, you must use the command-line interface. For more information, see “webadmin component parameter values” on page 18.

- d. To review your changes, click **Apply**; to close the Properties window, click **OK**. You can also confirm the changes by examining recent Change Notification entries in the Administration Log view.
- If you are using the `mqsichangeproperties` command, complete the following steps:
 - a. Configure the properties relating to the HTTP or HTTPS port to be used.
 - If you are using HTTP, run the following command:

```
mqsichangeproperties brokerName
-b webadmin -o HTTPConnector -n port -v portValue
```

 - *brokerName* is the name of your broker
 - *portValue* is the HTTP port value that you want to use for web administration
 - If you are using HTTPS, run the following command:

```
mqsichangeproperties brokerName
-b webadmin -o HTTPSConnector -n port,keystoreFile,keystorePass
-v portValue,fileName,password
```

 - *brokerName* is the name of your broker.
 - *portValue* is the HTTPS port value that you want to use for web administration.
 - *fileName* is your keystore file.
 - *password* is the password for the keystore file.

For more information about the properties that can be set for the HTTPConnector and HTTPSConnector objects, see “webadmin component parameter values” on page 18.

- b. Confirm that the properties are set correctly.
- If you are using HTTP, run the following command:

```
mqsireportproperties brokerName  
-b webadmin -o HTTPConnector -a
```

- If you are using HTTPS, run the following command:

```
mqsireportproperties brokerName  
-b webadmin -o HTTPSConnector -a
```

This command produces a response similar to this example (for the HTTPConnector):

```
HTTPConnector  
uid='HTTPConnector'  
address=''  
port='7054'  
allowTrace=''  
maxPostSize=''  
acceptCount=''  
bufferSize=''  
compressableMimeTypes=''  
compression=''  
connectionLinger=''  
connectionTimeout=''  
maxHTTPHeaderSize=''  
maxKeepAliveRequests=''  
maxSpareThreads=''  
maxThreads=''  
minSpareThreads=''  
noCompressionUserAgents='  
restrictedUserAgents=''  
socketBuffer=''  
tcpNoDelay=''  
enableLookups='false'
```

- c. Enable the web administration server for the broker.
- To enable web administration, and to use HTTP as the communication protocol between the broker and the web administration server, run the following command on a WebSphere Message Broker command line, where *brokerName* is the name of your broker:

```
mqsichangeproperties brokerName  
-b webadmin -o server -n enabled,enableSSL -v true,false
```

- To enable web administration, and to use HTTPS as the protocol for communication between the broker and the web administration server, run the following command:

```
mqsichangeproperties brokerName  
-b webadmin -o server -n enabled,enableSSL -v true,true
```

For more information about the properties of the web administration server, see “webadmin component parameter values” on page 18.

- d. Confirm that the web administration component is enabled by running the **mqsireportproperties** command, as shown in the following example.

```
mqsireportproperties brokerName -b webadmin -o server -a
```

This command produces a response similar to this example:

```
server=''  
uid='server'  
enabled='true'  
enableSSL='true'
```

2. To ensure that the changes take effect, restart the broker. For more information, see *Starting and stopping a broker* in the information center.
3. Open the web administration server interface in a web browser. Use the following URL:`protocol://serverAddress:port/` where
 - *protocol* has the value `http` or `https`, depending on whether you are using an HTTP or an HTTPS connector object
 - *serverAddress* identifies the web administration server address specified for the HTTP or HTTPS connector object; for example `127.0.0.1`
 - *port* identifies the port that you specified for the HTTP or HTTPS connector object in the preceding steps

Click **OK** to log in as user `ibmuser`. (You cannot change the user ID or password.)

Click the **Administration** tab to open a view where you can view and administer your broker resources.

Click the **Data viewer** tab to open a view where you can view your recorded data. Select the data store containing the data that you want to view from the **Select data store** menu. A data store shows in the **Select data Store** menu only if the `egForView` property was set in the corresponding `DataCaptureStore` configurable service. For more information about the fields in this view, see *The monitoring event* in the information center.

Next:

To replay a message, see Chapter 4, “Replaying data,” on page 21.

webadmin component parameter values

Select the objects and properties associated with the `webadmin` component that you want to change.

To change these properties, you must specify the broker name, `-b webadmin`, and `-o objectName`, where *objectName* takes one of the following values:

- `HTTPConnector`. Use this value to set properties for the HTTP port that listens for web administration requests. Property values and usage are the same as for the properties available on the `HTTPConnector` option on the `httplistener` component. See *httplistener component parameter values* in the information center.
- `HTTPSConnector`. Use this value to set properties for the secure HTTPS port that listens for web administration requests. Property values and usage are the same as for the properties available on the `HTTPSConnector` option on the `httplistener` component.
- `server`. Use this value to set the properties that control web administration support in WebSphere Message Broker.

You can select the following properties when you specify `server` as the *objectName*:

enabled

A Boolean value to establish whether the broker is to start the web administration server.

- Value type - Boolean
- Initial value - false

enableSSL

A Boolean value to specify whether HTTPS is to be used as the communication protocol between the broker and the web administration

server. You must set this value to true if you need to use HTTPS. If you specify a value of true, you must configure the HTTPSConnector option, ensuring that you set appropriate values in the keystore-related properties.

- Value type - Boolean
- Initial value - unset (equivalent to false)

Chapter 4. Replaying data

When you have recorded data, you can replay it to a WebSphere MQ queue.

Before you start:

To replay data, you must have configured your system to record data and to view that recorded data. For instructions, see Chapter 2, “Recording data,” on page 5 and Chapter 3, “Viewing recorded data,” on page 15.

You can replay data that you have recorded by using the Message Broker API or the web administration server interface.

Replaying data by using the Message Broker API

To replay data to a WebSphere MQ queue by using the Message Broker API, complete the following steps.

1. Use the DataDestination configurable service to define the WebSphere MQ queues to which you can replay data.

Create the configurable service by using the WebSphere Message Broker Explorer or the `mqsicreateconfigurableservice` command. You can set one destination for each instance of the DataDestination configurable service. This configurable service is dynamic, therefore you do not need to restart the broker for changes to take effect.

For a description of the properties of this configurable service, see “DataDestination configurable service properties” on page 23.

2. For an example of how to replay data with the Message Broker API, see “Recording and replaying data with a CMP application” on page 13.

Replaying data by using a web browser

To replay data to a WebSphere MQ queue from a web browser, complete the following steps.

1. Use the DataDestination configurable service to define the WebSphere MQ queues to which you can replay data.






Create the configurable service by using the WebSphere Message Broker Explorer or the `mqsicreateconfigurableservice` command. You can set one destination for each instance of the DataDestination configurable service. This configurable service is dynamic, therefore you do not need to restart the broker for changes to take effect.

For a description of the properties of this configurable service, see “DataDestination configurable service properties” on page 23.


2. Open the web administration server interface in a web browser, using the URL protocol://serverAddress:port/

For more information, see “Configuring a web administration server for data capture” on page 16.

3. At the login screen, click **OK** to log in as user `ibmuser`.
You cannot change the user ID or password.

4. To view recorded data, click the **Data viewer** tab.
For more information about the fields in this view, see *The monitoring event* in the information center.
5. Select the data store that contains the data that you want to replay.
Only data stores for which the `egForView` property is set in the corresponding `DataCaptureStore` configurable service are available.
6. Optional: Each row of data represents a message. To sort these rows into a particular order, click a column heading. To filter data, click **Filter**  , complete the appropriate fields, then click **Apply filter**  .
Filtering is case sensitive and allows the use of wildcards. To search for an exact match, enclose the search string in quotation marks. You can also search for a sub-string, and you can use an asterisk (*) to match zero or more characters.
7. Optional: You can download a bit stream by clicking the arrow () in the Bitstream column.
8. Select one or more rows to replay to a particular destination.
9. Click **Select one or more rows to replay**  .
10. Optional: To replay all selected data at once, ensure that **Replay as batch** is selected. This check box is selected by default.
If you clear this check box, messages are replayed one by one. If you sorted the messages, they are started in the specified order.
11. Optional: To close the view when all data has been replayed successfully, select **Close on success**. This check box is cleared by default.
12. In the Replay data dialog box, select the destination for your data, then click the arrow () to replay selected data to that destination.
The list of available destinations corresponds to the destinations that you defined in instances of the `DataDestination` configurable service.

Results:

The selected data is sent to the specified destination. Messages that are not delivered are listed in the Replay data dialog box. You can replay all undelivered messages or a single message. To replay a single message, click **Replay** () in the Single Replay column.

Next:

When you have replayed data to a WebSphere MQ queue, you can modify the original message flow, or create a new flow, with an MQInput node that gets messages from the queue.

DataDestination configurable service properties

The following table describes the properties that you can set for the DataDestination configurable service. Changes to these properties take immediate effect and do not require a restart of the broker.

Supplied configurable services that are created for each broker	Properties for each configurable service that is defined	Description of properties
WMQDestination	egForReplay	This property identifies the execution group to use for processing data that is replayed to this destination. The default value is default. This property is required; if you do not set a value, or if you set an invalid value, and you try to replay to this destination, you get a runtime error.
	endpoint	This property identifies the destination to which you want to replay data. The value of the property takes the form: <code>wmq:/msg/queue/queueName@queueManagerName</code> where <i>queueName</i> identifies the destination queue and <i>queueManagerName</i> identifies the queue manager that owns the queue. The default value is <code>wmq:/msg/queue/[QUEUE]@[QMGR]</code> . You must replace <i>[QUEUE]</i> and <i>[QMGR]</i> with the names of your queue and queue manager. If the value is not set, or if the queue or queue manager does not exist, and you try to replay to this destination, you get a runtime error.
	endpointType	This property identifies the type of the target destination. The default value for the property is WMQDestination. This property must be set to the default value.

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 information 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 information. The furnishing of this information does not give you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.*

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

*Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 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 information. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this information at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

*IBM United Kingdom Laboratories,
Mail Point 151,
Hursley Park,
Winchester,
Hampshire,
England
SO21 2JN*

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Programming License Agreement, or any equivalent agreement between us.

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

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

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 includes examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not

been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

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

(C) *(your company name)* *(year)*. Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. *_enter the year or years_*. All rights reserved.

Trademarks

IBM, the IBM logo, and ibm.com[®] are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at Copyright and trademark information at www.ibm.com/legal/copytrade.shtml.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft and Windows 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, or service names may be trademarks or service marks of others.

