

IBM WebSphere Transformation Extender



IBM WebSphere MQ Adapter

Version 8.1

Note

Before using this information, be sure to read the general information in "Notices" on page 47.

October 2006

This edition of this document applies to WebSphere Transformation Extender, 8.1 and to all subsequent releases and modifications until otherwise indicated in new editions.

To send us your comments about this document, e-mail DTX_doc_feedback@us.ibm.com. We look forward to hearing from you.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© **Copyright International Business Machines Corporation 2006. All rights reserved.**

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

Contents

Chapter 1. IBM WebSphere MQ Adapter overview	1
IBM WebSphere MQ message content	1
Transmission queues.	1
IBM WebSphere MQ client or server adapter	2
System requirements.	2
Client system	2
Server system	2
Chapter 2. Command aliases.	5
GET > Source settings	5
GET > Source > Command setting	5
Source > Transaction > OnSuccess setting.	6
Source > Transaction > OnFailure setting	6
Source> Transaction> Scope setting	6
Source> Transaction> Warnings setting	7
PUT > Target settings	7
PUT > Target > Command setting	7
Target > Transaction > OnSuccess setting	7
Target > Transaction > OnFailure setting	7
Target > Transaction > Scope setting	8
Target > Transaction > Warnings setting	8
PUT > Target > Retry settings	8
Chapter 3. IBM WebSphere MQ Adapter commands	11
Channel definition (-CD)	12
Cluster queue manager name (-CQMN)	13
Examples	13
Complete message (-CMMSG)	14
Convert (-CVT)	14
Correlation ID (-CID)	15
Default header (-DH)	15
Error queue manager name (-EQMN).	16
Error queue name (-EQN)	17
Global transaction management (-GTX)	17
Group message (-GRP)	17
Group message 2 (-GRP2)	18
Example 1.	19
Example 2.	20
Header (-HDR)	20
Hex correlation ID (-HCID)	21
Hex group message ID (-HGRP)	21
Hex message ID (-HMID).	21
Ignore group message (-XGRP)	22
Listen (-LSN)	22
Message buffer size (-MBS)	23
Message format (-FORMAT).	23
Message ID (-MID)	24
MQOPEN options (-MQOO).	25
Packet (-PKT).	26
Quantity (-QTY).	26
Queue manager name (-QMN)	27
Queue name (-QN)	27
Refresh message cursor (-REFRESH)	27
Require all messages (-ALLMSG)	29
Require all segments (-ALLSEG)	30

Trace (-T)	30
Transmission Error queue name (-XEQN)	31
Transmission queue name (-XQN)	31
Version 2 (-V2)	32
Wait Interval (-WI)	32
Chapter 4. Syntax summary	33
Data sources	33
Data targets	33
Wildcard support	33
Chapter 5. Using the messaging adapter	35
Threading for IBM WebSphere MQ	35
Chapter 6. Group message support	37
Physical message	37
Logical message	37
Message group	37
Incomplete message segments	38
Reassembly of segments	38
Retrieval of group messages	38
Chapter 7. Example files	39
Using the sample type tree	39
Mapping message data only	39
Mapping the message descriptor	40
Mapping the IBM WebSphere MQ Link for R/3 header	40
Chapter 8. Troubleshooting	41
Trace log	41
Chapter 9. Return codes and error messages	43
Messages	43
WebSphere MQ and error code relationships	44
Notices	47
Programming interface information	49
Trademarks and service marks	49
Index	51

Chapter 1. IBM WebSphere MQ Adapter overview

The IBM WebSphere MQ Adapter works with IBM WebSphere MQ to accomplish the task of maintaining data compatibility, handling all kinds of data transformations from the simplest to the most complex. The IBM WebSphere MQ Adapter can be used at either the source or target application, or at both. The adapters for IBM WebSphere MQ are available as a client adapter and a server adapter to be used with IBM WebSphere MQ client software and IBM WebSphere MQ server software, respectively.

IBM WebSphere MQ software transports data between applications. For the receiving application to be able to successfully use data, it (the data) must be transformed. IBM WebSphere MQ can successfully move data in heterogeneous computer networks, allowing systems of dissimilar computers to move data to and from each other. Although data successfully arrives at a target, it often cannot be used in its native form and must be transformed into a format acceptable for the receiving application's use.

IBM WebSphere MQ message content

The IBM WebSphere MQ message data content can be any content type. Under some conditions, message data contains a message header known as an IBM WebSphere MQ Message Descriptor structure. For example, if you use the IBM WebSphere MQ Link for R/3, the input message data always contains a header and a header must be created for an output message.

Use the IBM WebSphere MQ Adapter to include the message header in the data using the optional Header adapter command (-HDR). When the command is specified for an input message, the message header appears before the message content. If the command is specified for an output message, the adapter assumes that the first block of data supplied for the target consists of the appropriate size and values of a valid IBM WebSphere MQ Message Descriptor structure. When the Header adapter command (-HDR) is specified, any of the message header components may be included in mapping rules.

For mapping purposes, definitions for the message descriptors and the IBM WebSphere MQ Link for the R/3 header are provided in the **mq.mtt** type tree, located in the **examples\adapters\mq** folder.

Transmission queues

An IBM WebSphere MQ transmission queue is a local queue that stores messages destined for a remote queue. The messages are forwarded to their destination queue through a message channel when a communication program and link are available. The channel provides a one-way link to the remote queue manager. Messages are queued at the transmission queue until the channel can accept them. A message channel agent (channel program) that actually transmits the message must be associated with the transmission queue and the remote queue manager. When the message has been transmitted, it is deleted from the transmission queue. The channel definition must specify a transmission queue name at the transmitting end of the message channel.

The IBM WebSphere MQ Adapter supports the definition of transmission queues for data targets with the Transmission Queue Name adapter command (-XQN) and the Transmission Error Queue Name adapter command (-XEQN).

IBM WebSphere MQ client or server adapter

IBM WebSphere MQ supports both client and server modes either from a single platform or distributed across multiple platforms. To select the correct client or server adapter, you must specify the adapter to use for a source or target in map cards, execution settings, or on the command line.

If you do not know the version of IBM WebSphere MQ software you are running, contact your system administrator.

System requirements

The adapters for the IBM WebSphere MQ messaging system can be installed and operated in a Windows environment. The minimum system requirements and operating system requirements for this adapter are detailed in the release notes. The adapters must be installed on a machine running the IBM WebSphere MQ client or server software.

This section identifies the requirements to install and run the IBM WebSphere MQ Adapter in an environment running IBM WebSphere MQ client software or IBM WebSphere MQ server software.

Client system

IBM WebSphere MQ client system and configuration requirements include:

- IBM WebSphere MQ client software is installed and configured. One or more queue managers and queues must have been defined on the system. For information about how to configure IBM WebSphere MQ, see your IBM WebSphere MQ documentation.
- The access methods for the IBM WebSphere MQ adapter follow standard IBM WebSphere MQ authentication guidelines and do not require any special permissions to work with IBM WebSphere MQ. The adapter requires permission to those IBM WebSphere MQ functions used in the IBM WebSphere MQ Adapter. Although registering the user account as a member of the IBM WebSphere MQ Group (mqm) would simplify the configuration (because of not having to set individual user or group permissions), it is not a requirement. Permissions can be set at the user account level to connect to, get messages from, and put messages on queues. These minimal permissions are: Connect, Put, Get, Browse, and Inq. The Set permission is not required because the maps do not alter queue managers or queues.

Server system

IBM WebSphere MQ server system and configuration requirements include:

- IBM WebSphere MQ server software is installed and configured. One or more queue managers and queues must have been defined on the system. For information about how to configure IBM WebSphere MQ, see your IBM WebSphere MQ documentation.
- The access methods for the IBM WebSphere MQ Adapter follow standard IBM WebSphere MQ authentication guidelines and do not require any special permissions to work with IBM WebSphere MQ. The adapter requires permission to those IBM WebSphere MQ functions used. Although registering the user

account as a member of the IBM WebSphere MQ Group (mqm) would simplify the configuration (because of not having to set individual user or group permissions), it is not a requirement. Permissions can be set at the user account level to connect to, get messages from, and put messages on queues. These minimal permissions are: Connect, Put, Get, Browse, and Inq. The Set permission is not required because the maps do not alter queue managers or queues.

Chapter 2. Command aliases

Adapter commands can be specified by using a command string on the command line or by creating a command file that contains adapter commands. The execution command syntax is:

```
-IM[ alias } card_num
-OM[ alias ] card_num
```

where `-IM` is the Input Source - Override execution command and `-OM` is the Output Target - Override execution command, *alias* is the adapter alias, and *card_num* is the number of the input or output card, respectively. The following table shows the adapter aliases and execution commands.

Adapter	Alias	As Input	As Output
IBM WebSphere MQ (client)	MQSC	-IMMQSC <i>card_num</i>	-OMMQSC <i>card_num</i>
IBM WebSphere MQ (server)	MQS	-IMMQS <i>card_num</i>	-OMMQS <i>card_num</i>

When using an adapter alias with the execution command, the IBM WebSphere MQ Adapter commands can be issued on the command line or in a command file. Use the adapter commands to specify adapter functions such as:

- specifying a particular message identifier
- allowing output data to be broken up into multiple messages
- retrieving a logical message from a source queue with a correlation identifier

For example, to override the adapter commands defined in output card 1, the command string for the adapter in an MQSeries server environment might be:

```
-IMMQS1 '-QMN queuemgr -QN topqueue -T'
```

This sample execution command string specifies the queue manager named **queuemgr** and the queue named **topqueue** as the source for this card's data. The `-T` command dictates that a trace file be created to report adapter activity information, recording the events that occur while the adapter is retrieving this data.

GET > Source settings

Use the Map Designer **GET** settings to configure options for input map cards that use the IBM WebSphere MQ (client) or IBM WebSphere MQ (server) adapters. For the **Source** setting, select the data source as either **MQSeries (client)** or **MQSeries (server)** to use the IBM WebSphere MQ Adapters.

GET > Source > Command setting

Use the **Command** setting to enter adapter commands.

Source > Transaction > OnSuccess setting

Use the **OnSuccess** setting with adapter sources to prevent deletion of the source message(s) from the server from which the messages originate. The default value is **Delete**.

Value Description

Keep Keep messages on the source queue.

Delete Delete messages on the source queue depending upon the map completion status and the value of **OnFailure**.

For example, if the **Source** → **Transaction** → **OnSuccess** setting is **Delete** and the **Source** → **Transaction** → **OnFailure** setting is **Rollback** and the map fails, the message is rolled back and is left on the input queue. If the setting is **Rollback** and the map succeeds, the transaction is committed.

If **Rollback** is *not* set, there is no IBM WebSphere MQ transaction syncpoint control activation. The message remains deleted, regardless of the completion status of the map.

Source > Transaction > OnFailure setting

Use the **OnFailure** setting with adapter sources to select the rollback or commit behavior if the map does not successfully complete. The default value is **Rollback**.

Value Description

Rollback

If the map does not complete successfully, sources retrieved through the messaging adapter are not deleted.

Commit

The adapter behaves as if the map was successful and the action specified by the **OnSuccess** setting is taken.

For example, if the **Source** → **Transaction** → **OnSuccess** setting is **Delete** and the **Source** → **Transaction** → **OnFailure** setting is **Rollback** and the map succeeds, the messages processed on the input queue are deleted. If the map fails, the messages are rolled back and left on the input queue.

However, if the **Source** → **Transaction** → **OnSuccess** setting is **Delete** and **Source** → **Transaction** → **OnFailure** setting is **Commit**, there is no IBM WebSphere MQ transaction syncpoint control activation. The message remains deleted, regardless of whether the map is completed. The point is that if the **Source** → **Transaction** → **OnFailure** setting is not **Rollback**, a message can be deleted even if a map fails.

Source> Transaction> Scope setting

Use the **Scope** setting to specify when to check for success (**OnSuccess**) or failure (**OnFailure**) at the completion of the processing of either a map, burst, or card. This is so that the rollback and retry actions can be performed as specified. The default value is **Map**.

Value Description

- Map** Check for success or failure at the completion of each map. If the map successfully completes, use the **OnSuccess** setting. If the map fails, use the **OnFailure** setting.
- Burst** Check for success or failure at the completion of each burst. If the burst is successful, use the **OnSuccess** setting. If the burst fails, use the **OnFailure** setting.
- Card** Check for success or failure at the completion of each card. If the card is successfully processed, use the **OnSuccess** setting. If the card fails, use the **OnFailure** setting.

Source > Transaction > Warnings setting

Use the **Warnings** setting to determine whether to fail the map or ignore warning conditions for the specific input map card. The warning messages are written to the execution audit log. The default value is Ignore.

Value Description

- Ignore** Ignore warnings for this map card.
- Fail** Fail the map upon receiving a warning for this map card.

Warning messages have a positive return code.

PUT > Target settings

Use the Map Designer **PUT** settings to configure options for output map cards that use the IBM WebSphere MQ (client) or IBM WebSphere MQ (server) adapters. Use the **PUT > Target** setting to select the data target. Select **MQSeries (client)** or **MQSeries (server)** to use the IBM WebSphere MQ Adapters.

PUT > Target > Command setting

Use the **PUT > Target > Command** setting to enter adapter commands.

Target > Transaction > OnSuccess setting

Use the **OnSuccess** setting with adapter targets to prevent a message from being unnecessarily created during any processes. The default value is Create.

Value Description

- Create** Upon successful completion of the burst, map, card, or rule (determined by **Scope** settings), send the message.

CreateOnContent

Upon successful completion of the burst, map, card, or rule (determined by **Scope** settings), send the message only if it has content.

Target > Transaction > OnFailure setting

Use the **OnFailure** setting with adapter targets to select the rollback or commit behavior if the map does not complete successfully. The default value is Rollback.

Value Description

Rollback

If the map does not complete successfully, the message is not sent.

Commit

The adapter behaves as if the map was successful and the action (**Create** or **CreateOnContent**) specified by the **OnSuccess** setting is taken.

Target > Transaction > Scope setting

Use the **Scope** setting to specify when to check for success (**OnSuccess**) or failure (**OnFailure**) of a card so that the actions specified with the **OnSuccess/OnFailure** settings can be performed. The default value is **Map**.

Value	Description
-------	-------------

Map	Check for success or failure at the completion of each map. If the map completes successfully, use the OnSuccess setting. If the map fails, use the OnFailure setting.
------------	--

Burst	Check for success or failure at the completion of each burst. If the burst is successful, use the OnSuccess setting. If the burst fails, use the OnFailure setting.
--------------	---

Card	Check for success or failure at the completion of each card. If the card is processed successfully, use the OnSuccess setting. If the card fails, use the OnFailure setting.
-------------	--

Target > Transaction > Warnings setting

Use the **Warnings** setting to determine whether to fail the map or ignore warning conditions for the specific output map card. The warning messages are written to the execution audit log. The default value is **Ignore**.

Value	Description
-------	-------------

Ignore	Ignore warnings for this map card.
---------------	------------------------------------

Fail	Fail the map upon receiving a warning for this map card.
-------------	--

Warnings have a positive return code.

PUT > Target > Retry settings

Use the **Retry** settings to specify the adapter response to a failure to establish a connection to a queue manager or queue. **Retry** settings include **Switch**, **MaxAttempts**, and **Interval**.

If adapter errors occur, such as failure to attach to a queue and retry is specified (**Switch** is set to **ON**), the adapter process is retried at the interval specified in the **Interval** setting, up to as many times as requested in the **MaxAttempts** setting.

Target > Retry > Switch setting

Use the **Switch** setting to determine whether to activate **Retry** for the specific adapter (**ON**) or not (**OFF**). The default value is **OFF**.

Value	Description
-------	-------------

ON	This activates Retry for the specific adapter.
-----------	---

OFF	Retry is inactive.
------------	---------------------------

Target > Retry > MaxAttempts setting

Use the **MaxAttempts** setting to specify the maximum number of times that **Retry** is attempted. The default value is 0.

Value	Description
-------	-------------

0 - <i>value</i>	
------------------	--

	This is an integer representing the maximum number of times that Retry is attempted.
--	---

Target > Retry > Interval setting

Use the **Interval** setting to specify the number of seconds to wait before attempting **Retry** again. The default value is 0.

Value	Description
-------	-------------

0 - <i>value</i>	
------------------	--

	This is an integer representing the number of seconds to wait before attempting Retry again.
--	---

Chapter 3. IBM WebSphere MQ Adapter commands

The following table lists valid commands for the IBM WebSphere MQ Adapter, the command syntax, and whether the command is supported (✓) for use with sources, targets, or both:

Name	Syntax	Source	Target
Channel Definition (-CD) ¹ (use with Client side only)	-CD <i>channel_name/transport_type/ connection_name</i>	✓	✓
Cluster Queue Manager Name (-CQMN)	-CQMN [<i>cluster_manager_name</i>]		✓
Complete Message (-CMMSG)	-CMMSG	✓	
Convert (-CVT)	-CVT	✓	
Correlation ID (-CID)	-CID [<i>correlation_ID</i>]	✓	✓
Default Header (-DH)	-DH		✓
Error Queue Manager Name (-EQMN)	-EQMN <i>error_queue_mgr</i>	✓	✓
Error Queue Name (-EQN)	-EQN <i>error_queue_name</i>	✓	✓
Global Transaction Management (-GTX)	-GTX	✓	✓
Group Message (-GRP)	-GRP [<i>group_ID</i>]	✓	
Group Message 2 (-GRP2)	-GRP2	✓	
Header (-HDR)	-HDR	✓	✓
Hex Correlation ID (-HCID)	-HCID <i>hex_correlation_ID</i>	✓	✓
Hex Group Message ID (-HGRP)	-HGRP <i>hex_group_ID</i>	✓	
Hex Message ID (-HMID)	-HMID <i>hex_message_ID</i>	✓	✓
Ignore Group Message (-XGRP)	-XGRP	✓	
Listen (-LSN)	-LSN {0 S <i>wait_time_in_sec</i> }	✓	
Message Buffer Size (-MBS)	-MBS <i>bytes</i>	✓	
Message Format (-FORMAT)	-FORMAT <i>message_format</i>		✓
Message ID (-MID)	-MID [<i>message_ID</i>]	✓	✓
MQOPEN Options (-MQOO)	-MQOO <i>option_number</i>	✓	✓
Packet (-PKT)	-PKT <i>packet_size</i>		✓
Quantity (-QTY)	-QTY { <i>value</i> S}	✓	
Queue Manager Name (-QMN)	-QMN <i>queue_manager</i>	✓	✓
Queue Name (-QN)	-QN <i>queue_name</i>	✓	✓
Refresh Message Cursor (-REFRESH)	-REFRESH [<i>reset_time</i>]	✓	
Require All Messages (-ALLMSG)	-ALLMSG	✓	
Require All Segments (-ALLSEG)	-ALLSEG	✓	
Trace (-T)	-T[E S] [<i>full_path</i>]	✓	✓

Name	Syntax	Source	Target
Transmission Error Queue Name (-XEQN)	-XEQN <i>x_err_queue_name</i>	✓	✓
Transmission Queue Name (-XQN)	-XQN <i>x_queue_name</i>		✓
Version 2 (-V2)	-V2	✓	✓
Wait Interval (-WI)	-WI {0 S <i>wait_interval_in_milliseconds</i> }	✓	

¹The use of wildcards with this IBM WebSphere MQ Adapter command is supported.

Channel definition (-CD)

Use the optional Channel Definition adapter command (-CD) to specify the client-connection channel definition, which is used by the MQ Client adapter at runtime to establish the connection to the queue manager.

Note: The Channel Definition adapter command (-CD) can be used only with the IBM WebSphere MQ Client adapter. (The connection will fail if the IBM WebSphere MQ Server adapter is used.)

If this command is used, the MQSERVER, MQCHLLIB, and MQCHLTAB environment variables are ignored by the adapter for connecting to the WebSphere MQ Queue Manager.

By defining this adapter command, it is possible to connect to multiple queue managers from the same map using the MQ Client adapter, because each connection will be specified using a different -CD adapter command.

-CD *channel_name/transport_type/connection_name*

Option Description

channel_name

Specify the name of the server-connection channel that must be set up on the queue manager to which the WebSphere MQ Client adapter is connecting.

transport_type

Specify the name of the communication protocol used to establish the connection with the queue manager on the MQ server. The following values are valid:

- LU62
- TCP
- NETBIOS
- SPX
- DECNET

It is the user's responsibility to correctly specify this value. The chosen transport protocol must be supported in the underlying WebSphere MQ environment.

connection_name

Specify the name of the connection. The format of this value depends on the communication protocol that was selected with *transport_type*. For

example, for the TCP communication protocol, the connection name consists of the IP address and the port number on which the MQ listener on MQ server host is waiting for the client requests.

For example:

```
-CD CHANNEL1/TCP/127.0.0.1(1414)
```

This command specifies the following:

- **CHANNEL1** is the name of the server-connection channel.
- **TCP** is the name of the communication protocol used to establish the connection.
- **127.0.0.1(1414)** is the name of the connection.

Cluster queue manager name (-CQMN)

Use the Cluster Queue Manager Name adapter command (-CQMN) to put messages to a queue accessible from a cluster queue manager.

```
-CQMN [cluster_manager_name]
```

where *cluster_manager_name* is the optional name of the cluster queue manager upon which the target queue exists. If *cluster_manager_name* is not specified, IBM WebSphere MQ selects the queue manager from the cluster.

To put a message to a specific instance of a cluster queue (that is, a queue instance that resides on a particular queue manager), specify the name of that queue manager as the argument for the Cluster Queue Manager Name adapter command (-CQMN). This forces the local queue manager to send the message to the specified destination queue manager.

The Cluster Queue Manager Name adapter command (-CQMN) must be accompanied by the Queue Name adapter command (-QN) and the local Queue Manager Name adapter command (-QMN).

If the argument to -QN is the name of a cluster queue and the -CQMN argument is not supplied, the actual destination of messages sent using the queue handle returned by the MQOPEN call is chosen by the queue manager (or by a cluster workload exit if one exists) according to the value of the DefBind queue attribute. This attribute may have one of two values:

- If MQOO_BIND_ON_OPEN is specified, the queue manager selects a particular instance of the cluster queue during the processing of the MQOPEN call, and all messages put using this queue handle are sent to that instance.
- If MQOO_BIND_NOT_FIXED is specified, the queue manager may choose a different instance of the destination queue (residing on a different queue manager in the cluster) on each successive MQPUT call that uses this queue handle.

For more information about setting these options, see "MQOPENoptions (-MQOO)"

Examples

To put a message to a queue named **q1** (which exists on the cluster queue manager **cqm1**) where the adapter is connecting to the cluster queue manager **cqm_default**, enter:

```
-QMN cqm_default -QN q1 -CQMN cqm1
```

To put a message to a queue named **q2** residing on a cluster queue manager, for which the local queue manager of the cluster is **cqm_default**, enter:

```
-QMN cqm_default -QN q2 -CQMN
```

The Cluster Queue Manager Name adapter command (-CQMN) applies only to data targets. IBM WebSphere MQ does not support gets (data sources) from remote queues not residing on the local queue manager, which include cluster queues.

Complete message (-CMSG)

Use the Complete Message adapter command (-CMSG) for data sources to specify retrieval of complete logical messages within group messages.

```
-CMSG
```

The Complete Message adapter command (-CMSG) specifies that only a complete logical message can be returned by the MQGET call. If the logical message is segmented, the queue manager reassembles the segments and returns the complete logical message.

The Complete Message adapter command (-CMSG) is the only adapter command that causes the queue manager to reassemble message segments. Using the Complete Message adapter command (-CMSG) directs the messaging adapter to set the **MQGMO_COMPLETE_MSG** flag in the **MQGMO.Options** field.

If the queue contains segmented messages with some of the segments missing (perhaps the messages were delayed in the network and have not yet arrived), use the Complete Message adapter command (-CMSG) to prevent retrieval of segments belonging to incomplete logical messages.

For information about incomplete message groups contributing to the size of **CurrentQDepth**, see "Incomplete message segments" .

The Complete Message adapter command (-CMSG) implies that all segments in a logical message must be available for retrieval. Therefore, it is not necessary to specify the Require All Segments adapter command (-ALLSEG).

Convert (-CVT)

Use the Convert adapter command (-CVT) for data sources to enable the IBM WebSphere MQ server queue manager to perform a data conversion based upon the client-server coded character set identifier (CCSID) settings, or to execute data conversion user exits. For example, you can use this command to convert from the EBCDIC character set to the ASCII character set to implement distributed messaging using IBM WebSphere MQ.

```
-CVT
```

To satisfy all the requirements for IBM WebSphere MQ when using this command, see your IBM WebSphere MQ documentation.

Correlation ID (-CID)

Use the Correlation ID adapter command (-CID) to specify a particular correlation identifier (CID) for a source or target. Use this adapter command for a data source to retrieve messages from a queue with a specific correlation ID. You can also use it for a data target to assign a correlation ID to a message when placing it on a queue. A correlation identifier can be up to 24 bytes long and is expected to contain printable characters.

```
-CID [correlation_ID]
```

Spaces can also be specified in values. However, if the value contains a space, it must be enclosed with double quotation marks as shown in the example below:

```
-CID "my ID"
```

The double quotation mark character can also be used as part of a value. To do this, the double quotation mark value must be represented by a pair of double quotation marks. For example, if you wanted to specify a correlation ID of **My "New" ID**, you would need to enter the following:

```
-CID "My ""New"" ID"
```

When a correlation identifier is specified for a source, the adapter retrieves the first message on the queue with that correlation ID. If no argument is supplied, the correlation ID of the first message retrieved is used.

If the Correlation ID adapter command (-CID) is specified and is used with the Message ID adapter command (-MID), the retrieved messages begin with the first message on the queue that has the specified message ID and the correlation ID. Additional messages are retrieved that have the same message ID and correlation ID. If no correlation ID is specified, the message is identified only by the message ID.

When a correlation identifier is specified for a data target, the adapter places the message on the queue with the assigned correlation identifier.

Default header (-DH)

The Default Header adapter command (-DH) specifies that the default IBM WebSphere MQ Message Descriptor (MQMD) is used as the message descriptor for data targets.

```
-DH
```

Any null rules (=NONE) in the mapped header field are ignored and the default MQMD values generated by the queue manager are used. Any non-null header fields are set in the MQMD to replace the default MQMD values. The only exception to this is fields that describe the origin and identity context of the message.

Fields describing the origin context of the message are the following (listed in the order in which they appear in the example type tree, **mq.mtt**):

```
PutApplType  
PutApplName
```

PutDate
PutTime
ApplOriginData

Fields describing the identity context of the message are the following (listed in the order in which they appear in the example type tree, **mq.mtt**):

UserIdentifier
AccountingToken
ApplIdentityData

Do not try to specify values for these fields if you use the **-DH** option in the adapter command. Using that option causes values you provided in these fields to be ignored by the adapter and replaced with the default IBM WebSphere MQ values. If you need to explicitly specify values for these fields, use the **-HDR** option in the adapter command (without the **-DH** option).

The Default Header adapter command (**-DH**) overrides the Header adapter command (**-HDR**).

The appropriate header information must be mapped if the Default Header adapter command (**-DH**) is used. If the header information is not mapped and the Default Header adapter command (**-DH**) is used, the mqseries trace shows that the map fails with the following error:

```
<1476-2248>: [m4mqsPut]
<1476-2248>: | [intm4mqsObtainQueueNode]
<1476-2248>: | | Queue node found: QNAME
<1476-2248>: | [intm4mqsObtainQueueNode] (rc = 0) OK
<1476-2248>: | Adapter error rc = -1011 [Invalid message format]
<1476-2248>: [m4mqsPut] (rc = -1011) *** ERROR ***
<1476-2248>: [m4mqsEndTransaction]
<1476-2248>: | Rolling back unit of work
<1476-2248>: [m4mqsEndTransaction] (rc = 0) OK
```

Error queue manager name (**-EQMN**)

Use the Error Queue Manager Name adapter command (**-EQMN**) for data sources and data targets to specify the name of the queue manager for the error queue.

-EQMN *error_queue_manager_name*

Only use the Error Queue Manager Name adapter command (**-EQMN**) when the Error Queue Name adapter command (**-EQN**) is specified and you want to designate a different queue manager for the error queue than the default. The default queue manager is specified with the Queue Manager Name adapter command (**-QMN**).

Only use this adapter command when using the IBM WebSphere MQ Client adapter; using it with the IBM WebSphere MQ Server adapter causes an error. This is because only the IBM WebSphere MQ Client application can be connected to more than one queue manager at the same time.

Error queue name (-EQN)

Use the Error Queue Name adapter command (-EQN) for data sources and data targets to specify the error queue to which messages are copied when errors occur. If a message has the rollback option set and a map does not successfully complete, the message is copied to the designated error queue. The original message is removed from the source queue.

`-EQN error_queue_name`

The name of the error queue must be specified. By default, the queue manager used is the one specified with the Queue Manager Name adapter command (-QMN).

Global transaction management (-GTX)

Use the Global Transaction Management adapter command (-GTX) to indicate that the transactions for this card (whether input or output) should be processed as global transactions.

`-GTX`

For more information about global transaction management, see the Global Transaction Management documentation.

Group message (-GRP)

Use the Group Message adapter command (-GRP) for data sources to specify retrieval of messages from a single group from the source.

`-GRP [group_ID]`

Option Description

group_ID

This is the GroupId field of the MQ Message Descriptor (MQMD)

For example, to specify a message group with the GroupId of **GroupA**, the syntax would be:

`-GRP GroupA`

If the Quantity adapter command (-QTY) is not specified, only one message belonging to the group will be retrieved. If *group_ID* is not specified, the first message group encountered is considered the active message group and is retrieved. You can explicitly specify the number of messages to retrieve using the Quantity adapter command (-QTY). The group name has a maximum size of 24 bytes and may contain spaces. If the value contains a space, it must be enclosed with double quotation marks as shown in the example below:

`-GRP "Group A"`

The double quotation mark character can also be used as part of a value. To do this, the double quotation mark value must be represented by a pair of double quotation marks. For example, if you wanted to specify a message group with the GroupId of **Group "A"**, you would need to enter the following:

`-GRP "Group ""A"""`

If the Group Message adapter command (-GRP) is used with an event source queue, the adapter will initially browse the queue and look for a message that

belongs to the group whose *group_ID* was provided. If *group_ID* was not provided, the first message that belongs to any group will be retrieved and the adapter will keep note of the ID of that group. From that point on, only messages that belong to the same group to which the first message belonged will be retrieved.

Group message 2 (-GRP2)

Use the Group Message 2 adapter command (-GRP2) for data sources to specify retrieval of message groups and non-group messages from the source queue.

-GRP2

The -GRP2 adapter command enables better control over group messages than the -GRP adapter command. The biggest difference between the -GRP2 and -GRP adapter commands is that -GRP2 can be used to process multiple message groups on the source queue. In addition to this, message groups are processed as a unit with -GRP2 and not as individual messages.

While the -GRP adapter command detects a group on the queue and then starts retrieving messages that belong only to that group, the -GRP2 adapter command detects a group on the queue, retrieves all the messages from that group in a single block of data, and keeps looking for other message groups on the queue and other individual non-group messages.

The following restrictions apply to the -GRP2 adapter command:

- It has no arguments.
- It can only be used on input (input cards and GET map function).
- It cannot be combined with the -GRP adapter command.
- It assumes usage of the -XGRP, -ALLSEG, and -ALLMSG adapter commands. Whether explicitly specified or not, these three adapter commands will be automatically included by the -GRP2 adapter command.
- It can be used with the -EQMN and -EQN adapter commands. In case of an error, all messages from the retrieved groups as well as all retrieved non-group messages are moved to the specified error queue.
- in the Launcher scenario, it can be combined with the -REFRESH adapter command. This is particularly useful when there is a possibility that an incomplete message group gets skipped by the queue cursor in order to obtain another complete message group (or non-group message), and the skipped group later becomes complete.

The following table lists some examples of the -GRP2 adapter command usage.

Presume that the following content of queue **QUEUE1** is defined under queue manager **QMNAME**:

Message ID	Message in a group?	Group ID	Logical Sequence Number	Last message in the group?	Segmented message?	Offset	Last Segment?	Data
1	NO							PhysMess01
2	YES	1	1	NO	NO			PhysMess02
3	YES	1	2	YES	NO			PhysMess03
4	NO							PhysMess04

Message ID	Message in a group?	Group ID	Logical Sequence Number	Last message in the group?	Segmented message?	Offset	Last Segment?	Data
5	YES	1	3	NO	NO			PhysMess05
6	YES	2	1	NO	NO			PhysMess06
7	NO							PhysMess07
8	NO							PhysMess08
9	YES	2	2	NO	YES	0	NO	PhysMess09
10	YES	2	2	NO	YES	10	NO	PhysMess10
11	YES	2	2	NO	YES	20	YES	PhysMess11
12	YES	2	3	YES	NO			PhysMess12
13	YES	3	1	YES	NO			PhysMess13
14	NO							PhysMess14

There are few important things to note in this table:

- Physical messages on positions 1, 4, 7, 8 and 14 are logical messages that do not belong to any group.
- Physical messages on positions 2, 3 and 5 are logical messages that belong to a group with the group ID equal to "1".
- Physical messages on positions 6, 9, 10, 11 and 12 belong to a group with the group ID equal to "2". Physical messages on positions 6 and 12 are logical messages, and physical messages on positions 9, 10 and 11 are message segments that together form a single logical message. Physical message on position 11 is the last segment of the logical message. Physical message on position 12 is the last logical message of the group.
- Physical message on position 13 is a logical message and it belongs to a group with the group ID equal to "3". This message is the only logical message in the group.

Presume that a map has one input card and one output card. The input card is of "WebSphere MQ" type, and the output card is of "File" type. The data content obtained in the input card is stored to a file specified in the output card.

Example 1

For example, enter the following command:

```
-QMN QMNAME1 -QN QUEUE1 -GRP2 -QTY 2
```

The data returned will be:

```
PhysMess01PhysMess02PhysMess03PhysMess05
```

Explanation:

-QTY 2 required two message units to be retrieved from the queue. The first message on the queue is PhysMess01. This message does not belong to a group and is returned as a single message. It contributes to one unit in the -QTY 2 value.

The next message on the queue is PhysMess02. This message belongs to a complete group with the group id "1". The adapter retrieves the whole group (messages

PhysMess02, PhysMess03 and PhysMess05) and returns them as a chunk of data that contributes to the second unit in the -QTY 2 adapter command.

If the -GRP2 adapter command was omitted, the returned data would be:

```
PhysMess01PhysMess02
```

Example 2

For example, enter the following command:

```
-QMN QMNAME1 -QN QUEUE1 -GRP2 -QTY 4
```

The data returned will be:

```
PhysMess01PhysMess02PhysMess03PhysMess05PhysMess04  
PhysMess06PhysMess09PhysMess10PhysMess11PhysMess12
```

Explanation:

-QTY 4 required four message units to be retrieved from the queue. The adapter first retrieved PhysMess01, and then PhysMess02, PhysMess03 and PhysMess05, as in the previous example. After that, there are two more message units that need to be retrieved.

The third message unit is message PhysMess04. It is a non-group message, and the adapter retrieves it after retrieving the message group "1". Finally, the fourth unit is message group "2" that consists of the following physical messages:

```
PhysMess06, PhysMess09, PhysMess10, PhysMess11 and PhysMess12
```

If -GRP2 adapter command was omitted, the returned data would be:

```
PhysMess01PhysMess02PhysMess03PhysMess04
```

Header (-HDR)

Use the Header adapter command (-HDR) to specify the inclusion of the IBM WebSphere MQ message descriptor in the source and target data.

```
-HDR
```

When Header adapter command (-HDR) is used for a data source, the message descriptor appears first, before the message content. Use the Header adapter command (-HDR) to include any of the message descriptor components in mapping rules.

When the Header adapter command (-HDR) is used for a data target, the message descriptor must be the first component of the output type. Mapping rules for the message descriptor components must be provided.

Included in the **examples** folder is a type tree that can be used when specifying this command. For more information, see "Example files".

For data targets only, if the Header adapter command (-HDR) is used in conjunction with the Message ID adapter command (-MID), the Header command (-HDR) takes precedence over the Message ID command, which is ignored. If the Header adapter command (-HDR) is used in conjunction with the Version 2 adapter command (-V2), then the version information mapped in the header is ignored.

Hex correlation ID (-HCID)

Use the Hex Correlation ID adapter command (-HCID) to specify correlation identifiers using hexadecimal pair notation. This passes to the adapter the exact correlation ID to locate, even if that correlation ID contains non-printable values, spaces, or even null characters.

-HCID hex_correlation_ID

Option Description

hex_correlation_ID

This is the hexadecimal correlation identifier.

For example, the ASCII correlation ID My Msg is represented as 4D79204D7357 when specified with the Hex Correlation ID adapter command (-HCID). The space character must be encoded in the hex pair notation.

The hexadecimal correlation ID wildcard notation is not encoded in the hex pair notation. The wildcard character * is entered literally. For example, the hexadecimal pair notation forx*z is 78*7A, not 782A7A. The Hex Correlation ID (-HCID) cannot be used with the Correlation ID adapter command (-CID).

Hex group message ID (-HGRP)

Use the Hex Group Message ID adapter command (-HGRP) for data sources to specify group message identifiers using hexadecimal pair notation. This passes to the adapter the exact group message ID to locate, even if that group message ID contains non-printable values, spaces, or even null characters.

-HGRP hex_group_ID

Option Description

hex_group_ID

This is the hexadecimal group message identifier.

For example, the ASCII group message ID My Msg is represented as 4D79204D7357 when specified with the Hex Group Message ID adapter command (-HGRP). The space character must be encoded in the hex pair notation.

The hexadecimal group message ID wildcard notation is not encoded in the hex pair notation. The wildcard character * is entered literally. For example, the hexadecimal pair notation forx*z is 78*7A, not 782A7A. The Hex Group Message ID (-HGRP) cannot be used with the Group Message adapter command (-GRP).

Hex message ID (-HMID)

Use the Hex Message ID adapter command (-HMID) to specify message identifiers using hexadecimal pair notation. This passes to the adapter the exact message ID to locate, even if that message ID contains non-printable values, spaces, or even null characters.

-HMID hex_message_ID

Option Description

hex_message_ID

This is the hexadecimal message identifier.

For example, the ASCII message ID My Msg is represented as 4D79204D7357 when specified with the Hex Message ID adapter command (-HMID). The space character must be encoded in the hex pair notation.

The hexadecimal message ID wildcard notation is not encoded in the hex pair notation. The wildcard character * is entered literally. The hexadecimal pair notation for x*z is 78*7A, not 782A7A.

The Hex Message ID (-HMID) cannot be used with the Message ID adapter command (-MID).

Ignore group message (-XGRP)

Use the Ignore Group Message adapter command (-XGRP) for data sources to ignore the group messages on the queue and not to treat them as individual, physical messages. The default behavior of the IBM WebSphere MQ Adapter is to process all physical messages, including group messages, as individual events.

-XGRP

The Ignore Group Message adapter command (-XGRP) causes the adapter to ignore any group messages on that queue for the source event card. If both groups and non-group messages are to be processed as events from the same source queue, the Group Message 2 (-GRP2) command can be used as described on page Group message 2 (-GRP2).

If the Group Message adapter command (-GRP) is used with an event source queue, the adapter ignores non-group messages on that queue. It only processes the messages from the group specified with the -GRP adapter command, or, if no group was specified, the first group that it locates on the queue.

If neither -XGRP nor -GRP is used, all messages are treated as non-group messages and processed individually.

Listen (-LSN)

Use the Listen adapter command (-LSN) for data sources to dictate the length of time (in seconds) that the IBM WebSphere MQ queue manager allows the adapter to wait for a message to be received. Map execution is suspended until the message is received or until the specified time lapses.

-LSN {0|S|*wait_time_in_sec*}

Option Description

0 Do not wait at all. If there is no input available, the adapter does not wait for more messages.

S This is an infinite wait time.

wait_time_in_sec

This is the number of seconds that the adapter waits (in seconds) for a message.

If the Listen adapter command (-LSN) is not issued and the message is not available for the adapter to retrieve immediately, the map completes, but returns the following warning in the audit log:

(2033) MQSeries Warning: No message available.

Message buffer size (-MBS)

Use the Message Buffer Size adapter command (-MBS) to specify the size of the message buffer. The adapter allocates this message buffer size for retrieving messages from a source queue. If the buffer size is insufficient, the buffer is automatically reallocated to accommodate the messages.

-MBS *bytes*

Option Description

bytes This is the size of the message buffer in bytes. The default buffer size is 4096 bytes.

Message format (-FORMAT)

Use the Message Format adapter command (-FORMAT) for data targets to specify the format of the application data in the message. Either pre-existing (defined by IBM WebSphere MQ) or custom (user-defined) message formats can be used.

-FORMAT *message_format*

Option Description

message_format

A message format, either pre-existing (predefined by IBM WebSphere MQ) or custom (user-defined). The format can contain up to eight (8) characters. If less than eight, it will be padded with spaces.

For example, to specify MQFMT_STRING as the format of the message, the following adapter command should be used:

-FORMAT MQSTR

where MQSTR is the value of the MQFMT_STRING message format. For information about other built-in message formats, see your IBM WebSphere MQ documentation.

You must always specify a value when using the -FORMAT adapter command. Failure to do so will result in map failure and a corresponding error message is generated.

If you use a user-defined format, you must also provide the corresponding data-conversion exit so that receiving applications can read the message using the MQGMO_CONVERT option. For WebSphere Transformation Extender applications, this means using the -CVT adapter command for the IBM WebSphere MQ inputs.

If the message format is not provided (either with the usage of the -FORMAT adapter command or through the message descriptor type tree with the -HDR/-DH adapter command), the MQFMT_STRING message format is used as a default value.

To specify a format value that has spaces interleaved with text (or all spaces), use double quotes.

For example:
-FORMAT "ab cd"

or

to specify a blank format (all spaces), you can insert eight (8) spaces in double quotes:

```
-FORMAT "          "
```

Note that the value of the -FORMAT adapter command is the value of the predefined format constant, not the name of the constant.

For example, use:

```
-FORMAT MQSTR
```

or:

```
-FORMAT "MQSTR  "
```

and not:

```
-FORMAT MQFMT_STRING
```

MQFMT_STRING is the name of the predefined IBM WebSphere MQ message format constant, and its value is "MQSTR "

Message ID (-MID)

Use the Message ID adapter command (-MID) to specify a particular message ID for a data source or target.

```
-MID [message_ID]
```

Option Description

message_ID

A message ID can be up to 24 bytes long and may contain spaces.

If the value contains a space, it must be enclosed with double quotation marks as shown in the example below:

```
-MID "my ID"
```

The double quotation mark character can also be used as part of a value. To do this, the double quotation mark value must be represented by a pair of double quotation marks. For example, if you wanted to specify a message ID of **My "New" ID**, you would need to enter the following:

```
-MID "My ""New"" ID"
```

When using the Launcher, the message ID can be specified with wildcard notation. For more information about wildcard notation, see the Launcher documentation.

When a message ID is specified for a data source, the adapter retrieves the first message with the designated message ID. If no argument is supplied, the message ID of the first message retrieved will be used.

When a message ID is specified for a target, the adapter places the message on the queue with that message ID assigned. If this command is not specified for a target, the message is put on the queue using a unique message identifier generated by the queue manager.

For targets only, if the Header adapter command (-HDR) is used in conjunction with the Message ID adapter command (-MID), the Header adapter command (-HDR) takes precedence over the Message ID adapter command (-MID), which is ignored.

MQOPEN options (-MQOO)

Use the MQOPEN Options adapter command (-MQOO) to explicitly set the numeric value of the Options parameter in the MQOPEN API call made by the adapter to the MQSeries library when opening a queue.

`-MQOO option_number`

Option Description

option_number

Valid values are any positive integer that represents the value of the option parameter in the MQOPEN API call made when opening a queue.

As an example of the required option usage, if you want to specify an open option for the queue to be used on the output side with an MQOO_FAIL_IF QUIESCING property, you must specify more than the MQOO_FAIL_IF QUIESCING value (8192). You must also add the MQOO_OUTPUT option value (16), which indicates that the queue will be opened for output (storing messages). The correct syntax for this example would be:

`-MQOO 8208`

because $MQOO_OUTPUT + MQOO_FAIL_IF_QUIESCING = 16 + 8192 = 8208$.

The valid value is important; otherwise, the MQOPEN, MQGET, or MQPUT calls made by the adapter may fail. For the options and their corresponding values that can be used with the -MQOO adapter command, see the MQ Open Options section in the *IBM MQ Series Application Programming Reference*.

The MQOPEN Options adapter command (-MQOO) must be accompanied by the Queue Name adapter command (-QN) and the Queue Manager Name adapter command (-QMN).

For example, to specify the target queue named **target_q** under the queue manager **qmgr** for message outputs (where 16 is the numeric value of the MQOO_OUTPUT option), the syntax would be:

`-QMN qmgr -QN target_q -MQOO 16`

To specify the source queue named **source_q** under the queue manager **qmgr**, for message input in browse mode with shared access (where 10 is the numeric superposition of the MQOO_INPUT_SHARED (2) and MQOO_BROWSE (8) options), the syntax would be:

`-QMN qmgr -QN source_q -MQOO 10`

Packet (-PKT)

Use the Packet adapter command (-PKT) for data targets to break output data into multiple messages and to put these messages on the target queue with the same message identifier.

-PKT *packet_size*

Option Description

packet_size

The packet size must be specified in units of 1000 bytes with a minimum of one unit.

For example, to specify a packet size of 3000 bytes, the syntax would be:

-PKT 3

If one of the following conditions is met:

- If the -HDR or -DH command is used on output, and the message header has the **Version** field set to a value of one (1).
- If the Version 2 (-V2) command is used (which enforces the use of the Version 1 header from MQSeries 2.x). For more information about this command, see "Version 2 (-V2)" .

The created physical messages will be independent logical messages that will have the same message id. These messages cannot be automatically reassembled on input, however, this can be achieved through the proper map design.

On the other hand, if BOTH of the following conditions are met:

- If -HDR or -DH is used on output and if the message header has **Version** field set to a value of two (2).
- If the Version 2 (-V2) command is omitted (which would enforce the use of Version 1 header from MQSeries 2.x). For more information about this command, see "Version 2 (-V2)" .

The created physical messages will be message segments of the original message. The Offset property of the message will provide information about the relative position of each of the segments in the original message. In this case, -CMSSG adapter command can be used on input to automatically reassemble the original message.

The Packet adapter command (-PKT) is ignored for a message source.

Quantity (-QTY)

Use the Quantity adapter command (-QTY) to specify the number of messages to retrieve from the source queue, without regard to message IDs. If the Quantity adapter command (-QTY) is not specified, the default value is 1.

-QTY {*value*|S}

Option Description

S This returns all messages on the queue.

value This is a positive integer representing the number of concurrent multiple messages to be retrieved.

For example, to specify a quantity of ten messages, the syntax would be:

```
-QTY 10
```

The Quantity adapter command (-QTY) defines the maximum number of messages across all bursts for a map.

The number of bursts in a single map execution can be controlled using the combination of values set in the Map Designer **Source** → **FetchAs** → **FetchUnit** setting and the Quantity adapter command (-QTY).

When using the Quantity adapter command (-QTY) with the **S** option through either the Launcher or the Command Server, the Listen adapter command (-LSN) is also required with a value specified other than **S**.

Queue manager name (-QMN)

Use the Queue Manager Name adapter command (-QMN) to specify the name of the queue manager on which the queue (specified by the -QN command) exists.

```
-QMN queue_manager_name
```

Option Description

```
queue_manager_name
```

This is the name of the queue manager. This name is case-sensitive.

When the Queue Manager Name adapter command (-QMN) is not specified, the default queue manager is used.

Queue name (-QN)

Use the Queue Name adapter command (-QN) to specify the name of the queue from which the message(s) are to be retrieved or to which the message(s) are to be sent.

```
-QN queue_name
```

Option Description

```
queue_name
```

This is the name of the queue. This name is case sensitive.

The Queue Name adapter command (-QN) is a required adapter command.

Refresh message cursor (-REFRESH)

Use the Refresh Message Cursor adapter command (-REFRESH) to ensure that higher-priority messages are enqueued in front of any existing messages with a lower priority so that the higher-priority messages can trigger the associated maps to run. This command is only used when using an IBM WebSphere MQ event input with the Launcher. If messages of a higher priority are enqueued after messages of a lower priority, the default behavior is that the higher priority messages would not trigger the maps.

```
-REFRESH [reset_time]
```

Option Description

reset_time

This is a non-negative integer defining the number of seconds that elapse before the message cursor is reset to the head of the queue.

The Refresh Message Cursor adapter command (-REFRESH) resets the message cursor of the source queue message event listener to the head of the queue after every listener cycle (detection of all messages currently on the queue) is completed. You can optionally specify a non-negative integer argument (*reset_time*) that defines the number of seconds that elapse before the message cursor is reset to the head of the queue. If *reset_time* is set to zero, the message cursor will be constantly reset to the beginning of the queue after each event. Thus, the higher priority messages are processed as soon as possible. However, frequent repositioning of the cursor will degrade overall performance.

The Refresh Message Cursor adapter command (-REFRESH) is only needed when using an IBM WebSphere MQ event input with the Launcher, and only in two specific situations:

- When messages of higher priority are placed at the beginning of the queue that are not recognized because the message cursor has already begun moving towards the bottom of the queue.
- When there is a possibility of the message cursor skipping the uncommitted messages on the queue. As an example, suppose that two applications, APP1 and APP2, simultaneously place messages on an empty queue and the following sequence of operations occurs (a possibility because APP1 and APP2 are independent and are executing concurrently):

```
T0: APP1 PUT(M1)
T1: APP2 PUT(M2)
T2: APP2 COMMIT(M2)
T3: APP1 COMMIT(M1)
```

After T1 occurs, both messages have already been placed on the queue (with M1 in front of M2). However, they are not being recognized by the message cursor. After T2 occurs, the message cursor recognizes message M2 and it positions itself on that message. After T3 occurs, message M1 is also committed, but it is now beyond the message cursor. The message cursor will never be able to recognize that message unless it is repositioned to the beginning of the queue. Use the Refresh Message Cursor adapter command (-REFRESH) to accomplish this repositioning.

Using the Refresh Message Cursor adapter command (-REFRESH) may degrade event notification and mapping performance and should only be used when necessary. The adapter must be careful not to trigger the map more than once for the same input message. To ensure this, the adapter maintains a list of messages that have already been seen by the cursor. Each time the cursor is repositioned to the beginning of the queue, each message that the cursor finds on the queue is compared with all of the messages in this list. Only in situations in which this message is not in the adapter list is that message considered to be a new message to be used for the event notification. As part of this process, this new message will be added to the adapter list. To ensure that this list does not grow infinitely, there are two important limitations for the input cards that use the -REFRESH adapter command:

- The **Rollback** option for the **OnFailure** setting should not be used.
- The **Keep** option for the **OnSuccess** setting should also not be used.

If either of these two options is specified, running the Launcher for a long period of time with many messages being processed may cause excessive memory usage and may slow down processing time significantly, and eventually may cause the system to stop.

The reason for this is that messages once seen by the cursor are added to the internal list, and as long as they are not removed from the source queue, they are also not removed from the internal adapter's list so that they are not processed more than once. Messages are freed from the internal list only after they have been removed from the source queue.

In situations in which it is necessary to use the Refresh Message Cursor adapter command (-REFRESH) *and* to preserve the messages for the failed map, it is recommended that you use Error Queue Name adapter command (-EQN) in combination with -REFRESH adapter command for the input card. In this case, the **Source** → **Transaction** → **OnSuccess** setting should be **Delete** and the **Source** → **Transaction** → **OnFailure** setting should be **Rollback**. The messages for the map failure will be placed on the error queue and removed from the source queue. Thus, messages will be preserved (on the error queue), and no memory leak will occur because it will be possible to remove messages from the internal adapter's list upon moving them to the error queue. Note, however, that if, for some reason, messages cannot be stored on the error queue (for example, PUT is disabled for the error queue), the messages will remain on the input queue and in the internal adapter's list which again provides the possibility of a memory leak.

Require all messages (-ALLMSG)

Use the Require All Messages adapter command (-ALLMSG) for data sources to specify retrieval of messages in a group only when all messages in that group are available.

The use of the -ALLMSG adapter command automatically sets the requirement that group messages are returned in their logical order and that segments of logical messages are returned in the order of their offsets.

This must be considered when specifying additional selection criteria (such as the -MID or -CID adapter commands). It may be possible that the message which satisfies this additional criteria is not the message with the smallest sequence number and/or the smallest offset in its group, which will prevent it from being retrieved from the queue.

-ALLMSG

If the source queue contains an incomplete message group, the Require All Messages adapter command (-ALLMSG) prevents retrieval of any messages belonging to the incomplete groups.

For information about incomplete message groups contributing to the size of **CurrentQDepth**, see "Incomplete message segments".

The Require All Messages adapter command (-ALLMSG) implies that all segments in a logical message must be available for retrieval. Therefore, it is not necessary to specify the Require All Segments adapter command (-ALLSEG) in conjunction with -ALLMSG.

Require all segments (-ALLSEG)

Use the Require All Segments adapter command (-ALLSEG) for data sources to specify retrieval of segments in a logical message group only when all segments in the logical message are available.

The use of the -ALLSEG adapter command automatically sets the requirement that group messages are returned in their logical order and that segments of logical messages are returned in the order of their offsets.

This must be considered when specifying additional selection criteria (such as the -MID or -CID adapter commands). It may be possible that the message which satisfies this additional criteria is not the message with the smallest sequence number and/or the smallest offset in its group, which will prevent it from being retrieved from the queue.

-ALLSEG

If the source queue contains segmented messages with some missing segments, the Require All Segments adapter command (-ALLSEG) prevents retrieval of the message segments belonging to incomplete logical messages.

For information about incomplete message groups contributing to the size of **CurrentQDepth**, see "Incomplete message segments".

Trace (-T)

Use the Trace adapter command (-T) to produce a trace file that contains information about connections to IBM WebSphere MQ, message sizes, IBM WebSphere MQ error messages, and adapter error messages.

The default filename is **mqstrace.log** and it is created in the map directory unless otherwise specified.

For z/OS Batch and CICS environments, **mqstrace.log** does not conform to the naming convention. For z/OS Batch and CICS environments, the trace output is written to the dataset indicated by the ddname **MQTRACE** that is (in most cases) defined in the JCL as : //MQTRACE DD SYSOUT=*

If the trace file does not already exist, it is created. If it already exists it is automatically appended.

-T[E|S] [*full_path*]

Option Description

- E** Error mode. Produces a trace file containing only the adapter errors that occurred during map execution.
- S** Summary mode. Use this option to reduce the size of the trace file. The summary trace file excludes polling information and function entries and exits-reporting only minimal information, such as the activity that occurs within a function (when applicable).

full_path

Create a trace file with the specified name in the specified directory. By default, the directory is where the map is located and the file name is **mqstrace.log**.

You can override the adapter command line trace options dynamically using the Management Console. For detailed information, see "Dynamic Adapter Tracing" in the *Launcher* documentation .

Transmission Error queue name (-XEQN)

Use the Transmission Error Queue Name adapter command (-XEQN) for data sources and targets to specify the name of the transmission queue for the remote error queue to which messages are copied when errors occur. For more information about using the IBM WebSphere MQ Adapter with IBM WebSphere MQ transmission queues, see "Transmission Queues" .

`-XEQN x_err_queue_name`

Option Description

x_err_queue_name

This is the name of the transmission queue. The transmission queue name is case-sensitive.

For example, to specify that the transmission queue `xmit.q` should be used as the transmission queue for a remote error queue whose remote queue definition is named `error.q`, the syntax would be:

```
-QMN qmlocal -QN qlocal -EQN error.q -XEQN xmit.q
```

In this syntax, **qmlocal** is the name of the queue manager and **qlocal** is the name of the queue that is used as data source or data target in the map.

If a local transmission queue specified with the Transmission Error Queue Name adapter command (-XEQN) exists, it is used. If it does not exist, the queue specified by the IBM WebSphere MQ DefaultXmitQ attribute (DEFXMITQ in MQSC) on the local queue manager is used.

Transmission queue name (-XQN)

Use the Transmission Queue Name adapter command (-XQN) for data targets to specify the name of the transmission queue that holds messages destined for a remote queue. For more information about using the IBM WebSphere MQ adapter with IBM WebSphere MQ transmission queues, see "Transmission queues" .

`-XQN x_queue_name`

Option Description

x_queue_name

This is the name of the transmission queue. The transmission queue name is case sensitive.

For example, to specify that the local transmission queue **xmit.q** be used as the data target for a remote queue named **dest.q** and the local queue manager of **qmloc1**, the syntax would be:

```
-QMN qmloc1 -QN dest.q -XQN xmit.q
```

For this command to work as intended, the local transmission queue must have the same name as the remote queue manager. If the local object **xmit.q** is not defined, the queue manager default transmission queue will be used.

If a local transmission queue exists with the same name as the transmission queue specified with the Transmission Queue Name adapter command (-XQN), the local transmission queue is used. If the transmission queue specified with the Transmission Queue Name adapter command (-XQN) does not exist, the queue specified by the MQSeries DefaultXmitQ attribute (DEFXMITQ in MQSC) on the local queue manager is used.

Version 2 (-V2)

Use the Version 2 adapter command (-V2) to set all message descriptor and option structure version fields to `_VERSION_1`, which corresponds to IBM WebSphere MQ version 2.x.

-V2

If the Version 2 adapter command (-V2) is not specified, the default is `_VERSION_2`, which corresponds to IBM WebSphere MQ version 5.x.

The Version 2 adapter command (-V2) is used with sources and targets.

If the Header adapter command (-HDR) is used in conjunction with the Version 2 adapter command (-V2), then the version information mapped in the header is ignored.

Wait Interval (-WI)

Use the Wait Interval adapter command (-WI) to set the length of time (in milliseconds) that the IBM WebSphere MQ Adapter queue manager allows the adapter to wait for a message to be received. Map execution is suspended until the message is received or until the specified time lapses.

`-WI {0|S|wait_interval_in_milliseconds}`

Option Description

0 Do not wait at all. If there is no input available, the adapter does not wait for more messages.

S This is an infinite wait time.

wait_interval_in_milliseconds

This is the number of milliseconds that the adapter waits for a message.

If the Wait Interval adapter command (-WI) is not issued, the default time that the adapter waits for a message is once for each second.

If the Wait Interval adapter command (-WI) is issued, the Launcher cannot be paused or shut down for the length of time you specified in the *wait_interval_in_milliseconds* command option. Do not specify the S command option and attempt to pause or shut down the Launcher. This will cause the Launcher to stop responding.

Chapter 4. Syntax summary

Data sources

Syntax of the IBM WebSphere MQ Adapter commands used for data sources:

```
-QN queue_name
[-ALLMSG]
[-ALLSEG]
[-CD channel_name/transport_type/connection_name]
[-CID [correlation_ID] | -HCID hex_correlation_ID]
[-CMSG]
[-CVT]
[[-EQMN error_queue_manager_name] -EQN error_queue_name]
[[-GRP [group_ID] ] [-HGRP hex_group_ID] [-XGRP]]
[-GTXT]
[-HDR]
[-LSN {0|S|wait_time_in_sec}]
[-MBS]
[-MID [message_ID] | -HMID hex_message_ID]
[-MQ00 option_number]
[-QMN queue_manager_name]
[-QTY value]
[-REFRESH [reset_time] ] [-T[E|S] [full_path]]
[-V2]
[-WI {0|S|wait_interval_in_milliseconds}]
[-XEQN x_err_queue_name]
```

Data targets

Syntax of the IBM WebSphere MQ Adapter commands used for data targets:

```
-QN queue_name [-CD channel_name/transport_type/connection_name]
[-CQMN [cluster_manager_name]]
[-GTXT]
[-HDR | -DH
  | [-MID [message_ID] | -HMID hex_message_ID]
  | [-CID [correlation_ID] | -HCID hex_correlation_ID]
  | [-FORMAT message_format]
]
[-MQ00 option_number]
[-PKT packet_size]
[-QMN queue_manager_name]
[-T[E|S] [full_path]]
[-V2]
[-XEQN x_err_queue_name]
[-XQN x_queue_name]
```

Wildcard support

The use of wildcards with the IBM WebSphere MQ Adapter is supported for the following adapter commands:

- Message ID (-MID)
- Hex Message ID (-HMID)
- Correlation ID (-CID)
- Hex Correlation ID (-HCID)
- Group Message (-GRP)

- Hex Group Message ID (-HGRP)

For more information about using wildcards, see the Launcher documentation.

Chapter 5. Using the messaging adapter

When using the IBM WebSphere MQ Adapter for a source, messages are served by default from the queue in a first-in-first-out (FIFO) manner unless you specify a particular message using a message, correlation, or group identifier. When using the adapter for a target, messages are placed on the queue you specify.

When running the Launcher with the IBM WebSphere MQ Adapter, for two messages on the same queue to be distinguishable from each other, they must have unique combinations of the message ID, correlation ID, group ID, and sequence number and offset. At least one of these properties must be different for two messages on the same queue to ensure proper functioning of the Launcher.

By default, IBM WebSphere MQ assigns unique message IDs to the messages. However, in situations in which messages are specifically assigned message IDs before they are placed on the triggered queue (the queue for which a watch is defined), it is necessary to ensure that either the application-defined custom message IDs are unique or that the other message qualifiers (correlation ID, group ID, sequence number, and offset) must be used to make those messages unique.

When running maps through the Command Server, this limitation does not exist because the messages are read from the queue only once and are processed immediately. In the Launcher, however, the messages are first browsed (trigger events) and then processed in the executed (triggered) maps.

When using the Launcher to execute maps using the IBM WebSphere MQ Adapter, ensure that the Launcher service is always started by a user account that is registered in the MQM group.

Threading for IBM WebSphere MQ

The IBM WebSphere MQ Adapter is thread-safe on the Windows, AIX, Sun, and HP platforms. However, due to the IBM WebSphere MQ default system configuration parameters for queues and queue managers, the default IBM WebSphere MQ resources may be insufficient to support multithreading with the IBM WebSphere MQ Adapter.

When operating in the multithreaded mode, the Launcher executes one thread for each map being concurrently executed. In addition to the map thread, the IBM WebSphere MQ queue manager connection for each distinct queue manager on a source or target card requires a separate connection thread. Because IBM WebSphere MQ does not allow connection sharing between threads, each map thread maintains its own set of connection threads.

In addition to the map and connection threads, the IBM WebSphere MQ system spawns threads to service internal requests and processes. The result is that, as the number of concurrent maps increases, the use of IBM WebSphere MQ and system resources multiplies. When the system resources exceed a threshold based upon the IBM WebSphere MQ and system configuration settings, the excessive thread context-switching and system memory resource allocation can significantly reduce the efficiency of the overall IBM WebSphere MQ message throughput.

In addition to the reduction in efficiency, it is possible that when the IBM WebSphere MQ systems resources are exceeded, the IBM WebSphere MQ behavior can become unpredictable and can result in messages being ignored on event queues.

To address these limitations, the following recommendations should be implemented:

- Limit the number of concurrent messaging adapter threads to 20. This is a configuration parameter (MaxThreads) set in the **dtx.ini** file.
- Ensure that the maximum number of handles (queue and queue manager connections) supported by the queue manager exceeds the total number of messaging adapter and IBM WebSphere MQ threads. This value may be changed using an MQSC command at the **runmqsc** prompt as:

```
alter qmgr MAXHANDS(2048)
```

- Set the maximum number of uncommitted messages allowed by the queue manager to an acceptably large value:

```
alter qmgr MAXUMSGS(100000)
```

- Set the maximum number of messages allowed on a queue to an acceptable value:

```
alter ql(my.queue.name) MAXDEPTH(200000)
```

The limit on non-z/OS Batch and CICS systems is 640K. The queue manager may enforce smaller actual limits based upon DASD resource limitations.

Chapter 6. Group message support

The adapter for IBM WebSphere MQ supports retrieval of IBM WebSphere MQ group messages as data source objects. The group message feature allows multiple messages to be placed on a message queue while retaining an organizational connection through the **GroupId** field of the MQ Message Descriptor (MQMD).

- A message group consists multiple logical messages.
- A logical message consists of one or more physical messages.
- When a logical message contains more than one physical message, the logical message is segmented. Each physical message is a segment.
- Each logical message has the same **GroupId** and a distinct **MsgSeqNumber**.
- Each physical message has distinct MQ Message Descriptor (MQMD).
- Each segment has the same **GroupId** **MsgSeqNumber** and a distinct **Offset**.

Physical message

A physical message is the smallest unit of information that can be placed on or removed from a queue. A physical message often corresponds to the information specified in or retrieved from a single MQPUT or MQGET call. Each physical message has a unique message descriptor (MQMD).

Generally, physical messages are distinguished by differing values for the message identifier (**MsgId** field in the MQMD) although this is not enforced by the queue manager.

Logical message

A logical message is a single unit of application information. In the absence of system constraints, a logical message is the same as a physical message. Logical messages may be extremely large; system constraints may make it advisable or necessary to split a logical message into two or more physical messages known as segments.

A logical message that has been segmented consists of two or more physical messages that have the same non-null group identifier (**GroupId** field in MQMD) and the same message sequence number (**MsgSeqNumber** field in MQMD). The segments are distinguished by differing values for the segment offset (**Offset** field in MQMD), which gives the offset of the data in the physical message from the start of the data in the logical message.

Message group

A message group is a set of one or more logical messages that have the same non-null group identifier. The logical messages in the message group are distinguishable by differing values for the message sequence number. The message sequence number is an integer in the range 1 to n , where n is the number of logical messages in the group. If one or more of the logical messages is segmented, the group will contain more than n physical messages in the group.

Incomplete message segments

If the queue contains segmented messages with some of the segments missing (perhaps the messages were delayed in the network and have not yet arrived), specifying the Complete Message adapter command (-CMMSG) prevents the retrieval of segments belonging to incomplete logical messages.

However, those message segments belonging to incomplete logical messages still contribute to the value of the **CurrentQDepth** queue attribute. This segmentation could result in no retrievable logical messages, even when **CurrentQDepth** is greater than zero.

Reassembly of segments

Each physical message that is a segment has a unique message descriptor. For segments that comprise a single logical message, most of the fields in the message descriptor are the same for all segments in the logical message. Usually, it is only the **MsgId**, **Offset**, and **MsgFlags** fields that differ between segments in the logical message. However, when segments take different paths through the network, and some of those paths have MCA sender conversion enabled, it is possible for the **CodedCharSetId** and **Encoding** fields to differ between segments when the segments eventually arrive at the target queue.

A logical message consisting of segments in which the **CodedCharSetId** and/or **Encoding** fields differ cannot be reassembled by the queue manager into a single logical message. Instead, the queue manager reassembles and returns the first few consecutive segments at the start of the logical message having the same character-set identifiers and encodings, and the MQGET call completes with completion code MQCC_WARNING and reason code MQRC_INCONSISTENT_CCIDS or MQRC_INCONSISTENT_ENCODINGS, as appropriate.

When the queue manager reassembles a logical message, the message descriptor is returned with the values from the message descriptor for the first segment; the only exception is the **MsgFlags** field, which the queue manager sets to indicate that the reassembled message is the only segment.

Retrieval of group messages

The IBM WebSphere MQ Adapter selects group messages for retrieval based upon three possible selection criteria:

- group ID (GroupID)
- message ID (MsgId)
- correlation ID (CorrelId)

When the Group Message adapter command (-GRP) is specified, the GroupId is used as the sole selection criteria for group message retrieval. If the Message ID adapter command (-MID) or the Correlation ID adapter command (-CID) is specified, the respective selection criteria are also used.

Be sure to organize message segments and groups on output when you specify the **GroupId**, **MsgSequenceNumber**, **Offset**, and **MsgFlags** fields in the MessageDescriptor type. Improper use of these fields could result in incomplete message groups and illogical messages on the output queue.

Chapter 7. Example files

The IBM WebSphere MQ type tree (named **mq.mtt**) is provided in the **examples\adapters\mq** folder. Types describing different IBM WebSphere MQ message formats are provided so that you can easily select the message types needed for input or output or the descriptor that you need to include with any message data you want.

When mapping information to or from the message descriptor, you must include the message descriptor in your data by using the Header adapter command (-HDR).

Using the sample type tree

Depending on how you want to map, select one of the **Message** types as input or output or add a message content type as a **MessageData** component.

The **mq.mtt** type tree is arranged so that you can easily integrate the provided definitions with any message data that you want to define.

The **mq.mtt** type tree has definitions for:

- **MQSeries Header MessageDescriptor**

The MessageDescriptor Header type defines the components and format of the message descriptor.

- **MQSeries Header MessageDescriptor2**

The MessageDescriptor2 Header type defines the components and format of the message descriptor for IBM WebSphere MQ 5.x.

- **MQSeries Header MQSLink4R3**

The MQSLink4R3 Header type defines the components and format of the IBM WebSphere MQ Link for R/3 Header. When the IBM WebSphere MQ Link for R/3 is used for source data, the input message data always contains a header; however, you must create the header for an output message. The MQSeries Link for R/3 Header is not optional. It always appears in your input data and it must be included when mapping output data.

The **mq.mtt** type tree is organized so that you can drag the **MQSeries** root type to another type tree to easily use any of the IBM WebSphere MQ-specific headers. For example:

Mapping message data only

By default, the adapter does not use message descriptors. On input, it returns only the body of the received message, and on output, it treats the data that it receives as the message body.

Even if you do not need to use message descriptors, you can still use this same example tree to define the content of the message.

To use the type tree for mapping message data only:

1. Define the message content for your message (shown below as **Message** group under **CreditInfo** category).

2. Add it as a component of **MessageData**. By default, **MessageData** contains **Text** item that represents plain textual message content. The **Text** item should be replaced with your own message content type, such as **Message** group under **CreditInfo** category in this example.
3. Use **MessageData** as an input or output card type.

Mapping the message descriptor

If the Header (-HDR) or Default Header (-DH) adapter command is specified, the message descriptor appears in the data, immediately preceding the message data when used for a source.

For a target, you must provide the message descriptor (all components (for -HDR), or components of interest [for -DH]), which is included as the first part of your output data.

The following example shows a sample type tree definition:

Mapping the IBM WebSphere MQ Link for R/3 header

If you use IBM WebSphere MQ to send and receive data from an SAP R/3 system, you may want to use the IBM WebSphere MQ Link for R/3. This link communicates with R/3 using inbound and outbound server queues.

When using the IBM WebSphere MQ Link for R/3, message data is in IDoc format, prefixed by the IBM WebSphere MQ Link for R/3 Header.

The **mq.mtt** type tree includes the definition of this header. You also need the IDoc Importer to automatically generate the types that describe the format of your IDoc data.

Chapter 8. Troubleshooting

For information about error codes and messages returned by the adapters, see "Return codes and error messages".

Trace log

Use the Trace adapter command (-T) to create a trace file to report adapter activity information, recording events that occur while the adapter is retrieving and sending data. The trace command produces a log file with the default name **mqstrace.log** in the map execution directory.

The trace file contains information about connections to IBM WebSphere MQ, message sizes, IBM WebSphere MQ error messages, and adapter error messages.

Chapter 9. Return codes and error messages

Return codes and messages are returned when the particular activity completes. Return codes and messages may also be recorded as specified in the audit logs, trace files, execution summary files, and so on.

Messages

The following is a listing of all the codes and messages that can be returned as a result of using the IBM WebSphere MQ Adapter for sources or targets.

In addition to these codes, the IBM WebSphere MQ Adapter can return WebSphere MQ error codes. Consult the WebSphere MQ documentation for descriptions of these codes.

Adapter return codes with positive numbers are warning codes that indicate a successful operation. Adapter return codes with negative numbers are error codes that indicate a failed operation.

Return Code

Message

- 1001 Trace file could not be accessed
- 1002 Adapter command is incorrect
- 1003 Queue could not be opened
- 1004 Message could not be stored on the queue
- 1005 Message could not be retrieved from the queue
- 1006 Connection could not be closed
- 1007 Global transaction could not be started
- 1008 Transaction could not be rolled back explicitly
- 1009 Transaction could not be committed
- 1010 Queue could not be closed
- 1011 Invalid message format
- 1012 Transaction monitor not supported
- 1013 Error queue operation could not complete
- 1014 Unable to Begin Commitment Control (HP NonStop ZLE)
- 1015 Unable to Backout (HP NonStop ZLE)
- 1016 Unable to Commit (HP NonStop ZLE)
- 1017 Segmentation not supported (HP NonStop ZLE)

See "WebSphere MQ and error code relationships" for more information.

WebSphere MQ and error code relationships

The IBM WebSphere MQ Adapter returns various error codes when it encounters a problem during code execution. Some of these error codes are related to the errors returned by the MQ calls (such as MQCONN, MQGET) and some of them are not.

The following table lists all MQ calls made by the adapter, and how errors from these calls are mapped to error codes:

IBM WebSphere MQ Native Function Action Taken by Adapter

- MQCONN** If MQCONN fails while trying to connect to the queue manager specified in the -QMN adapter command, the error code returned by the adapter is MPIRC_E_BAD_CONNECTION (-13). However, if MQCONN fails while trying to connect to the error queue manager specified by the -EQMN adapter command, the adapter returns error code MPIRC_E_MQS_EQUEUE_OPERATION (-1013) to indicate that the error occurred while transferring messages to the error queue.
- MQOPEN** If MQOPEN fails while trying to open the queue specified in the -QN adapter command, the error code returned by the adapter is MPIRC_E_MQS_OPEN_QUEUE (-1003). However, if MQOPEN fails while trying to open the error queue specified by the -EQMN adapter command, the error code returned by the adapter is MPIRC_E_MQS_EQUEUE_OPERATION (-1013) to indicate that the error occurred while transferring messages to the error queue. Additionally, if the MQOPEN call issued for opening the queue manager object fails (this is done as part of the connection validation process), the returned error code is MPIRC_E_BAD_CONNECTION (-13) to indicate to the Resource Manager that the connection is invalid.
- MQGET** If MQGET call fails and this error cannot be corrected or handled by the adapter, the adapter returns error code MPIRC_E_MQS_GET_MESSAGE (-1005). Additionally, if MQGET call fails inside of the listener thread, and the MQ reason code is MQRC_CONNECTION_BROKEN (2009L), the adapter returns error code MPIRC_E_BAD_CONNECTION (-13) to signal that the connection is broken and needs to be reestablished.
- MQPUT** If MQPUT call fails and this error cannot be corrected or handled by the adapter, the adapter returns error code MPIRC_E_MQS_PUT_MESSAGE (-1004).
- MQCMIT** If MQCMIT call fails while committing the messages upon successful map completion, the adapter returns error code MPIRC_E_MQS_COMMIT (-1009).
- MQBACK** The adapter does not return an error code as a result of MQBACK failure. This is because if the MQBACK is called, that means that the map has already failed for another reason.
- MQBEGIN**
This function is called only when the global transactions are used (-GTX adapter command). If MQBEGIN fails, the adapter returns MPIRC_E_MQS_BEGIN_GTX (-1007). On NonStop ZLE, if the BEGINTRANSACTION() call fails, the adapter returns error code MPIRC_E_MQS_BEGINUOW_ERR (-1014).
- MQCLOSE**
A failure in MQCLOSE does not cause the map to fail (but a textual message indicating this will be traced). The only time a failure in MQCLOSE results in an error code returned by the adapter, is when the queue manager

connection is being validated. If MQCLOSE fails for the previously opened queue manager object for validation, the adapter returns error code MPIRC_E_BAD_CONNECTION (-13).

MQDISC If MQDISC fails, the returned error code is MPIRC_E_MQS_DISCONNECT (-1006).

MQINQ If MQINQ fails, the adapter does not return any error.

MQCONN

If MQCONN fails while trying to connect to the remote queue manager specified in the (-QMN) adapter command, the error code returned by the adapter is MPIRC_E_BAD_CONNECTION (-13). Note that the MQCONN command is used only by the IBM WebSphere MQ (client) adapter, and only if the (-CD) adapter command is specified.

MQPUT1, MQSET

Not used by the adapter.

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 grant 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 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

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

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

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

IBM Corporation
577 Airport Blvd., Suite 800
Burlingame, CA 94010
U.S.A.

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

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

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

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

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

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

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

Programming interface information

Programming interface information, if provided, is intended to help you create application software using this program.

General-use programming interfaces allow you to write application software that obtain the services of this program's tools.

However, this information may also contain diagnosis, modification, and tuning information. Diagnosis, modification and tuning information is provided to help you debug your application software.

Warning: Do not use this diagnosis, modification, and tuning information as a programming interface because it is subject to change.

Trademarks and service marks

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States or other countries, or both:

i5/OS
IBM
the IBM logo
AIX
AIX 5L
CICS
CrossWorlds
DB2
DB2 Universal Database
Domino
HelpNow
IMS
Informix
iSeries
Lotus
Lotus Notes
MQIntegrator
MQSeries
MVS
Notes
OS/400
Passport Advantage
pSeries
Redbooks
SupportPac
Tivoli
WebSphere
z/OS

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

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

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

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

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

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

This product includes software developed by the Eclipse Project (<http://www.eclipse.org/>).



IBM WebSphere Transformation Extender, Version 8.1

Index

Special characters

- ALLMSG adapter command 29
- ALLSEG adapter command 30
- CID adapter command 15
- DH adapter command 15
- FORMAT adapter command 23
- GRP adapter command 17
- GRP2 adapter command 18
- GTX adapter command 17
- HCID adapter command 21
- HDR adapter command 20
- HGRP adapter command 21
- HMID adapter command 21
- LSN adapter command 22
- MBS adapter command 23
- MID adapter command 24
- MQOO adapter command 25
- PKT adapter command 26
- QMN adapter command 27
- QN adapter command 27
- QTY adapter command 26
- REFRESH adapter command 27
- T adapter command 30
- V2 adapter command 32
- WI adapter command 32
- XEQN adapter command 31
- XGRP adapter command 22
- XQN adapter command 31

C

- CD adapter command 12
- Channel definition (-CD) 12
- Cluster Queue Manager Name (-CQMN) adapter command 13
- CMSG adapter command 14
- commands 12
 - Cluster Queue Manager Name (-CQMN) 13
 - Complete Message (-CMSG) 14
 - Convert (-CVT) 14
 - Correlation ID (-CID) 15
 - Default Header (-DH) 15
 - Error Queue Manager Name (-EQMN) 16
 - Error Queue Name (-EQN) 17
 - Global Transaction Management (-GTX) 17
 - Group Message (-GRP) 17
 - Group Message 2 (-GRP2) 18
 - Header (-HDR) 20
 - Hex Correlation ID (-HCID) 21
 - Hex Group Message ID (-HGRP) 21
 - Hex Message ID (-HMID) 21
 - Ignore Group Message (-XGRP) 22
 - list of 11
 - Listen (-LSN) 22
 - Message Buffer Size (-MBS) 23
 - Message Format (-FORMAT) 23
 - Message ID (-MID) 24
 - MQOPEN Options(-MQOO) 25
 - Packet (-PKT) 26
 - Quantity (-QTY) 26

commands (*continued*)

- Queue Manager Name (-QMN) 27
- Queue Name (-QN) 27
- Refresh Message Cursor (-REFRESH) 27
- Require All Messages (-ALLMSG) 29
- Require All Segments (-ALLSEG) 30
- Trace (-T) 30
- Transmission Error Queue Name (-XEQN) 31
- Transmission Queue Name (-XQN) 31
- Version 2 (-V2) 32
- Wait Interval (-WI) 32

Complete Message (-CMSG) adapter command 14

Convert (-CVT) adapter command 14

Correlation ID (-CID) adapter command 15

CQMN adapter command 13

CVT adapter command 14

D

Default Header (-DH) adapter command 15

E

EQMN adapter command 16

EQN adapter command 17

error code relationships 44

error messages 43

Error Queue Manager Name (-EQMN) adapter command 16

Error Queue Name (-EQN) adapter command 17

examples

- IBM WebSphere MQ adapter 39

G

GET > Source > Command setting 5

GET > Source settings 5

- Command 5
- OnSuccess 6

Global Transaction Management (-GTX) adapter command 17

Group Message (-GRP) adapter command 17

Group Message 2 (-GRP2) adapter command 18

group message support 37

H

header

- message descriptors 39

Header (-HDR)

- using 40

Header (-HDR) adapter command 20

Hex Correlation ID (-HCID) adapter command 21

Hex Group Message ID (-HGRP) adapter command 21

Hex Message ID (-HMID) adapter command 21

I

IBM WebSphere MQ adapter

- example files 39

Ignore Group Message (-XGRP) adapter command 22

Interval setting 9

L

Launcher

start up requirement in MQSeries 35

Link for R/3 39

Listen (-LSN) adapter commands 22

M

mapping

message data 40

message descriptor 40

MaxAttempts setting 9

Message Buffer Size (-MBS) adapter command 23

message data

mapping 40

message descriptor

mapping 40

Message Format (-FORMAT) adapter command 23

Message ID (-MID) adapter command 24

messages

content of 1

mq.mtt 39

MQOPEN Options(-MQOO) adapter command 25

MQSLink4R3 header 39

mqstrace.log 30, 41

O

OnFailure setting 6, 7

OnSuccess setting 6, 7

P

Packet (-PKT) adapter command 26

PUT > Target > Command settings 7

PUT > Target settings 7

Q

Quantity (-QTY) adapter command 26

Queue Manager Name (-QMN) adapter command 27

Queue Name (-QN) adapter command 27

R

Refresh Message Cursor (-REFRESH) adapter command 27

Require All Messages (-ALLMSG) adapter command 29

Require All Segments (-ALLSEG) adapter command 30

Retry setting 8

return codes 43

S

sample type tree definition 40

Scope setting 6, 8

Source > Transaction settings

OnFailure 6

Scope 6

Warnings 7

Switch setting 8

syntax summary 33

T

Target > Transaction settings

OnFailure 7

OnSuccess 7

Scope 8

Warnings 8

Target settings

Retry 8

Interval 9

MaxAttempts 9

Switch 8

Trace (-T) adapter command 30

trace log 41

Transmission Error Queue Name (-XEQN) adapter

command 31

Transmission Queue Name (-XQN) adapter command 31

type tree 39

V

Version 2 (-V2) adapter command 32

W

Wait Interval (-WI) adapter command 32

Warnings setting 7, 8

wildcards 21, 22



Printed in USA