

WebSphere MQ Integrator - Remote command line utilities

Version 1.0

19 March 2002

Simon Lazenby

Property of IBM

Take Note!

Before using this report be sure to read the general information under "Notices".

First Edition, March 2002

This edition applies to Version 1.0 of *WebSphere MQ Integrator - Remote command line utilities* and to all subsequent releases and modifications unless otherwise indicated in new editions.

© **Copyright International Business Machines Corporation 2002**. All rights reserved. Note to US Government Users -- Documentation related to restricted rights -- Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule contract with IBM Corp.

Table of Contents

Notices	v
Trademarks and service marks	v
Summary of Amendments.....	vi
Preface	vii
Bibliography	viii
Chapter 1. Getting Started	1
What are remote commands	1
Who should use remote commands.....	1
Prerequisites.....	1
Chapter 2. Installation procedures	2
Installing the remote commands	2
Installation on Windows NT and Windows 2000	2
Installation Directories	3
Environment variables	4
Trace log size	4
Chapter 3. Remote command line utilities syntax.....	6
mqsiremotechangetrace.....	6
Required parameters	6
Optional parameters	6
mqsiremotereadlog.....	9
Required parameters	9
Optional parameters	9
mqsiremotereporttrace	11
Required parameters	11
Optional parameters	11
Chapter 4. Using remote commands line utilities.....	13
Chapter 5. User Exits	14
Appendix A. Tracing the remote command.....	15

Appendix B. Performance	16
Appendix C. Broker Communication Routing	17
Default Routing	17
Alternate routing capability	17
Alternate routing prerequisites	17
Appendix D. Message Catalogues.....	18
Appendix E. MQSeries Connection failures.....	19

Notices

The following paragraph does not apply in any 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 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.

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates.

Any reference to an IBM licensed program or other IBM product in this publication is not intended to state or imply that only IBM's program or other product may be used. Any functionally equivalent program that does not infringe any of the intellectual property rights may be used instead of the IBM product.

Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, is the user's responsibility.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, New York 10594, USA.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS-IS. The use of the information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item has been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Trademarks and service marks

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

- IBM
- MQSeries
- WebSphere MQ Integrator

The following terms are trademarks of other companies:

- Windows NT, Windows 2000 Microsoft Corporation

Summary of Amendments

Date	Changes
19 March 2002	Initial release

Preface

This SupportPac provides three command line utilities to assist with remote problem diagnosis within WebSphere MQ Integrator V2.1 brokers.

Bibliography

- *MQSeries. Using Java*, IBM Corporation. SC34-5456-06

Chapter 1. Getting Started

This chapter gives an overview of WebSphere MQ Integrator SupportPac ih05, Remote command line utilities, (henceforth know as remote commands).

What are remote commands

Remote commands are a set of commands that enable a user to remotely access a broker. With these remote commands a user can:

- Retrieve the brokers trace log
- Report a brokers trace settings
- Change a brokers trace settings

Who should use remote commands

Any enterprise that uses a WebSphere MQ Integrator V2.1 Control Center on a machine that is remote from the broker and deploys configuration to a broker (the broker may be on any platform) will benefit by using these remote commands.

These remote commands are customized to use the Control Center installation environment.

Prerequisites

It is recommended that a WebSphere MQ Integrator V2.1 Control Center should be installed on the platform where the remote commands are to be used before to installing this SupportPac. If this is the case then no other prerequisites are required.

If this is not the case then the following software should be installed:

- Java Runtime Environment (JRE), (at least V1.2.2)
- MQSeries SupportPac MA88 - MQSeries classes for Java.

Be aware that these remote commands are targeted at the Windows NT / Windows 2000 platform with a Control Center installed.

Chapter 2. Installation procedures

This chapter describes how to install the remote commands.

Installing the remote commands

This SupportPac is supplied as two compressed format files that are available from the MQSeries Web site, <http://www.ibm.com/software/mqseries/txppacs>

ih05.zip contains the following files:

ih05.jar	This code includes all the runtime support.
ih05ue.jar	This code is supplied as an example user exit.
ih05.properties	This includes the set of messages issued by this SupportPac.
mqsiremotechangeTrace.cmd	The shell script that initiates the change trace command.
mqsiremoteReadLog.cmd	The shell script that initiates the read trace log command.
mqsiremoteReportTrace.cmd	The shell script that initiates the report trace command.
ih05.pdf	This Document, a description of the SupportPac.
MQUserExit.java	Source code for the example user exit ih05ue.jar.

ih05_option_win.zip, WebSphere MQ Integrator v2.1 CSD01 files (second **optional** compressed file):

bipXercesC.dll	Supplied to enable mqsiformatlog.exe
dbghelp.dll	Supplied to enable mqsiformatlog.exe
GenXmlParser2.dll	Supplied to enable mqsiformatlog.exe
MessageServices.dll	Supplied to enable mqsiformatlog.exe
NativePrincipals.dll	Supplied to enable mqsiformatlog.exe
xmlcommon2.dll	Supplied to enable mqsiformatlog.exe
WMQlv210.properties	Supplied to enable mqsiformatlog.exe

Note: The second compressed file is required when only the WebSphere Integrator V2.1 Control Center is installed. These files are for the V2.1 product and enable the V2.1 mqsiformatlog executable to run. Alternatively these files may be copied from a broker installations product directory but only when that broker is on the same platform and at the same release/service level as the Control Center.

Installation on Windows NT and Windows 2000

This section describes how to install the remote commands on the Windows platform.

1. Locate the installation directory for the WebSphere MQ Integrator Control Center. By default this would be "C:\Program Files\IBM\WebSphere MQ Integrator 2.1".

2. Copy file ih05.zip to this directory and decompress it.
3. Delete file ih05.zip.
4. Follow the instructions in the chapter “Using remote commands line utilities”.

If only the WebSphere MQ Integrator V2.1Control Center is installed (not the full broker):

1. Copy file ih05_option_win.zip to same directory mentioned for the file ih05.zip.
2. Decompress file ih05_option_win.zip. (Files will be extracted into temporary directory “ih05\” within this directory).
3. Copy the files in the temporary directory “ih05\” into the Control Centers directories (as indicated by the subfolders within the temporary directory ... bin, book and messages). If any files already exist then you should not overwrite them. These files are not normally present in a “Control Center only installation”. They are normally found within a “broker installation”. If these files already exist then you do not need the files from the compressed file ih05_option_win.zip.
4. Delete the temporary directory “ih05\” and file ih05_option_win.zip.

Note: ih05_option_win.zip contains the set of files required to enable the V2.1 command, mqsiformatlog.exe to run. If the full broker is installed there is no need to decompress this file, doing so could be harmful to those files currently installed. Do not overwrite files that already exist.

An alternative to using ih05_option_win.zip is to copy these files from an existing broker installation product directory that is on the same platform type and at the same release/service level as the Control Center on the machine you are using.

Installation Directories

The SupportPac files are installed in the following directories. Note that install_dir is the directory in which the WebSphere MQ Integrator V2.1 Control Center is installed.

ih05.zip:

Directory	Files
install_dir\bin	mqsiremotechangetrace.cmd mqsiremotereadlog.cmd mqsiremotereporttrace.cmd
install_dir\classes	ih05.jar ih05ue.jar
install_dir\messages	ih05.properties
install_dir\examples\ih05	MQUserExit.java
install_dir\book\ih05	ih05.pdf

ih05_option_win.zip (optional files):

Directory	Files
install_dir\ih05\bin	bipXercesC.dll dbghelp.dll GenXmlParser2.dll MessageServices.dll NativePrincipals.dll xmlcommon2.dll
install_dir\ih05\messages	WMQIv210.properties

Environment variables

After installation, ensure that environment variable MQSI_HOME is set before running the remote commands. This variable is used by the remote commands to locate the installation directory (install_dir). For example:

```
set MQSI_HOME=C:\Program Files\IBM\WebSphere MQ Integrator 2.1
```

Alternatively, modify the files mqsiremotechangetrace.cmd, mqsiremotereporttrace.cmd and mqsiremotereadlog.cmd by hand (there is a commented out line placed in these files as an example).

If this environment variable is not set then the remote command will assume that the product is located in the following directory:

```
C:\Program Files\IBM\WebSphere MQ Integrator 2.1
```

Trace log size

With default settings the maximum trace log size that can be retrieved by the mqsiremotereadlog command is 4 MB. Modification of the following MQSeries resources will allow this to be increased:

- On the machine with the Configuration Manager installed ensure that the Receiver channel for the broker Queue Manager to the Configuration Managers Queue Manager has the maximum message length set to 104857600.

```
runmqsc configmgrQueueManager
```

```
alter channel ('broker.to.configmgr') CHLTYPE(RCVR) MAXMSGL(104857600)
```

- On the machine with the broker installed ensure that the transmission queue to the Configuration Manager Queue Manager has a maximum message length of 104587600. Also ensure that the Sender channel for the broker Queue Manager to the Configuration Manager Queue Manager also has its maximum message length set to 104587600.

```
runmqsc brokerQueueManager
```

```
alter qlocal ('configmgr') MAXMSGL(104857600)
```

```
alter channel ('broker.to.configmgr') CHLTYPE(SDR) MAXMSGL(104857600)
```

Once the two prior steps to increase the MQSeries maximum message length have been carried out and the MQSeries Queue Managers have been restarted, then, the `mqsiremotechangetrace` command can be used to increase the size of the trace file that a broker can return. It is recommended that the trace size specified on the `mqsiremotechangetrace` command not exceed 100 MB. Preferably, the trace size should be just adequate to contain the trace required by the user.

Chapter 3. Remote command line utilities syntax

This chapter describes the syntax for the remote commands:

mqsiremotechangetrace

This command allows a brokers trace settings to be changed.

```

— mqsiremotechangetrace — brokerQueueManagerName — -u —
                                     — -t —
— -e executionGroup —
                                     — -f messageFlow —
                                     — -l level —
— -m mode — -c size — -r —
— -h hostname — -p port — -q configmgrQueueManager —

```

Required parameters

BrokerQueueManagerName

The Queue Manager name of the broker to which the command is directed. This will most likely be the same as the name of the broker. (Please refer to Appendix C. Broker Communication Routing, for details about the communication mechanisms employed.)

-e *executionGroup*

Identifies the execution group for which trace options are to be modified.

Optional parameters

-u Specifies that user trace options are to be modified.

-t Specifies that service trace options are to be modified.

-f *messageFlow*

Identifies the message flow for which trace options are to be modified.

-l *level*

Set the level of the trace. This must be one of:

normal. This provides a basic level of trace information.

none. This sets tracing off.

debug. This provides a more comprehensive trace.

Each component is created with a default value of none. If this parameter is not specified then the current value is unchanged. Once successfully changed, this value is persistent.

-m mode

Indicates the way trace information is to be buffered:

safe. This mode causes trace entries to be written to file when they are generated.

fast. This mode causes trace entries to be buffered, and only written to file in batches.

Each component starts with a default value of safe. If this parameter is not specified the current value is unchanged.

Selecting this option sets tracing for the execution group. Once successfully changed, this value is persistent.

-c size

The size of the trace file in KB (kilobytes). If this parameter is not specified then the current value is left unchanged. Each component starts with a default value of 4096 KB. Specifying this option will reset the value. The maximum value that should be used depends on how the log will subsequently be retrieved, when using the **mqsiremotereadlog** command the maximum value is about 70 MB, (-c 70000).

It is recommended that the size be as small as possible, either by using a low value for this parameter or by using the reset (-r) option on this command to clear the trace log. The benefit of adopting this approach is that the length of time taken to retrieve the trace log (**mqsiremotereadlog**) will be greatly reduced. In addition, the formatting process (**mqsiformatlog**) will consequently be much faster and require less resource to carry out its task.

-r

This option requests that the trace log is reset, that is, all current records are discarded. Use this option when starting a new trace to ensure that all records in the log are unique to the new trace.

-h hostname

The name or IP address of the machine where the MQSeries Queue Manager specified in option '-q' is running.

-p port

The port on which the MQSeries Queue Manager specified in option '-q' is listening for incoming connection requests.

-q configmgrQueueManager

The name of the MQSeries Queue Manager used by the WebSphere MQ Integrator Configuration Manager.

Note: It is not expected that options -h, -p and -q will normally be selected. The expectation is that a WebSphere MQ Integrator Control Center is already installed (and has connected to the Configuration Manager at least once). In this case the command will be able to discover the routing information

required automatically without the user having to provide the options -h, -p or -q. (Please refer to Appendix C. Broker Communication Routing, for details about the communication mechanisms employed.)

mqsiremotereadlog

This command allows a brokers trace log to be retrieved.

```

— mqsiremotereadlog — brokerQueueManagerName — -u —
                                     — -t —
— -e executionGroup —————
                                     — -o outputFileName —
—————
| — -h hostname — | — -p port — | — -q configmgrQueueManager — |

```

Required parameters

brokerQueueManagerName

The Queue Manager name of the broker to which the command is directed. This will most likely be the same as the name of the broker. (Please refer to Appendix C. Broker Communication Routing, for details about the communication mechanisms employed.)

-e *executionGroup*

Identifies the execution group for which the trace log will be retrieved.

Optional parameters

-u Specifies that user trace log will be retrieved.

-t Specifies that service trace log will be retrieved.

-o *outputFileName*

The name of the file into which trace log data will be written. Specifying a full path name will result in the file being created in the directory specified. Specifying just the filename will cause the file to be created in the current working directory. The contents of the file are written in code page UTF-8, which is platform-independent and therefore preserves data such as DBCS characters. A file name must be specified when the log is to be formatted using the **mqsiformatlog** command. Not selecting a filename will cause the trace contents to be written to stdout.

-h *hostname*

The name or IP address of the machine where the MQSeries Queue Manager specified in option '-q' is running.

-p *port*

The port on which the MQSeries Queue Manager specified in option '-q' is listening for incoming connection requests.

-q *configmgrQueueManager*

The name of the MQSeries Queue Manager used by the WebSphere MQ Integrator Configuration Manager.

Note: It is not expected that options -h, -p and -q will normally be required. The expectation is that a WebSphere MQ Integrator Control Center is already installed (and has connected to the Configuration Manager at least once). In this case the command will be able to discover the routing information required automatically without the user having to provide the options -h, -p or -q. (Please refer to Appendix C. Broker Communication Routing, for details about the communication mechanisms employed.)

mqsiremotereporttrace

This command allows a brokers trace settings to be reported.

```

— mqsiremotereporttrace — brokerQueueManagerName — -u —
                                     — -t —
— -e executionGroup —————
                                     — -f messageFlow —
—————
— -h hostname — — -p port — — -q configmgrQueueManager —

```

Required parameters**brokerQueueManagerName**

The Queue Manager name of the broker to which the command is directed. This will most likely be the same as the name of the broker. (Please refer to Appendix C. Broker Communication Routing, for details about the communication mechanisms employed.)

-e executionGroup

Identifies the execution group for which trace options are to be reported.

Optional parameters

-u Specifies that user trace options are to be reported.

-t Specifies that service trace options are to be reported.

-f messageFlow

Identifies the message flow for which trace options are to be reported.

-h hostname

The name or IP address of the machine where the MQSeries Queue Manager specified in option '-q' is running.

-p port

The port on which the MQSeries Queue Manager specified in option '-q' is listening for incoming connection requests.

-q configmgrQueueManager

The name of the MQSeries Queue Manager used by the WebSphere MQ Integrator Configuration Manager.

Note: It is not expected that options -h, -p and -q will normally be required. The expectation is that a WebSphere MQ Integrator Control Center is already installed (and has connected to the Configuration Manager at least once). In this case the command will be able to discover the routing information

required automatically without the user having to provide the options -h, -p or -q. (Please refer to Appendix C. Broker Communication Routing, for details about the communication mechanisms employed.)

Chapter 4. Using remote commands line utilities

The three remote commands supplied are very similar in operation to the local versions of the commands (mqsichangetrace, mqsireadlog and mqsirepotrtrace).

If the broker was created with the following command:

```
mqsicreatebroker MQSI_SAMPLE_BROKER -q MQSI_SAMPLE_QM ...
```

Issuing the following remote command (as long as a WebSphere MQ Integrator Control Center was installed and had been used at least once), would report the trace settings for execution group 'default':

```
mqsiremotereporttrace MQSI_SAMPLE_QM -u -e default
```

Note that the brokers Queue Manager name is specified in the command rather than the brokers actual name.

Similarly, issuing:

```
mqsiremotechangetrace MQSI_SAMPLE_QM -u -e default -l normal -c 4000 -r
```

Would set the trace level to normal, trace log size to 4 MB and also reset the trace log to zero length at the same time.

Following with the command:

```
mqsiremotereadlog MQSI_SAMPLE_QM -u -e default -o trace.xml
```

would retrieve the trace log from the broker.

This trace log could then be browsed with an XML browser or formatted using:

```
mqsiformatlog -i trace.xml -o trace.txt
```

Chapter 5. User Exits

The prerequisite product, MQSeries Classes for Java, has three user exits: security, send and receive. If required, these user exits can be invoked by setting the following environment variables to indicate the location of the user exit code.

For example:

```
set MQSI_SECEXIT=com.ibm.broker.wmqcmds.userexit.MQUserExit
set MQSI_RECEXIT=com.ibm.broker.wmqcmds.userexit.MQUserExit
set MQSI_SENDEXIT=com.ibm.broker.wmqcmds.userexit.MQUserExit
mqsiremotechangetrace MQSI_SAMPLE_QM -u -e default -l normal -m safe
```

In the example all three exits were used. There is no requirement that all three be specified together or that they use the same program. The source code for the above example can be found in file 'example\ih05\MQUserExit.java' and may be altered or replaced as required. The supplied example simply outputs a text message to indicate that it has been called. By default these exits are not enabled.

The environment variables that control the user exits can be added to the three shell scripts (mqsiremotechangetrace.cmd, mqsiremotereadlog.cmd and mqsiremotereporttrace.cmd) if required, a commented out line making reference to the supplied example exists for each user exit.

Please ensure that the CLASSPATH environment variable is correctly set if the supplied example user exits are not used.

Appendix A. Tracing the remote command

If a remote command does not operate normally or fails unexpectedly then its operation can be traced in order to discover the reason for this behavior. Trace can be enabled by setting the environment variable `MQSI_UTILITY_TRACE` before running the remote command. This environment variable can be set to either 'normal' or 'debug'. Trace is written to a file called 'cmdtrace.log'.

```
set MQSI_UTILITY_TRACE=normal
```

```
mqsiremotereadlog MQSI_SAMPLE_QM -t -e default -o trace.xml
```

Additionally, options can be set to trace the prerequisite product, MQSeries Classes for Java.

Within a file called 'wmqima88.properties' in the current working directory, add a line indicating the level of trace required (1 to 5, as documented in the manual, "MQSeries Classes for Java").

For example:

```
trace=1
```

This results in a file named 'wmqima88.log' being created that contains the MQSeries classes for Java trace.

Appendix B. Performance

Excessive use of the command `mqsiremotereadlog` may lead to some performance concerns. This remote command can retrieve large amounts of data from the brokers trace log. The length of time that the remote command takes to execute is directly proportional to the size of the trace log being retrieved. Care should be taken to avoid retrieving large trace logs. This can be achieved by reducing the trace logs maximum size with the `mqsiremotechangetrace` option `'-c'` or by resetting the trace log to a zero length, option `'-r'`, before the actions to be traced are initiated.

The remote commands have been implemented in such a way as to make use of the route that the Control Center uses to communicate with the broker. Degradation in performance may be noticed if this route is being used by large numbers of people retrieving trace log data.

Using the command options `-h`, `-p` and `-q` allows the remote commands to access the broker by a different route. This can lead to a reduction in performance bottlenecks. (Please refer to Appendix C. Broker Communication Routing, for details about the communication mechanisms employed.)

Appendix C. Broker Communication Routing

This Appendix explains how the remote commands communicate with a broker. This information will show users how to use the remote commands to reduce performance bottlenecks.

Default Routing

The remote commands have been implemented to enable ease of use for the normal user, that is, a user who has installed and run the Control Center. This user set his routing information in the Control Center preferences dialogue window. The routing information was a hostname, port and Queue Manager name. These settings form the basis for the route that the Control Center will use to deploy to a broker via the Configuration Manager executables MQSeries Queue Manager.

The remote commands locate and use this information to route requests to the brokers Queue Manager via the Configuration Managers Queue Manager.

This allows for minimum user configuration and also ensures that remote command requests are sent through a known path. However, this will increase the load on the Configuration Managers Queue Manager, especially if the user requests large trace logs.

Alternate routing capability

Selecting the remote command override flags -h, -p and -q will allow the remote command requests to be redirected to a different Queue Manager. Use the data from the Control Center preferences dialogue as an example to try out these flags.

The selection of the Queue Manager (-q) could be the brokers Queue Manager or another Queue Manager entirely (similar in configuration to the Configuration Managers Queue Manager, i.e. supplying a route to the broker).

The flag -q must specify a Queue Manager that is either a broker Queue Manager or a transmission queue that can connect to a broker Queue Manager.

When using the -q flag to route the request directly to the broker the argument specified with the -q flag will be the same as the unflagged argument in the remote command (i.e. brokerQueueManagerName).

For example :

```
mqsi remotereporttrace MQSI_SAMPLE_QM -u -e default -q MQSI_SAMPLE_QM -p
listerPortNumber -h machineName
```

Alternate routing prerequisites

If this alternate routing mechanism is used, then ensure that the following Server Connection Channel is created on the machine (-h) where the destination Queue Manager (-q) exists.

```
runmqsc brokerQueueManagerName

define channel('SYSTEM.BKR.CONFIG') CHLTYPE(SVRCONN)
```

Appendix D. Message Catalogues

If a message is displayed indicating that a message catalogue could not be located then this message catalogue should be available on the machine where the broker is running. The message will indicate the name of the message catalogue file (i.e. MQSiv202 would mean the following file names, MQSiv202*.properties). Take a copy of these files from the broker's machine and place them in the products (Control Center installation) "messages" directory on the platform where the remote command is issued.

These types of file (*.properties) have exactly the same format on all platforms, if you use FTP to transfer these files you should use the "ASCII" rather than "BINARY" form of transfer.

Appendix E. MQSeries Connection failures

This set of command utilities makes use of the MQSeries Queue Managers and Queues that are required to connect the Control Center to the Configuration Manager and then to the broker. If this set of MQSeries resources is incorrectly defined or not available for use, then the command utilities may fail or wait indefinitely.

- The most common problem occurs where the broker name is used in the command utility rather than the broker transmission queue name. This is simply corrected by using the transmission queue name in the command.
- The next most common problem occurs when the trace file requested by the `mqsiremotereadlog` command exceeds the maximum message length defined for the MQSeries resources. Again this is simply corrected by increasing the maximum message length, as described in the section entitled 'Trace log size'.
- The third most common error occurs when MQSeries messages that the command utilities have requested are sent to the MQSeries Dead Letter Queue.

This usually occurs when the brokers Queue Manager transmission queue to the Configuration Manager Queue Manager is defined with property 'default persistence' set to 'Persistent'.

Set this property to 'Not Persistent' and restart the brokers Queue Manager. Reissue the command.

This problem will be noticed especially when communicating with a z/OS broker. Resetting the transmission queue property (on z/OS the property is dynamically enabled) will resolve this problem.

Alternately, APAR IY29128 can be applied to the broker (all platforms). Once this corrective service is applied there is no need to alter the transmission queue property.