WebSphere MQ
API Trace

Version 5.2.1

SupportPac MA0W

1 November, 2012

Osamu Inoue
IBM Japan, Ltd.
e-mail : oinoue@jp.ibm.com
IBM internal : Osamu 2 Inoue/Japan/IBM

Nineteenth Edition, November 2012

This edition applies to Version 5.2.1 of WebSphere MQ API Trace and to all subsequent releases and modifications unless otherwise indicated in new editions.

Table of Contents

Notices

Trademarks and services marks

Preface

Who this books is for

This book is for users who want to get trace data of WebSphere MQ API calls for performance measurement, analysis of MQ usage, problem determination and so on.

Your feedback comments to me

Your feedback comments such as proposal for functional enhancement, usability improvement, problem reporting and so on are highly appreciated.

Please refer to “Your feedback comments to me” in page 77 for sending your comments.

Web site where latest package is uploaded

The latest package is stored at the following site:

http://www-1.ibm.com/support/docview.wss?rs=171&uid=swg24010343&loc=en_US&cs=utf-8&lang=en

Summary of Changes

Version	*Level*	*Changes*
5.2.1	1 November, 2012 API Exit 7.52 Channel Exit 7.52 Reformat API trace Utility 3.52 Summary Utility 2.52 Channel Summary Utility 2.52 GUI for parameter file 1.0.1	● Improved internal logic ● Fixed defect in handling TargetQueue parameters in parameter file
5.2.0	1 July, 2012 API Exit 7.50 Channel Exit 7.50 Reformat API trace Utility 3.50 Summary Utility 2.50 Channel Summary Utility 2.50 GUI for parameter file 1.0.1	● Support WebSphere MQ V7.5
5.1.2	1 May, 2012 API Exit 7.45 Channel Exit 7.45 Reformat API trace Utility 3.45 Summary Utility 2.45 Channel Summary Utility 2.45 GUI for parameter file 1.0.1	● API exit interface at MQ client side (MQ 7.1 and subsequent releases)
5.1.1	1 March, 2012 API Exit 7.43 Channel Exit 7.43 Reformat API trace Utility 3.43 Summary Utility 2.43 Channel Summary Utility 2.43 GUI for parameter file 1.0.0	● Refine output format for MQCONN and MQCONNX
5.1.0	1 December, 2011 API Exit 7.41 Channel Exit 7.41 Reformat API trace Utility 3.41 Summary Utility 2.41 Channel Summary Utility 2.41 GUI for parameter file 1.0.0	● Support WebSphere MQ V7.1 ● Changed program names ● Changed name of input/output files ● API exit for 64 bit program on 64 bit Windows with MQ V7.1 ● GUI for parameter file manipulation on Windows ● Trace XA calls ● Obsolete reformat MQ trace utility
5.0.7	15 March, 2010 API Exit 7.22 Channel Exit 7.22 Reformat API trace Utility 3.22 Reformat MQ trace Utility 3.22 Summary Utility 2.07 Channel Summary Utility 2.07	● Fixed defect for showing MQCHARV data

5.0.6	19 September, 2009 API Exit 7.21 Channel Exit 7.21 Reformat API trace Utility 3.21 Reformat MQ trace Utility 3.21 Summary Utility 2.07 Channel Summary Utility 2.07	● Add ShowMaxLenCharv and ShowMaxLenCharvString parameters ● Add –v parameter to reformat utility ● Fix defects in reformat API trace utility
5.0.5	1 September, 2009 API Exit 7.20 Channel Exit 7.20 Reformat API trace Utility 3.20 Reformat MQ trace Utility 3.20 Summary Utility 2.07 Channel Summary Utility 2.07	● Support WebSphere MQ V7.0.1
5.0.4	1 May, 2009 API Exit 7.12 Channel Exit 7.12 Reformat API trace Utility 3.12 Reformat MQ trace Utility 3.12 Summary Utility 2.06 Channel Summary Utility 2.06	● Refined output of PCF data ● Show CCSID description in output ● Show 64 bit integer values in decimal
5.03	1 January, 2009 API Exit 7.10 Channel Exit 7.10 Reformat API trace Utility 3.10 Reformat MQ trace Utility 3.10 Summary Utility 2.04 Channel Summary Utility 2.04	● Format message content for reference message header (MQHREF) ● Format message content for distribution header (MQHDIST) ● Support CCSID 437 and 850 ● Show description of message format
5.02	28 November, 2008 API Exit 7.09 Channel Exit 7.09 Reformat API trace Utility 3.09 Reformat MQ trace Utility 3.09 Summary Utility 2.04 Channel Summary Utility 2.04	● Fix SIGFPE error on Red Hat Enterprise Linux 3 and 4 ● Show length of parsed message content ● Support automatic CCSID selection (ShowHexCharCcsid parameter) in parameter file ● Show topic string for subscription handle ● Add –s option to message browsing utility ● Fix minor defects
5.01	20 October, 2008 API Exit 7.05 Channel Exit 7.05 Reformat API trace Utility 3.05 Reformat MQ trace Utility 3.05 Summary Utility 2.02 Channel Summary Utility 2.02 Message Browsing Utility 1.12	● Add –r option to message browsing utility ● Treat return code 2107 (MQRC_XWAIT_CANCELED) as informational result ● Show multiple RFH2 data in an MQ message in parsed format ● Show CCSID description for all CCSID values ● Fix defects
5.00	1 September, 2008 API Exit 7.03 Channel Exit 7.03	● Support WebSphere MQ V7.0 ● Add ExecFlag=env parameter supporting trace enabling function by specifying MA0W environment value

	Reformat API trace Utility 3.00 Reformat MQ trace Utility 3.00 Summary Utility 2.00 Channel Summary Utility 2.00	
4.32	12 November, 2007 API Exit 6.50 Channel Exit 6.50 Reformat API trace Utility 2.91 Reformat MQ trace Utility 2.91 Summary Utility 1.61 Channel Summary Utility 1.41	● Add ShowMqVersion parameter ● Add PutAlertFileInfo parameter ● Add ShowChannelAttributes parameter ● Support MQCD Version 8 fields ● Message content before code conversion is shown in MQGET when ShowDataConvOnGet=yes parameter is specified. ● Fix defects
4.31	17 September, 2007 API Exit 6.31 Channel Exit 6.31 Reformat API trace Utility 2.71 Reformat MQ trace Utility 2.71 Summary Utility 1.51 Channel Summary Utility 1.31	● Show suppressed sequence number in detail trace ● Improve internal logic ● Fix defects
4.3	1 September, 2007 API Exit 6.30 Channel Exit 6.30 Reformat API trace Utility 2.70 Reformat MQ trace Utility 2.70 Summary Utility 1.50 Channel Summary Utility 1.30	● Show parsed MQRFH name value data ● Show parsed PCF attributes for publish/subscribe ● Assign unique file name for errors in channel exit. ● Show performance counter in 20 digits linear decimal on Windows ● Format all parts of a message content when it contains multiple parts ● Identify encoding (big endian and little endian) regarding trace data for channel exit. ● Suppress error in showing parsed MQMD data for MQGET (before API execution, after API execution with error) ● Improve performance ● Improve internal logic for file I/O ● Improve internal logic against memory corruption ● Fix defects
4.21	1 April, 2007 API Exit 6.10 Channel Exit 6.10 Reformat API trace Utility 2.50 Reformat MQ trace Utility 2.50 Summary Utility 1.40 Channel Summary Utility 1.20	● Support Linux for s390x (zSeries) ● Support inline comment in parameter file ● Add ShowSummaryCorrelId parameter
4.2	1 January, 2007 API Exit 6.01 Channel Exit 6.01 Reformat API trace Utility 2.40 Reformat MQ trace Utility 2.40 Summary Utility 1.30 Channel Summary Utility 1.10	● Add ShowSummaryDataSuppress parameter ● Add suffix “.t” to name of active trace files ● Synchronization of date/time information in trace file and alert file ● Show character image for MsgId, CorrelId and GroupId ● Show as unknown name when invalid object handle is passed ● Show version in parsed data type ● Improved internal logic ● Fix defect when invalid object handle is passed
4.1	1 October, 2006 API Exit 5.80 Channel Exit 5.80 Reformat API trace Utility 2.30	● Show parsed message contents. PCF, trigger, dead letter header, event and MD extension can be formatted ● Reduced memory usage by more than 600KB in each MQ thread (API exit, channel exit) ● Show parsed message content by channel exit ● Show parsed message content by reformat MQ trace utility

	Reformat MQ trace Utility 2.30 Summary Utility 1.23 Channel Summary Utility 1.02	● Reduced module size by optimizing program code ● Change default WriteBufferSize parameter value to 10000 ● Improved logic to invoke trace file compression program ● Added process parameters ● Change parameter name "ShowParsedXmlData" to "ShowParsedMsg Content" ● Change parameter name "MaxParsedXmlLineLength" to "MaxParsed OutputLineLength" ● Added options to reformat API trace utility ● Added options to reformat MQ trace utility ● Obsolete –mq53aix parameter from Reformat MQ trace utility ● Fix defect for ShowPerformanceOnly=yes parameter
4.0	1 August, 2006 API Exit 5.51 Channel Exit 5.51 Reformat API trace Utility 2.20 Reformat MQ trace Utility 2.20 Summary Utility 1.21 Channel Summary Utility 1.00	● Provided channel exit trace ● Provided reformat utility for MQ trace ● Provided summary utility for channel exit trace ● Enhanced reformat API trace utility for channel exit trace ● Show parsed MQCLOSE options ● Show locale information for UNIX ● Added ShowProcessEnvValue parameter ● Fix defect in formatting invalid RFH2 data (PMR 56959,840,758) ● Fix defect when error message text exceeds 200 bytes ● Fix defect in handling decimal point in summary utility. Both period and comma are supported as decimal point.
3.2	10 April, 2006 API Exit 5.42 Reformat API trace Utility 2.11 Summary Utility 1.12	● Show MQINQ/MQSET values descriptively as string ● Show dynamic queue name and distribution list exactly as object name ● Show explanation of reason code in alert file ● Show message when error statements exist in parameter file ● Show null bit values as "(null)" for message id, correlation id, group id, accounting token and some other bit attributes ● Show encoding information in detail ● Show parsed MQXQH data ● Support generic queue name specification for TargetQueue parameter ● Added PutAlertSuppress parameter for suppressing alert for particular messages ● Added CountSelectedCallsOnly parameter for counting API calls ● Refined logic to enable/disable trace dynamically ● Automatic CCSID selection in reformat utility regarding locale ● Fix error with message AMQ6175/AMQ7214 at startup of 32-bit MQ applications on Linux (PowerPC) ● Fix defect not showing parsed XML data in reformat utility ● Change package file format to gzip for UNIX systems to reduce package size
3.1	1 February, 2006 API Exit 5.36 Reformat API trace Utility 2.05 Summary Utility 1.12	● Added PutAlertFileError parameter and PutAlertFileWarning parameter to indicate error occurrence ● Added WriteStream parameter for putting trace data in non I/O stream mode ● Added WriteAfterEachApi parameter for putting trace data after processing for each API call ● Obsolete ForceWrite parameter ● Change default WriteCache parameter value to "yes" ● Change default ShowParsedXmlData parameter value to "yes" ● Treat ReasonCode 2059 (MQRC_Q_MGR_NOT_AVAILABLE) as informational result. (not error result) ● Eliminate trace for endmqm process to suppress unexpected AMQ7216 message to MQ log ● Show split summary when ShowPerformanceIntervalTime parameter value or ShowPerformanceIntervalCount parameter value is greater than 0 ● Show message after API call summary data when unsupported MQ attributes by the utility were found

		● Change name of error message file ● Change name of default output file of summary data generation utility ● Added MQ attribute definitions in formatting output data ● Fixed defects
3.0	1 January, 2006 API Exit 5.21 Reformat API trace Utility 2.01 Summary Utility 1.10	● Support Solaris 10 for x86-64 ● Support HP-UX 11iV2 (11.23) for IA-64 ● Support enabling/disabling trace function dynamically after starting queue manager ● Added summary data generation utility ● Added CheckUpdateRequestInterval parameter in parameter file ● Show MQCONNX Options newly added by MQ V6.0 ● Change default WriteCache parameter value to “no” ● MaxFileSize parameter is effective with ShowSummaryOnly=yes ● Fix defect for ShowDataConvOnGet parameter
2.0	8 November, 2005 API Exit 5.10 Reformat API trace Utility 2.00	● Support Linux for PowerPC ● Support Linux for x86-64 ● Added ShowPerformance… parameters in parameter file ● Added CreateEmptyFile parameter in parameter file ● Added ExecFlag parameter in parameter file ● Added ShowParsedXmlData and MaxParsedXmlLineLength parameter in parameter file ● Change default ShowDataConvOnGet parameter value to “no” ● Show message length in summary data ● Show process time in summary as performance data ● Show API process time in microsecond on Windows ● Show queue name in summary for MQOPEN ● Show parsed MQSCO, MQAIR, LDAPUserName ● Put message id in summary data so that specific message can be traced. ● Update output format of summary data ● Change file name for trace inactive processes ● Change output file name of summary trace ● Change output file name to lowercase character. ● Unified tar.Z files in package for each operating environment ● Support MQIA_xxxx attributes for MQINQ/MQSET API newly added by MQ V6.0 ● Fix defect in showing queue name in summary record when object handle is reused
1.0	10 August, 2005 API Exit 4.30 Reformat API trace Utility 1.20	● Initial release

Chapter 1. Overview

Difficulty in checking detail of WebSphere MQ API call

When WebSphere MQ (specified as “MQ” hereafter) users want to verify MQ API calls performed by application programs, one of basic methods is to look into source code of MQ application program. Source code should tell all of programming logic including MQ API calls. However it is sometimes not realistic to understand all of MQ API calls since application programs might be very complicated. In addition, it might not be realistic to identify run time message data in MQ queues by looking into MQ applications because message data might come from various external components such as database systems, online transaction systems via network and so on.

Another candidate to check MQ API call by application programs is to use MQ tracing facility. But it is not easy for general users to understand trace data at a glance because API parameters are shown just in hexadecimal representation and character representation. It is also not suitable for long time trace regarding huge data volume.

The objective of API trace utility is to provide method to trace all MQ API calls including parameters and process results in easy way.

Exit interface provided by WebSphere MQ

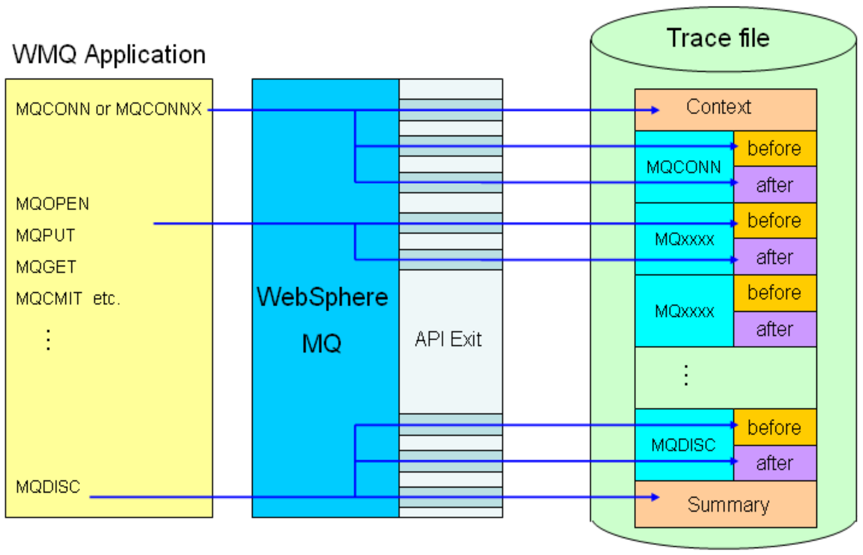
WebSphere MQ V5.3 has newly introduced exit interface to interact with each MQ API call. API exit can be invoked both before API execution and after API execution. In calling API exit programs, all parameters passed by MQ API caller are passed to API exit as they are. All parameters returned by MQ are also passed to API exit after API execution.

In addition to API exit, exit interface in sending and receiving message data via MQ channels has become available.

Functions provided by WebSphere MQ API trace utility

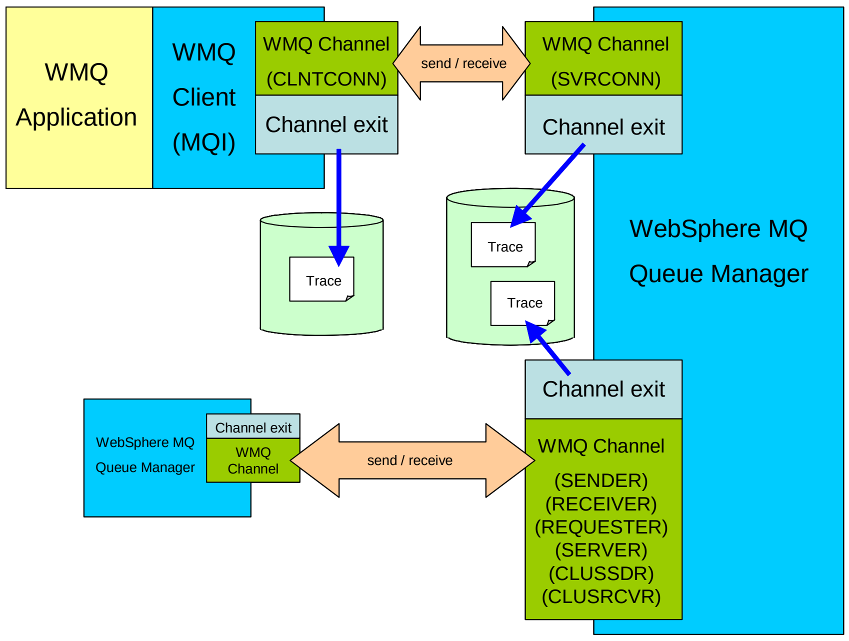
Overview of functions using API exit interface

By understanding exit interface provided by MQ, you will find that all MQ API calls can be captured completely using API exit interface. API trace utility provides trace function of MQ API calls exploiting API exit interface. Users can identify detail of MQ API calls as actual results without looking into program logic of MQ application programs.



Overview of functions using channel exit interface

In addition to API trace described previously, MQ messages transferred via MQ channels can be traced exploiting channel exit interface. Detail information in sending and receiving message via channels can be traced in addition to channel status information provided by MQ administration interface.



Functional highlights in getting trace data

- All parameter values passed by MQ applications can be traced completely. (API exit)
- All parameter values set by MQ can be traced completely including completion code and reason code. (API exit)
- Time before API call, time after API call and process time of each API call are shown for each API call. Time unit is in microsecond on UNIX systems. On Windows, time unit is in millisecond and 64-bit performance counter based on CPU clock is shown, in addition, so that process time can be identified in detail. (API exit)
- All MQ messages transferred via MQ channels can be traced completely. (channel exit)
- Parameter values are parsed and shown in the format of easy view.
- Total call count for each API is shown as summary. Success count, warning count and fail count are also shown respectively.
- Process time of each API call is shown in detail output and summary output. Average time, minimum time and maximum time are also shown at the last of trace data. (API exit)
- Multiple byte character set (MBCS) handling is considered for text output.

Supplemental functions in getting trace data

- Trace data can be put to multiple files when data volume becomes large. In default setting, new trace files are created in every 100MB trace data.
- File compression program can be invoked at the creation time of new trace file to reduce disk space usage. It is useful for long run test putting large data volume.
- Operating environment, process name, and related process/thread information are shown as a part of context information.
- Target queue name and/or process name to be traced can be specified to eliminate unnecessary data for minimizing output data and also for minimizing performance degradation.
- Alert files can be generated when MQ API call error occurs. User can find MQ API call errors more easily by finding alert files.

Utility program for WebSphere MQ trace

- Utility for reformatting and adding parsed information to trace data generated by WebSphere MQ trace function is provided.

The other utility programs

- Utility program for reformatting and adding parsed information to trace data generated by API trace utility is provided.
- Utility program to generate API call summary is provided. API call data in multiple trace files can be summarized into a summary data.
- Utility program to summarize API call count, result count, average/minimum/maximum process time, process statistics for each queue is provided.
- Utility program to browse messages in a queue is provided.

Policies in developing API trace utility

The utility has been developed based on the following policies.

Minimize impact to existing MQ environment

Only system function calls for file I/O for putting trace data and reading parameter file, getting process time information, asynchronous compress program invocation (optional), memory allocation and free at termination are performed in API Trace Utility modules. No other system calls such as shared memory access, semaphore operation, mutex operation, timer wait and new thread creation exist in the utility to minimize impact to existing MQ environment.

Minimize performance degradation

By optimizing program logic, efforts to minimize performance degradation in using the utility have been made. Efforts to minimize memory usage have also been made.

Ease of use

Considerations for simple installation, ease of use and ease in reading/understanding output trace data have been made. Users can understand the utility with basic MQ skills and reference of WebSphere MQ API call interface.

Give me your feedback comments

Your feedback comments such as proposal for functional enhancement, usability improvement, problem reporting and so on are highly appreciated. Please do not hesitate to send it.

Please send your comments to the following e-mail address either using ma0w_your_comment_to_me.txt file in this package or free form.

e-mail :	oinoue@jp.ibm.com
IBM internal :	Osamu 2 Inoue/Japan/IBM

Summary of methods for getting WebSphere MQ API trace

Methods for getting trace of MQ API calls performed by MQ application programs and MQ messages transferred via MQ channels are summarized in the following charts. Choose best method regarding your objectives for getting trace data and processing environment.

For functional test, problem determination

Target	**Method** **Configuration parameters** **MQ commands**	**Advantage**	**Usage notes**	**Reference page**
MQ API calls performed by applications at MQ queue manager side (server side)	API Trace TargetProcess= TargetQueue=	Formatted output for easy view Showing attributes descriptively Suitable for long time trace (process selection, target queue selection, compress function, multiple files)	Queue manager must be restarted after modifying qm.ini file.	22, 25, 25, 27, 28, 58, 59, 92, 100
	Standard MQ Trace strmqtrc -t api (-t all) dspmqtrc	Standard method provided by MQ Getting MQ internal trace	Data volume might become large. Difficulty in reading attribute values at a glance Reformatting required for UNIX	
MQ API calls performed by applications at MQ client side	Channel trace (for CLNTCONN channel)	Formatted output for easy view Showing attributes descriptively Suitable for long time trace (compress function, multiple files)		22, 25, 32, 34, 36, 83, 92, 97
	Channel trace by setting MQCONNX parameters	Formatted output for easy view Showing attributes descriptively Suitable for long time trace (compress function, multiple files)	Client application program must be modified.	22, 25, 83, 85, 92, 97
	Standard MQ Trace strmqtrc -t api (-t all) dspmqtrc	Standard method provided by MQ Getting MQ internal trace	Data volume might become large. Difficulty in reading attribute values at a glance Reformatting required for UNIX	83
	API Trace (MQ V7.1 client and subsequent releases)	Formatted output for easy view Showing attributes descriptively Update and restart at queue manager side are not needed.	Supported on MQ V7.1 client and subsequent releases only. (Version of queue manager can be different.) Connection via SVRCONN / CLNTCONN channel only.	22, 25, 30, 92, 100
Messages transferred via MQ channel	Channel trace	Formatted output for easy view Showing attributes descriptively Suitable for long time trace (compress function, multiple files)		22, 25, 32, 34, 36, 83, 92, 97

	API Trace
	TargetProcess=amqrmppa
	TargetProcess=runmqchl
		Formatted output for easy view
		Showing attributes descriptively
		Suitable for long time trace (compress function, multiple files)
			Queue manager must be restarted after modifying qm.ini file.	22, 25, 25, 27, 28, 58, 92, 100

	Standard MQ Trace strmqtrc -t comms (-t all) dspmqtrc	Standard method provided by MQ Getting MQ internal trace	Data volume might become large. Difficulty in reading attribute values at a glance Reformatting required for UNIX	

For performance measurement, statistics report

Target	Method **Configuration parameters** **MQ commands**	Advantage	Usage notes	Reference page
MQ queue manager	API Trace ShowSummaryOnly=yes Summary utility	Easy operation Easy view Minimum performance degradation Minimum trace data volume Report for each queue	Queue manager must be restarted after modifying qm.ini file.	22, 25, 25, 27, 28, 58, 70, 98, 107
	Standard MQ functions	Standard functions provided by MQ		

MQ queue	API Trace
	ShowSummaryOnly=yes
	TargetQueue=..
	
	Summary utility
		Easy operation
		Easy view
		Minimum performance degradation
		Minimum trace data volume
		Report for each queue
			Queue manager must be restarted after modifying qm.ini file.	22, 25, 25, 27, 28, 58, 59, 70, 98, 107

	Standard MQ functions	Standard functions provided by MQ		

MQ channel	API Trace
	TargetProcess=amqrmppa
	TargetProcess=runmqchl
	ShowSummaryOnly=yes
	
	Channel trace
	ShowSummaryOnly=yes
		Easy operation
		Easy view
		Minimum performance degradation
		Minimum trace data volume
			Queue manager must be restarted after modifying qm.ini file.	22, 25, 25, 27, 28, 32, 34, 36, 58, 58, 70, 83, 98

	Standard MQ functions	Standard functions provided by MQ		

For tracking specific MQ messages

Target	Method	Advantage	Usage notes	Reference page
MQ message	API Trace ShowSummaryOnly=yes Channel trace ShowSummaryOnly=yes	Easy operation Easy view Minimum performance degradation Minimum trace data volume	Queue manager must be restarted after modifying qm.ini file.	22, 25, 25, 27, 28, 32, 34, 36, 58, 70, 83, 97
	Standard MQ functions	Standard functions provided by MQ	Enhanced by MQ V6.0	

Chapter 2. Supported Environments and API Calls

Supported Environments

WebSphere MQ versions

The following MQ versions are supported by the utility.

- WebSphere MQ Version 7.5
- WebSphere MQ Version 7.1
- WebSphere MQ Version 7.0.1
- WebSphere MQ Version 7.0
- WebSphere MQ Version 6.0 (This version is now end of service.)

Operating systems and addressing modes

Operating environments and addressing modes supported by the utility are shown in the following chart..

Programs provided in this SupportPac should work normally on the operating environments listed in the chart as far as they are supported by WebSphere MQ.

Operating System	**CPU**	**WebSphere MQ V7.1 and subsequent**		**WebSphere MQ V6.0, V7.0, V7.0.1**	
		32 bit mode	**64 bit mode**	**32 bit mode**	**64 bit mode**
AIX	PowerPC	OK	OK	OK	OK
Solaris	SPARC	OK	OK	OK	OK
	x86-64	OK	OK	OK	OK
Linux	x86-32	OK	(N/A)	OK	(N/A)
	x86-64	OK	OK	OK	OK
	PowerPC	OK	OK	OK	OK
	s390x	(*1)	(*1)	(*1)	(*1)
HP-UX	PA-RISC	(N/A)	(N/A)	OK	OK
	IA-64 (IPF)	(*2)	(*2)	OK	OK
Windows	x86-32	OK	(N/A)	OK	(N/A)
	x86-64	OK	OK	OK	(*3)

(*1) Not supported due to missing environment though the platform was supported in the previous versions.
(*2) Programs are built on MQ V7.0.1. Restrictions exist with MQ V7.1.
(*3) API exit for 64 bit application programs are not supported.

System resource

Efforts to minimize system impact in using the utility have been made by optimizing program logic.

For throughput regarding CPU usage, memory usage and required disk space, refer to “Chapter 8. Performance”.

Supported API Calls

The following chart shows trace capability of the utility for each WebSphere MQ API call.
It is based on the provision of API exit interface by WebSphere MQ.

API name	**WebSphere MQ V7.0 and subsequent releases**	**WebSphere MQ V6.0**
MQCONN	Traced	Traced
MQCONNX	Traced	Traced
MQDISC	Traced	Traced
MQBEGIN	Traced	Traced
MQCMIT	Traced	Traced
MQBACK	Traced	Traced
MQOPEN	Traced	Traced
MQCLOSE	Traced	Traced
MQPUT	Traced	Traced
MQPUT1	Traced	Traced
MQGET	Traced	Traced
MQINQ	Traced	Traced
MQSET	Traced	Traced
DATA_CONV_ON_GET (Before data conversion in MQGET)	Traced	Traced
MQSUB	Traced	(N/A)
MQSUBRQ	Traced	(N/A)
MQCB	Traced	(N/A)
MQCTL	Traced	(N/A)
MQCB_FU (Callback routine set by MQCB)	Traced	(N/A)
MQSTAT	Traced	(N/A)
MQCRTMH	(*1)	(N/A)
MQDLTMH	(*1)	(N/A)
MQBUFMH	(*1)	(N/A)
MQMHBUF	(*1)	(N/A)
MQSETMP	(*1)	(N/A)
MQDLTMP	(*1)	(N/A)
MQINQMP	(*1)	(N/A)

(*1) Not traced because API exit interface is not provided by MQ.

Chapter 3. Installation

Step 1 and 2 are common steps required for all users.
The other steps are optional.

Step 1. Download modules	Required step for all users.
Step 2. Prepare directory for trace data	Required step for all users.
Step 3. Define API exit (queue manager side)	Optional. Definitions depend on user’s requirements.
Step 4. Define API exit (client side)	Optional. Definitions depend on user’s requirements.
Step 5. Define Channel Exit	Optional. Definitions depend on user’s requirements.

Step 1. Download modules

Content in the SupportPac package

Download package file and extract files in it. The following files are to be extracted:

File name	**Description**
ma0w_readme.txt	Introduction
(files in license directory)	License information files
ma0w_your_comment_to_me.txt	Comment sheet
(files listed in “Programs for Windows” section in page 24.)	Programs for WIndows
ma0w_AIX.tar.gz	Programs for AIX
ma0w_SunOS_sparc.tar.gz	Programs for Solaris (SPARC)
ma0w_SunOS_x86.tar.gz	Programs for Solaris (x86-64)
ma0w_HP-UX_parisc.tar.gz	Programs for HP-UX (PA-RISC)
ma0w_HP-UX_ia64.tar.gz	Programs for HP-UX (IA-64)
ma0w_Linux_ppc.tar.gz	Programs for Linux (PowerPC)
ma0w_Linux_x86.tar.gz	Programs for Linux (x86-32, x86-64)

For UNIX environments, unpack .tar.gz files after transferring to target operating environment in binary mode.
Commands for unpacking are as follows:

```
gunzip     ma0w_xxxxx_yyyyy.tar.gz
tar  -xvf  ma0w_xxxxx_yyyyy.tar
```

Programs for AIX and Linux

Components	Current version		In previous versions of this SupportPac
	For MQ 7.0.1 and former releases	For MQ 7.1 and subsequent releases	
API exit	qx0api32	qx1api32	iselmqapiexit32
	qx0api32_r	qx1api32_r	iselmqapiexit32_r
	qx0api64	qx1api64	iselmqapiexit64
	qx0api64_r	qx1api64_r	iselmqapiexit64_r
Channel exit	qx0chl32	qx1chl32	iselmqchannelexit32
	qx0chl32_r	qx1chl32_r	iselmqchannelexit32_r
	qx0chl64	qx1chl64	iselmqchannelexit64
	qx0chl64_r	qx1chl64_r	iselmqchannelexit64_r
Utilities	qxreform32		iselmqreformat32
	qxreform64		iselmqreformat64
	qxsumapi		iselmqsummary32
			iselmqsummary64
	qxsumchl		iselmqchlsummary32
			iselmqchlsummary64
Supplemental	qxset		set_apitrace
	qx.def		iselmqapiexit.def
	qx_setlink		(none)

Programs for Solaris

Components	Current version		In previous versions of this SupportPac
	For MQ 7.0.1 and former releases	For MQ 7.1 and subsequent releases	
API exit	qx0api32	qx1api32	iselmqapiexit32
	qx0api64	qx1api64	iselmqapiexit64
Channel exit	qx0chl32	qx1chl32	iselmqchannelexit32
	qx0chl64	qx1chl64	iselmqchannelexit64
Utilities	qxreform32		iselmqreformat32
	qxreform64		iselmqreformat64
	qxsumapi		iselmqsummary32
			iselmqsummary64
	qxsumchl		iselmqchlsummary32
			iselmqchlsummary64
Supplemental	qxset		set_apitrace
	qx.def		iselmqapiexit.def
	qx_setlink		(none)

Programs for HP-UX

Programs are built on MQ V7.0.1.
With MQ 7.1 and subsequent releases, programs should work only under primary installation environment.

Components	Current version		In previous versions of this SupportPac
	For MQ 7.0.1 and former releases	For MQ 7.1 and subsequent releases (IA-64 only) (primary installation only)	
API exit	qx0api32		iselmqapiexit32
	qx0api32_r		iselmqapiexit32_r
	qx0api64		iselmqapiexit64
	qx0api64_r		iselmqapiexit64_r
Channel exit	qx0chl32		iselmqchannelexit32
	qx0chl32_r		iselmqchannelexit32_r
	qx0chl64		iselmqchannelexit64
	qx0chl64_r		iselmqchannelexit64_r
Utilities	qxreform32		iselmqreformat32
	qxreform64		iselmqreformat64
	qxsumapi		iselmqsummary32
			iselmqsummary64
	qxsumchl		iselmqchlsummary32
			iselmqchlsummary64
Supplemental	qxset		set_apitrace
	qx.def		iselmqapiexit.def
	qx_setlink		(none)

Programs for Windows

Components	Current version		In previous versions of this SupportPac
	For MQ 7.0.1 and former releases	For MQ 7.1 and subsequent releases	
API exit	qx0api32.dll	qx1api32.dll	iselmqapiexit32.dll
	qx0api64.dll	qx1api64.dll	iselmqapiexit64.dll
Channel exit	qx0chl32.dll	qx1chl32.dll	iselmqchannelexit32dll
	qx0chl64.dll	qx1chl64.dll	(none)
Utilities	qxreform32.exe		iselmqreformat32.exe
	qxreform64.exe		(none)
	qxsumapi.exe		iselmqsummary32.exe
	qxsumchl.exe		iselmqchlsummary32
Supplemental	qxset.bat		set_apitrace.bat
	qx.def		iselmqapiexit.def
	qxgui.exe		(none)

Step 2. Prepare directory for trace data

Prepare a directory where trace data is to be put. It must be met the following conditions.

- Directory must be write-accessible by group mqm users.
- Total length of directory name cannot exceed 32 bytes.

Step 3. Define API exit (queue manager side)

Define API exit modules (AIX, Solaris, HP-UX and Linux)

(Step 3-1) Change access permission and module ownership

Exit modules are loaded as part of MQ modules. They must be executable by all local MQ users.
The following commands are example for granting file access.

```
chmod 755 <download_directory>
chmod 755 <parent_directories_of_download_directory>
chmod 755 <download_directory>/qx*
chown mqm:mqm <download_directory>/qx*
```

(Step 3-2) Define symbolic link to exit modules

Symbolic links to exit modules must be defined.
Move current directory to the directory where downloaded programs are unpacked and then run qx_setlink.

```
cd <download_directory>
./qx_setlink
```

After execution of the script file, the following links are to be created and shown, for example.

AIX:

```
ls -l /var/mqm/exits
lrwxrwxrwx    1 ma0w    mqm    27 Dec 08 17:32 qx0api -> /home/ma0w/package/qx0api32
lrwxrwxrwx    1 ma0w    mqm    29 Dec 08 17:32 qx0api_r -> /home/ma0w/package/qx0api32_r
lrwxrwxrwx    1 ma0w    mqm    27 Dec 08 17:32 qx1api -> /home/ma0w/package/qx1api32
lrwxrwxrwx    1 ma0w    mqm    29 Dec 08 17:32 qx1api_r -> /home/ma0w/package/qx1api32_r

ls -l /var/mqm/exits64
lrwxrwxrwx    1 ma0w    mqm    27 Dec 08 17:32 qx0api -> /home/ma0w/package/qx0api64
lrwxrwxrwx    1 ma0w    mqm    29 Dec 08 17:32 qx0api_r -> /home/ma0w/package/qx0api64_r
lrwxrwxrwx    1 ma0w    mqm    27 Dec 08 17:32 qx1api -> /home/ma0w/package/qx1api64
lrwxrwxrwx    1 ma0w    mqm    29 Dec 08 17:32 qx1api_r -> /home/ma0w/package/qx1api64_r
```

Solaris:

```
ls -l /var/mqm/exits
lrwxrwxrwx   1 ma0w    mqm     34 Dec  8 17:41 qx0api -> /export/home/ma0w/package/qx0api32
lrwxrwxrwx   1 ma0w    mqm     34 Dec  8 17:41 qx1api -> /export/home/ma0w/package/qx1api32

ls -l /var/mqm/exits64
lrwxrwxrwx   1 ma0w    mqm     34 Dec  8 17:41 qx0api -> /export/home/ma0w/package/qx0api64
lrwxrwxrwx   1 ma0w    mqm     34 Dec  8 17:41 qx1api -> /export/home/ma0w/package/qx1api64
```

HP-UX:

```
ls -l /var/mqm/exits
lrwxrwxrwx   1 ma0w    mqm     27 Dec 08 17:30 qx0api -> /home/ma0w/package/qx0api32
lrwxrwxrwx   1 ma0w    mqm     29 Dec 08 17:30 qx0api_r -> /home/ma0w/package/qx0api32_r

ls -l /var/mqm/exits64
lrwxrwxrwx   1 ma0w    mqm     27 Dec 08 17:30 qx0api -> /home/ma0w/package/qx0api64
lrwxrwxrwx   1 ma0w    mqm     29 Dec 08 17:30 qx0api_r -> /home/ma0w/package/qx0api64_r
```

Linux:

```
ls -l /var/mqm/exits
lrwxrwxrwx 1 ma0w mqm  27 2011-12-08 17:45 qx0api -> /home/ma0w/package/qx0api32
lrwxrwxrwx 1 ma0w mqm  29 2011-12-08 17:45 qx0api_r -> /home/ma0w/package/qx0api32_r
lrwxrwxrwx 1 ma0w mqm  27 2011-12-08 17:45 qx1api -> /home/ma0w/package/qx1api32
lrwxrwxrwx 1 ma0w mqm  29 2011-12-08 17:45 qx1api_r -> /home/ma0w/package/qx1api32_r

ls -l /var/mqm/exits64
lrwxrwxrwx 1 ma0w mqm  27 2011-12-08 17:45 qx0api -> /home/ma0w/package/qx0api64
lrwxrwxrwx 1 ma0w mqm  29 2011-12-08 17:45 qx0api_r -> /home/ma0w/package/qx0api64_r
lrwxrwxrwx 1 ma0w mqm  27 2011-12-08 17:45 qx1api -> /home/ma0w/package/qx1api64
lrwxrwxrwx 1 ma0w mqm  29 2011-12-08 17:45 qx1api_r -> /home/ma0w/package/qx1api64_r
```

(Step 3-3) Define API exit in qm.ini file.

Define API exit using parameters of ApiExitLocal stanza in /var/mqm/qmgrs/<*qmgr_name*>/qm.ini file.

Parameters:

Parameters	**Value**	**Remark**
Name	Any name	Must be unique. Use only valid characters for MQ objects. Maximum length is 48 bytes.
Sequence	Call priority	Consideration is needed only if other exit definitions coexist.
Function	Entry point name in the module	Value must be "qxEntry". It cannot be changed.
Module	API exit module	Specify module name "qx1api" for MQ V 7.1 and subsequent releases. Specify module name "qx0api" for MQ V 7.0.1 and former releases.
Data	Directory where trace data is to be put.	Specify directory prepared in the previous step. It must be write-accessible by group mqm users. Value cannot exceed 32 bytes.

Example:

```
ExitPath:
  ExitDefaultPath=/var/mqm/exits/
     .........
     .........
ApiExitLocal:
  Name=ma0w
  Sequence=100
  Function=qxEntry
  Module=qx1api
  Data=<data_directory>
```

Define API exit modules (32 bit Windows)

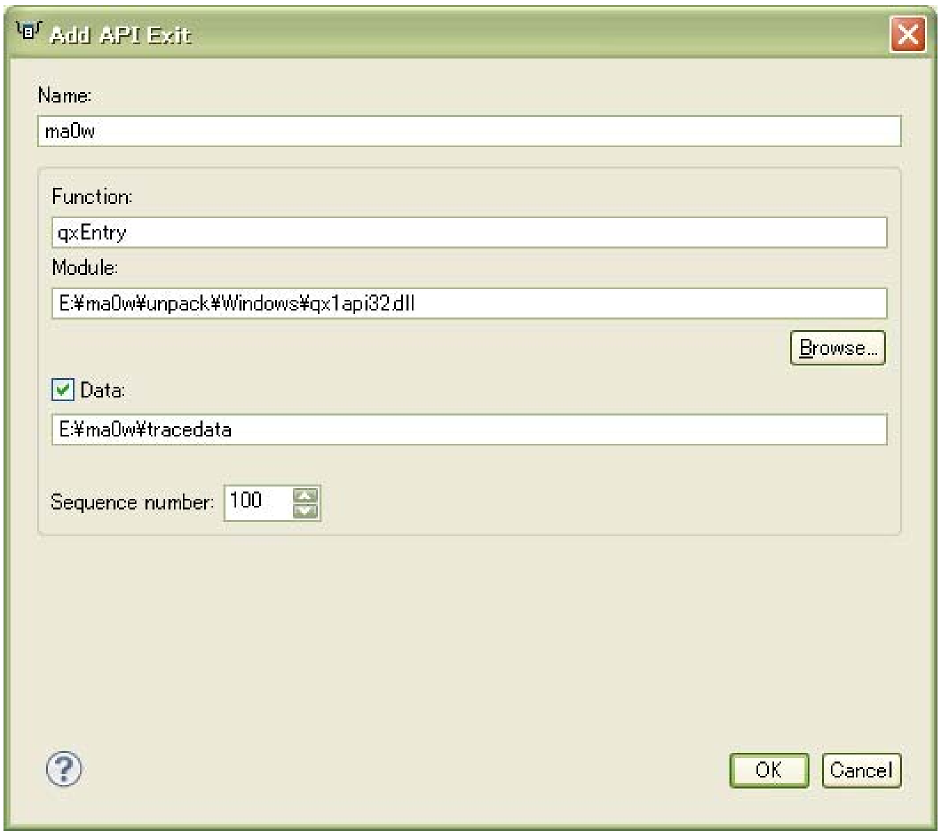
Open MQ Explorer window and define API exit with parameters.

(Step 3-1) Define API exit

Parameters:

Parameters	**Value**	**Remark**
Name	Any name	Must be unique. Use only valid characters for MQ objects. Maximum length is 48 bytes.
Function	Entry point name in the module	Value must be “qxEntry”. It cannot be changed.
Module	API exit module	Specify module name “qx1api32.dll” for MQ V 7.1 and subsequent releases. Specify module name “qx0api32.dll” for MQ V 7.0.1 and former releases.
Data	Directory where trace data is to be put.	Specify directory prepared in the previous step. It must be write-accessible by MQ users. Value cannot exceed 32 bytes.
Sequence number	Call priority	Consideration is needed only if other exit definitions coexist.

Example:



Name:
ma0w

Function:
qxEntry

Module:
E:¥ma0w¥unpack¥Windows¥qx1api32.dll

☑ Data:
E:¥ma0w¥tracedata

Sequence number: 100

Note: “¥” characters shown in this section are shown as backslash in English environment and most of the other language versions of Windows. (Editing environment for this document is in Japanese Windows environment.)

Define API exit modules (64 bit Windows)

API exit for 64 bit programs on Windows is supported by WebSphere MQ V7.1 and subsequent releases.

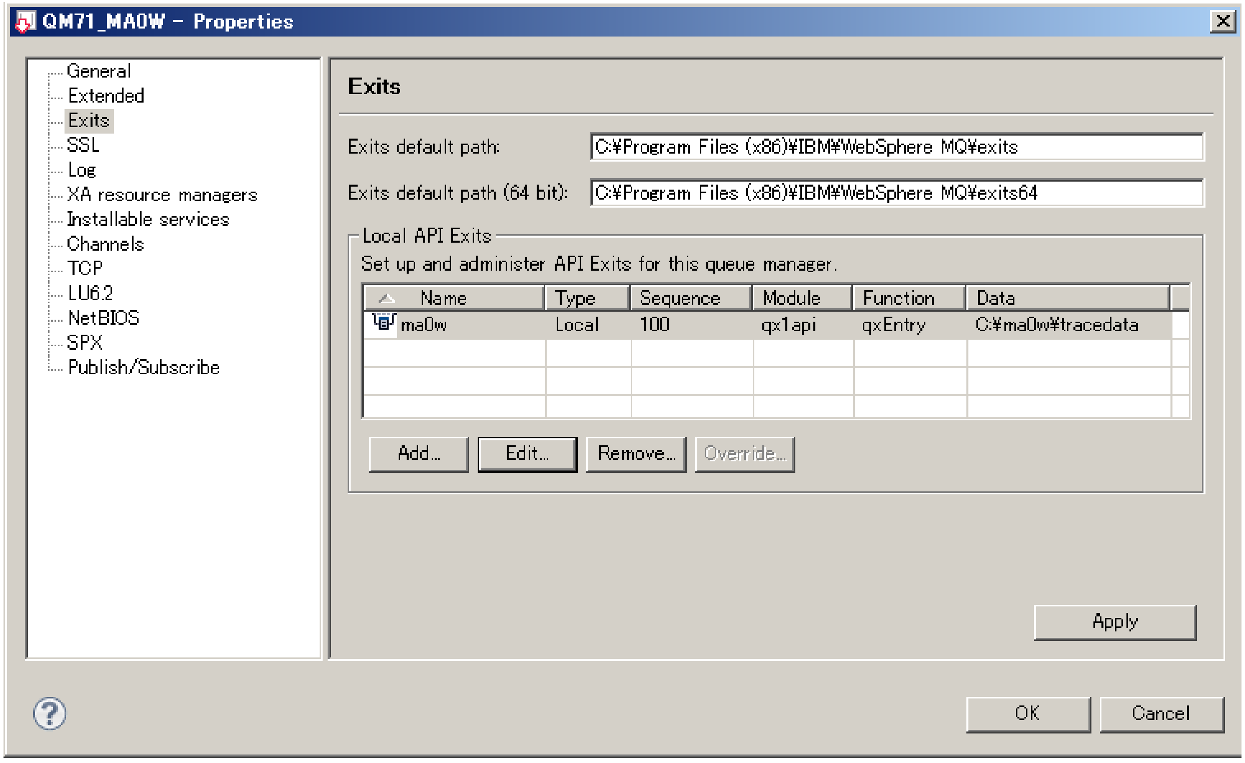
Note: WebSphere MQ itself runs in 32 bit mode. If all MQ application programs runs in 32 bit mode (64 bit MQ application program does not exist) and version of WebSphere MQ is 7.0.1 or less, follow the instruction specified in the section "Define API exit modules (32 bit Windows)" in page 27".

(Step 3-1) Copy and rename exit modules to default exit paths

Both 32 bit exit program qx1api32.dll and 64 bit exit program qx1api64.dll must be copied to exits default paths, respectively. Check exit default paths and copy the files.

In the following example Window, Exits default path definitions are:

```
C:¥Program Files (x86)¥IBM¥WebSphere MQ¥exits
C:¥Program Files (x86)¥IBM¥WebSphere MQ¥exits64
```



Exits

Exits default path: C:¥Program Files (x86)¥IBM¥WebSphere MQ¥exits

Exits default path (64 bit): C:¥Program Files (x86)¥IBM¥WebSphere MQ¥exits64

Local API Exits

Set up and administer API Exits for this queue manager.

Name	Type	Sequence	Module	Function	Data
ma0w	Local	100	qx1api	qxEntry	C:¥ma0w¥tracedata

File name of copied files must be identical to the name specified in "Module" field in the previous step. Therefore, copied files should be renamed to "qx1api.dll", without bit mode "32" and "64".

Example of copy command to be executed:

```
copy qx1api32.dll "C:¥Program Files (x86)¥IBM¥WebSphere MQ¥NonPrimary01¥exits¥qx1api.dll"
copy qx1api64.dll "C:¥Program Files (x86)¥IBM¥WebSphere MQ¥NonPrimary01¥exits64¥qx1api.dll"
```

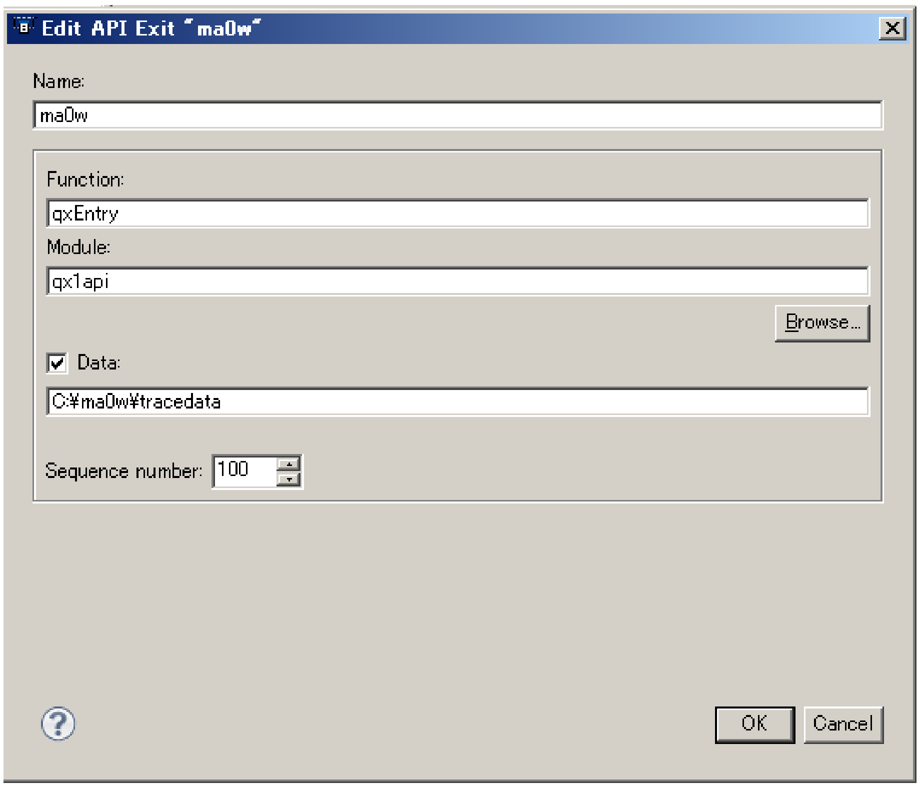
Note: "¥" characters shown in this section are shown as backslash in English environment and most of the other language versions of Windows. (Editing environment for this document is in Japanese Windows environment.)

(Step 3-2) Define API exit

Parameters:

Parameters	Value	Remark
Name	Any name	Must be unique. Use only valid characters for MQ objects. Maximum length is 48 bytes.
Function	Entry point name in the module	Value must be "qxEntry". It cannot be changed.
Module	API exit module	Specify module name "qx1api" for MQ V 7.1 and subsequent releases. Specify module name "qx0api" for MQ V 7.0.1 and former releases. Do not specify directory so that both 32 bit exit program and 64 bit exit program can be called as expected. Bit mode (32 or 64) and file extenstion (.DLL) is also unnecessary.
Data	Directory where trace data is to be put.	Specify directory prepared in the previous step. It must be write-accessible by MQ users. Value cannot exceed 32 bytes.
Sequence number	Call priority	Consideration is needed only if other exit definitions coexist.

Example:



Name: ma0w
Function: qxEntry
Module: qx1api
[✓] Data: C:¥ma0w¥tracedata
Sequence number: 100

Note: "¥" characters shown in this section are shown as backslash in English environment and most of the other Windows environments. (Editing environment for this document is in Japanese Windows environment.)

Step 4. Define API exit (client side)

Note : Description in this section is applicable to MQ client package and it is also applicable to client part in queue manager package.

Supported environments

API exit interface at client side is newly provided by WebSphere MQ V7.1. This interface can be used when MQ client program is connected to queue manager via SVRCONN/CLNTCONN channels.

Versions of queue managers to connect can vary. (V7.0, V6.0 and others)

The following chart shows summary of supported environments.
As shown in it, exit interface is not available if MQ client program is locally connected to queue manger (without using channel).

Operating system	**WebSphere MQ V7.1 client and subsequent releases**		**WebSphere V7.0 client and former releases**
	Connection via SVRCONN / CLNTCONN channel	**Local connection without channel**	
AIX	OK	Not available	Not available
Solaris	OK		
Linux	OK		
HP-UX	Not supported		
Windows	OK		

Define API exit modules (AIX, Solaris and Linux)

(Step 4-1) Change access permission and module ownership (for AIX, Solaris and Linux)

Exit modules are loaded as part of client programs.
The following commands are to be executed, for example.

```
chmod 755 <download_directory>
chmod 755 <parent_directories_of_download_directory>
chmod 755 <download_directory>/qx*
chown mqm:mqm <download_directory>/qx*
```

(Step 4-2) Define API exit in mqclient.ini file.

Define API exit as a set of parameters of ApiExitLocal stanza in mqclient.ini file.

mqclient.ini file can reside in some directories. (Refer to WebSphere MQ documentation for more information.)

- Directory where application executes
- Location defined using MQCLNTCF environment variable

Parameters:

Parameters	Value	Remark
Name	Any name	Must be unique. Use only valid characters for MQ objects. Maximum length is 48 bytes.
Sequence	Call priority	Consideration is needed only if other exit definitions coexist.
Function	Entry point name in the module	Value must be "qxEntry". It cannot be changed.
Module	API exit module	Specify one of the following module names regarding addressing mode (32-bit or 64-bit) and threading condition of application program. Directory can be specified in addition to module name. For AIX and Linux `qx1api32` (32-bit application, single thread) `qx1api32_r` (32-bit application, multi thread) `qx1api64` (64-bit application, single thread) `qx1api64_r` (64-bit application, multi thread) For Solaris `qx1api32` (32-bit application) `qx1api64` (64-bit application) For Windows `qx1api32.dll` (32-bit application) `qx1api64.dll` (64-bit application)
Data	Directory where trace data is to be put.	Specify directory prepared in the previous step. It must be write-accessible by group mqm users. Value cannot exceed 32 bytes.

Example (on AIX, Solaris and Linux):

```
ApiExitLocal:
  Name=ma0w
  Sequence=100
  Function=qxEntry
  Module=/home/ma0w/qx1api32
# Module=/home/ma0w/qx1api32_r      (comment line)
# Module=/home/ma0w/qx1api64        (comment line)
# Module=/home/ma0w/qx1api64_r      (comment line)
  Data=/home/mq71/mqclient
```

Example (on Windows):

```
ApiExitLocal:
  Name=ma0w
  Sequence=100
  Function=qxEntry
  Module=D:¥mqclient¥qx1api32.dll
# Module=D:¥mqclient¥qx1api64.dll     (comment line)
  Data=D:¥mqclient
```

Note: "¥" characters shown in this section are shown as backslash in English environment and most of the other Windows environments. (Editing environment for this document is in Japanese Windows environment.)

Step 5. Define Channel Exit

If MQ messages transferred via channels are to be traced, define channel exit as specified in this section.

Define channel exit modules (AIX, HP-UX, Linux)

Note: For HP-HX, channel exit modules for MQ V7.1 and subsequent releases are not provided.

Change module ownership and access permission

Exit modules are loaded as part of MQ modules. They must be executable by all local MQ users.
The following commands are to be executed to grant file access.

```
chmod 755 <download_directory>
chmod 755 <parent_directories_of_download_directory>
chmod 755 <download_directory>/qx0chl*
chmod 755 <download_directory>/qx1chl*
chown mqm:mqm <download_directory>/qx0chl*
chown mqm:mqm <download_directory>/qx1chl*
```

Definition (Channel type: SDR, RCVR, CLUSSDR, CLUSRCVR, SVR, RQSTR)

Execute the following MQSC command.

```
ALTER CHANNEL(chl_name) CHLTYPE(chl_type)
      SENDEXIT('<download_directory>/chl_exit_program(qxEntry)') SENDDATA('<trace_directory>')
       RCVEXIT('<download_directory>/chl_exit_program(qxEntry)')  RCVDATA('<trace_directory>')
       MSGEXIT('<download_directory>/chl_exit_program(qxEntry)')  MSGDATA('<trace_directory>')
```

Choose ***chl_exit_program*** regarding MQ version and operating environment.

	MQ V7.0.1 and former	*MQ V7.1 and subsequent*
AIX, HP-UX, Linux (64 bit mode)	qx0chl64_r	qx1chl64_r
Linux (32 bit mode)	qx0chl32_r	qx1chl32_r

Definition (Channel type: SVRCONN)

Execute the following MQSC command.

```
ALTER CHANNEL(SVRCONN) CHLTYPE(SVRCONN)
      SENDEXIT('<download_directory>/chl_exit_program(qxEntry)') SENDDATA('<trace_directory>')
       RCVEXIT('<download_directory>/chl_exit_program(qxEntry)')  RCVDATA('<trace_directory>')
```

Choose ***chl_exit_program*** regarding MQ version and operating environment.

	MQ V7.0.1 and former	*MQ V7.1 and subsequent*
AIX, HP-UX, Linux (64 bit mode)	qx0chl64_r	qx1chl64_r
Linux (32 bit mode)	qx0chl32_r	qx1chl32_r

Definition (Channel type: CLNTCONN)

Execute the following MQSC command.
Note that exit program to be chosen depends on bit mode and threading condition of MQ client program.

```
ALTER CHANNEL(chl_name) CHLTYPE(CLNTCONN)
      SENDEXIT('<download_directory>/chl_exit_program(qxEntry)') SENDDATA('<trace_directory>')
       RCVEXIT('<download_directory>/chl_exit_program(qxEntry)')  RCVDATA('<trace_directory>')
```

Choose *chl_exit_program* regarding MQ version, bit mode and threading attribute of MQ client program.

	MQ V7.0.1 and former	*MQ V7.1 and subsequent*
32-bit application, single thraed	qx0chl32	qx1chl32
32-bit application, multi thread	qx0chl32_r	qx1chl32_r
64-bit application, single thraed	qx0chl64	qx1chl64
64-bit application, multi thread	qx0chl64_r	qx1chl64_r

For identifying addressing mode and threading condition of client program, refer to “How to identify addressing mode and threading condition of MQ applications” in page 95.

Notes for channel definition

1 For SENDEXIT, SENDDATA, RCVEXIT, RCVDATA, MSGEXIT and MSGDATA parameters, value must be enclosed by single quote so that lowercase characters are defined as they are.

2 For SENDDATA, RCVDATA and MSGDATA parameters, specify directory where trace data is to be put. (Directory prepared in the previous step)
It must be write-accessible by group mqm users. Value cannot exceed 32 bytes.

Example of channel definition

```
dis chl(CH1) all

AMQ8414: Display Channel details.
   CHANNEL(CH1)                            CHLTYPE(SVRCONN)
   ALTDATE(2011-12-07)                     ALTTIME(18.41.52)
   COMPHDR(NONE)                           COMPMSG(NONE)
   DESCR( )                                DISCINT(0)
   HBINT(300)                              KAINT(AUTO)
   MAXINST(999999999)                      MAXINSTC(999999999)
   MAXMSGL(100000000)                      MCAUSER( )
   MONCHL(QMGR)                            RCVDATA(/home/mq71/tracedata)
   RCVEXIT(/home/ma0w/package/qx1chl64_r(qxEntry))
   SCYDATA( )                              SCYEXIT( )
   SENDDATA(/home/mq71/tracedata)
   SENDEXIT(/home/ma0w/package/qx1chl64_r(qxEntry))
   SHARECNV(10)                            SSLCAUTH(REQUIRED)
   SSLCIPH( )                              SSLPEER( )
   TRPTYPE(TCP)
```

Define channel exit modules (Solaris)

Change module ownership and access permission

Exit modules are loaded as part of MQ modules. They must be executable by all local MQ users.
The following commands are to be executed to grant file access.

```
chmod 755 <download_directory>
chmod 755 <parent_directories_of_download_directory>
chmod 755 <download_directory>/qx0chl*
chmod 755 <download_directory>/qx1chl*
chown mqm:mqm <download_directory>/qx0chl*
chown mqm:mqm <download_directory>/qx1chl*
```

Definition (Channel type: SDR, RCVR, CLUSSDR, CLUSRCVR, SVR, RQSTR)

Execute the following MQSC command.

```
ALTER CHANNEL(chl_name) CHLTYPE(chl_type)
      SENDEXIT('<download_directory>/chl_exit_program(qxEntry)') SENDDATA('<trace_directory>')
       RCVEXIT('<download_directory>/chl_exit_program(qxEntry)')  RCVDATA('<trace_directory>')
       MSGEXIT('<download_directory>/chl_exit_program(qxEntry)')  MSGDATA('<trace_directory>')
```

Choose ***chl_exit_program*** regarding MQ version.

MQ V7.0.1 and former	*MQ V7.1 and subsequent*
qx0chl64	qx1chl64

Definition (Channel type: SVRCONN)

Execute the following MQSC command.

```
ALTER CHANNEL(chl_name) CHLTYPE(SVRCONN)
      SENDEXIT('<download_directory>/chl_exit_program(qxEntry)') SENDDATA('<trace_directory>')
      RCVEXIT('<download_directory>/chl_exit_program(qxEntry)')  RCVDATA('<trace_directory>')
```

Choose ***chl_exit_program*** regarding MQ version.

MQ V7.0.1 and former	*MQ V7.1 and subsequent*
qx0chl64	qx1chl64

Definition (Channel type: CLNTCONN)

Execute the following MQSC command.
Note that exit program to be chosen depends on bit mode of MQ client program.

```
ALTER CHANNEL(chl_name) CHLTYPE(CLNTCONN)
      SENDEXIT('<download_directory>/chl_exit_program(qxEntry)') SENDDATA('<trace_directory>')
       RCVEXIT('<download_directory>/chl_exit_program(qxEntry)')  RCVDATA('<trace_directory>')
```

Choose *chl_exit_program* regarding MQ version and bit mode of MQ client program.

	MQ V7.0.1 and former	*MQ V7.1 and subsequent*
32-bit application	qx0chl32	qx1chl32
64-bit application	qx0chl64	qx1chl64

For identifying addressing mode of client program, refer to “How to identify addressing mode and threading condition of MQ applications” in page 95.

Notes for channel definition

1 For SENDEXIT, SENDDATA, RCVEXIT, RCVDATA, MSGEXIT and MSGDATA parameters, value must be enclosed by single quote so that lowercase characters are defined as they are.

2 For SENDDATA, RCVDATA and MSGDATA parameters, specify directory where trace data is to be put. (Directory prepared in the previous step) It must be write-accessible by group mqm users. Value cannot exceed 32 bytes.

Example of channel definition

```
dis chl(CH1) all

AMQ8414: Display Channel details.
   CHANNEL(CH1)                            CHLTYPE(SVRCONN)
   ALTDATE(2007-03-16)                     ALTTIME(15.43.41)
   COMPHDR(NONE)                           COMPMSG(NONE)
   DESCR( )                                HBINT(300)
   KAINT(AUTO)                             MAXMSGL(100000000)
   MCAUSER( )                              MONCHL(QMGR)
   RCVDATA(/home/ma0w/tracedata)
   RCVEXIT(/home/ma0w/package/qx1chl64(qxEntry))
   SCYDATA( )                              SCYEXIT( )
   SENDDATA(/home/ma0w/tracedata)
   SENDEXIT(/home/ma0w/package/qx1chl64(qxEntry))
   SSLCAUTH(REQUIRED)                      SSLCIPH( )
   SSLPEER( )                              TRPTYPE(TCP)
```

Define channel exit modules (Windows)

Note: “¥” characters shown in this section are shown as backslash in English environment and most of the other Windows environments. (Editing environment for this document is in Japanese Windows environment.)

Definition (Channel type: SDR, RCVR, CLUSSDR, CLUSRCVR, SVR, RQSTR)

Execute the following MQSC command.

```
ALTER CHANNEL(chl_name) CHLTYPE(chl_type)
      SENDEXIT(‘<download_directory>¥chl_exit_program(qxEntry)‘) SENDDATA(‘<trace_directory>’)
       RCVEXIT(‘<download_directory>¥chl_exit_program(qxEntry)‘) SENDDATA(‘<trace_directory>’)
       MSGEXIT(‘<download_directory>¥chl_exit_program(qxEntry)‘) SENDDATA(‘<trace_directory>’)
```

Choose *chl_exit_program* regarding MQ version.

MQ V7.0.1 and former	*MQ V7.1 and subsequent*
qx0chl32	qx1chl32

Definition (Channel type: SVRCONN)

Execute the following MQSC command.

```
ALTER CHANNEL(chl_name) CHLTYPE(SVRCONN)
      SENDEXIT(‘<download_directory>¥chl_exit_program(qxEntry)‘) SENDDATA(‘<trace_directory>’)
       RCVEXIT(‘<download_directory>¥chl_exit_program(qxEntry)‘) SENDDATA(‘<trace_directory>’)
```

Choose *chl_exit_program* regarding MQ version.

MQ V7.0.1 and former	*MQ V7.1 and subsequent*
qx0chl32	qx1chl32

Definition (Channel type: CLNTCONN)

Execute the following MQSC command.
Note that exit program to be chosen depends on bit mode of MQ client program.

```
ALTER CHANNEL(chl_name) CHLTYPE(CLNTCONN)
      SENDEXIT(‘<download_directory>¥chl_exit_program(qxEntry)‘) SENDDATA(‘<trace_directory>’)
       RCVEXIT(‘<download_directory>¥chl_exit_program(qxEntry)‘)  RCVDATA(‘<trace_directory>’)
```

Choose *chl_exit_program* regarding MQ version and bit mode of MQ client program.

	MQ V7.0.1 and former	*MQ V7.1 and subsequent*
32-bit application	qx0chl32	qx1chl32
64-bit application	qx0chl64	qx1chl64

Notes for channel definition

1 For SENDEXIT, SENDDATA, RCVEXIT, RCVDATA, MSGEXIT and MSGDATA parameters, value must be enclosed by single quote so that lowercase characters are defined as they are.

2 For SENDDATA, RCVDATA and MSGDATA parameters, specify directory where trace data is to be put. (Directory prepared in the previous step) It must be write-accessible by group mqm users. Value cannot exceed 32 bytes.

3 Do not specify file extension ".DLL" to SENDEXIT, RCVEXIT and MSGEXIT parameters.

Examples of channel definitions

```
display channel(TO.QM2) all

AMQ8414: Display Channel details.
   CHANNEL(TO.QM2)                          CHLTYPE(SDR)
   ALTDATE(2007-05-30)                      ALTTIME(14.38.54)
   BATCHHB(0)                               BATCHINT(0)
   BATCHSZ(50)                              COMPHDR(NONE)
   COMPMSG(NONE)                            CONNAME(localhost(1415))
   CONVERT(NO)                              DESCR( )
   DISCINT(6000)                            HBINT(300)
   KAINT(AUTO)                              LOCLADDR( )
   LONGRTY(999999999)                       LONGTMR(1200)
   MAXMSGL(4194304)                         MCANAME( )
   MCATYPE(PROCESS)                         MCAUSER( )
   MODENAME( )                              MONCHL(QMGR)
   MSGDATA(D:¥MQTest¥apiexit¥tracedata)
   MSGEXIT(D:¥MQTest¥apiexit¥qx1chl32(qxEntry))
   NPMSPEED(FAST)                           PASSWORD( )
   RCVDATA(D:¥MQTest¥apiexit¥tracedata)
   RCVEXIT(D:¥MQTest¥apiexit¥qx1chl32(qxEntry))
   SCYDATA( )                               SCYEXIT( )
   SENDDATA(D:¥MQTest¥apiexit¥tracedata)
   SENDEXIT(D:¥MQTest¥apiexit¥qx1chl32(qxEntry))
   SEQWRAP(999999999)                       SHORTRTY(10)
   SHORTTMR(60)                             SSLCIPH( )
   SSLPEER( )                               STATCHL(QMGR)
   TPNAME( )                                TRPTYPE(TCP)
   USERID( )                                XMITQ(QM2)
```

Chapter 4. Migration

Program names and file names in the package

Mapping of program names are specified as matrix in “Chapter 3. Installation”.

Migration procedure

Follow the procedure specified in “Chapter 3. Installation” as a new installation.
No migration utility is provided.

Exit defitions

API Exit

Definitions must be updated as specified in “Step 3. Define API exit” of “Chapter 3. Installation”.
Function name (entry point) has been changed from “IselTraceExit” to “qxEntry”.
Program names (module names) have also been changed.

Channel Exit

Definitions must be updated as specified in “Step 5. Define Channel Exit” of “Chapter 3. Installation”.
Function name (entry point) has been changed from “IselTraceExit” to “qxEntry”.
Program names (module names) have also been changed.

Paramter file

Parameter files (iselmqapiexit.def file) used in the previous versions of this SupportPac can be continuously used by renaming file name to “qx.def”.

If you want to rename parameter files with confirmation of values on Windows, newly supported GUI is useful. Refer to “GUI for parameter file manipulation” in page 63 for more information.

Name of output files

Prefix of output files has been changed to “qx_”.
Refer to “Appendix A. Specification of Output Trace file” in page 81 for more information.

Chapter 5. Customization

Execution options can be customized by setting values in the parameter file.

Parameter file (configuration file)

Copy parameter file (qx.def file) in installation package to the directory prepared in installation step 2 (and also set as Data parameter of ApiExitLocal: stanza in qm.ini file) and modify parameters in the file.

File name “qx.def” is case sensitive. All letters of the file name are in lower case.

Syntax in parameter file

File format

File format of parameter file is simple text format. Each text line is delimited by carriage return.

Comment line, inline comment

All data after pound (#) character in each line are ignored as comment description.
If first non blank character in a line is pound character, whole line is ignored as comment description.
If pound character is found in the parameter line, all subsequent characters in the line are ignored as inline comment.

Blank lines are accepted. They are also ignored.

Parameter value

At most one parameter can be specified in each line.
All valid parameters should be specified in the following format.

parameter_name = parameter_value

Parameter name and parameter value are case sensitive. Be careful in specifying parameters. It is recommended to exploit parameter file supplied with the SupportPac package to avoid mistakes.

Blank characters before parameter name and parameter value are accepted. They are simply ignored.

Multiple parameter values are to be specified using commas for some parameters. Refer to the description of the parameters for more information. Blank characters before comma and after comma are ignored.

Summary of parameters in parameter file

All parameters are case sensitive. Specify uppercase characters and lowercase characters as they are.

Parameter **(case sensitive)**	**Description**	**Unit**	**Default value**	**Applicable exit type**	**Page**
Exection, Selection of output data format					
ExecFlag	Whether trace is enabled or not.	(N/A)	Yes	API Channel	44
ShowSummaryOnly	Whether only summary data is to be shown or not.	(N/A)	No	API Channel	56
ShowPerformanceOnly	Whether only performance summary data is to be shown or not.	(N/A)	No	API Channel	56
Output - general					
ShowParsedData	Whether parsed data is to be shown or not.	(N/A)	yes	API Channel	51
ShowHexCharImage	Whether character image of each byte is to be put to trace file.	(N/A)	yes	API Channel	49
ShowHexCharCcsid	CCSID for character image of hexadecimal data.	(N/A)	0	API Channel	47
ShowMaxLenCharv	Maximum MQCHARV data length to be put to trace file	Byte	-1 (all data)	API Channel	50
ShowMaxLenCharvString	Maximum MQCHARV data length to be put to trace file in string format	Byte	400	API Channel	50
Output – message body					
ShowParsedMsgContent	Whether parsed data is to be shown or not	(N/A)	yes	API Channel	53
MaxParsedOutputLineLength	Maximum line length of formatted message content	Byte	100	API Channel	44
ShowMaxLenParsedContent	Maximum data length to be parsed and put to trace file	Byte	-1 (all data)	API Channel	51
ShowMaxLenPut ShowMaxDataLengthPut	Maximum data length to be put to trace file for MQPUT/MQPUT1.	Byte	-1 (all data)	API	51
ShowMaxLenGet ShowMaxDataLengthGet	Maximum data length to be put to trace file for MQGET.	Byte	-1 (all data)	API	51
ShowMaxLenChannel ShowMaxDataLengthChannel	Maximum data length to be put to trace file for channel data	Byte	-1 (all data)	Channel	49
ShowDataConvOnGet	Whether conversion data is to be put to trace file for MQGET.	(N/A)	No	API	46
Output - supplemental data					
ShowMqVersion	Whether MQ version information is to be shown or not	(N/A)	Yes	API Channel	51
ShowProcessEnvValue	Whether process environment values are to be shown or not.	(N/A)	No	API Channel	56
ShowSummaryCorrelId	Whether correlation id is to be shown in summary trace or not.	(N/A)	Yes	API Channel	57
ShowChannelAttributes	Whether channel attributes are to be put to trace file	(N/A)	Yes	Channel	46
CountSelectedCallsOnly	Whether only selected API calls regarding TargetQueue parameter are counted.	(N/A)	No	API	43

Data selection					
TargetQueue	Target queue and process types (MQGET, MQPUT, MQINQ, MQSET) to be traced.	(N/A)	(all)	API	59
TargetProcess	Target process name to be traced.	(N/A)	(all)	API	58
ExcludeProcess	Target process name not to be traced.	(N/A)	(none)	API	43
ShowXaCalls	Whether XA calls are to be traced or not. (MQ V7.0.1 and later)	(N/A)	No	API	58
ShowSummaryDataSuppress	Suppress summary data in particular conditions.	(N/A)	(none)	API	57
Alert file					
PutAlertFileError	Whether alert file is generated for API error return code.	(N/A)	yes	API Channel	45
PutAlertFileWarning	Whether alert file is generated for API warning return code.	(N/A)	No	API Channel	45
PutAlertFileInfo	Whether alert file is generated for particular reason codes.	(N/A)	No	API Channel	45
PutAlertSuppress	Suppress alert for particular conditions.	(N/A)	(none)	API Channel	46
Process control					
ShowPerformanceIntervalTime	Minimum time interval time to show performance summary data.	Second	0 (only last)	API Channel	56
ShowPerformanceIntervalCount	API call interval to show performance summary data.	API call count	0 (only last)	API Channel	56
CheckUpdateRequestInterval	Time interval to check update request.	Second	30	API Channel	42
NoExitProcessIfProcNameMismatch	Exit process is not performed at all for each MQ API call if process name does not match.	(N/A)	No	API	44
Trace file					
MaxFileSize	Maximum file size of a trace file	Byte	100000000 (100MB)	API Channel	44
CompressCommand	Compress command to be invoked after closing each trace file.	(N/A)	(none)	API Channel	42
CompressFileSizeThreshold	Minimum file size for the invocation of compression command.	Byte	1000000 (1MB)	API Channel	42
CreateEmptyFile	Whether trace file not matching TargetProcess specification is to be generated or not.	(N/A)	yes	API Channel	43
WriteStream	Whether trace data is to be put in I/O stream mode or low level I/O mode	(N/A)	no	API Channel	61
WriteAfterEachApi	Whether trace data is to be written after processing each API.	(N/A)	yes	API Channel	59
WriteCache	Whether write cache is enabled or disabled.	(N/A)	yes	API Channel	60
WriteBufferSize	Size of buffer where trace data is cached before putting to file	Byte	10000 (10KB)	API Channel	60
WriteInterval	Time interval to be put to output file.	Second	30	API Channel	60

Description of each parameter

CheckUpdateRequestInterval

Abstract:	Time interval to check update request.
Description:	If specified time has been passed from previous update check, update request file created by qxset command is scanned. (Refer to “Chapter 6. Enabling/Disabling Trace Function” in page 64 for more information about qxset command.)
Valid values:	Numeric value. Unit is in second. If value 0 is specified, update check is not performed.
Default value:	30

CompressCommand

Abstract:	Compress command to be invoked after closing each trace file.
Description:	Command for trace file compression can be invoked right after closing trace file. This function is useful when you want to get huge trace data. Specified compression program is called only when file size is no less than CompressFileSizeThreshold parameter value when trace file is closed.
Valid values:	Example of using gzip command for trace file compression is `CompressCommand = gzip $` where dollar character ($) is to be substituted by actual trace file name at invocation time of compress command. If compression program is not in common path, specify path information together. (Example: `CompressCommand = D:¥dir_a¥gzip $`)
Default value:	(none)

CompressFileSizeThreshold

Abstract:	Minimum file size for the invocation of compress command.
Description:	If file size of trace file is no less than specified value, compression command is called right after closing trace file. This parameter is ignored when CompressCommand parameter is not specified.
Valid values:	Numeric value. Unit is in bytes. Minimum byte is 100000 (100KB).
Default value:	1000000 (1MB)

CountSelectedCallsOnly

Abstract:	Whether only selected API calls regarding TargetQueue parameters are counted.
Description:	If “yes” is specified, only selected API calls are counted in summary data. If “no” is specified, all API calls in each thread are counted.
Valid values:	“yes”, “no”
Default value:	“no” (All API calls are counted.)

CreateEmptyFile

Abstract:	Whether trace file not matching TargetProcess parameter specification is to be generated or not. Whether trace file is to be generated or not when ExecFlag=no is specified.
Description:	If “yes” is specified or this parameter is not specified, process parameters are put to trace file for verification of input parameters. If “no” is specified, trace file is not generated at all.
Valid values:	“yes”, “no”
Default value:	“yes”

ExcludeProcess

Abstract:	Target process name not to be traced.
Description:	If MQ API call for specific process name is not to be traced, it can be done by specifying ExcludeProcess parameters. Up to 100 ExcludeProcess parameters can be specified. Asterisk (*) character can be specified at the last byte of queue name as generic specification. Parameter values are case insensitive. Lowercase characters and uppercase characters are not distinguished. This parameter is ignored if one or more TargetProcess parameters are specified.
Valid values:	Any character string values
Default value:	(none - all MQ application programs are to be traced).

ExecFlag

Abstract:	Whether trace function is to be enabled or not.
Description:	If value “yes” is specified, trace data is put to trace file. If value “no” is specified, no trace data is put to trace file. If value “env” is specified, trace data is put to trace file only for processes running with MA0W environment variable. (non-null value) This parameter is useful for temporarily disabling tracing.
Valid values:	“yes”, “no”, “env”
Default value:	“yes”

MaxFileSize

Abstract:	Maximum file size of a trace file.
Description:	If total size of a trace file exceeds specified size, file is closed and then new file is opened for next output.
Valid values:	Numeric value. Unit is in byte. Minimum value is 1000000 (1MB). Maximum value is 2000000000 (2GB).
Default value:	100000000 (100MB)

MaxParsedOutputLineLength

Abstract:	Maximum line length of formatted message content.
Description:	This parameter is ignored if “ShowParsedMsgContent = no” is specified.
Valid values:	Numeric value. Unit is in byte. Minimum value is 40. Maximum value is 999.
Default value:	100

NoExitProcessIfProcNameMismatch

Abstract:	Exit process is not performed at all for each MQ API call if process name does not match.
Description:	When TargetProcess/ExcludeProcess parameters are specified and process name does not meet the specified criteria, exit module is not called in each MQ API call at all if value “yes” is specified to the parameter. This parameter is effective to minimize impact to existing MQ environment. Note that trace function cannot be enabled or disabled dynamically to the processes not matching TargetProcess/ExcludeProcess parameters if value “yes” is specified to the parameter.
Valid values:	“yes”, “no”
Default value:	“no”

PutAlertFileError

Abstract:	Whether alert file is to be generated when error return code was returned.
Description:	Specify whether alert file is to be generated when error return code was returned by MQ API call or not.
Valid values:	“yes”, “no”
Default value:	“yes”

PutAlertFileInfo

Abstract:	Whether alert file is to be generated when process completed with particular reason codes.
Description:	Specify whether alert file is to be generated when one of the following reason codes is returned. ● 2033 (MQRC_NO_MSG_AVAILABLE) ● 2059 (MQRC_Q_MGR_NOT_AVAILABLE) ● 2161 (MQRC_Q_MSG_QUIESCING) ● 2437 (MQRC_NO_RETAINED_MSG) ● 2471 (MQRC_PROPERTY_NOT_AVAILABLE)
Valid values:	“yes”, “no”
Default value:	“no”

PutAlertFileWarning

Abstract:	Whether alert file is to be generated when warning return code was returned.
Description:	Specify whether alert file is to be generated when warning return code was returned by MQ API call or not.
Valid values:	“yes”, “no”
Default value:	“no”

PutAlertSuppress

Abstract:	Suppress alert for particular conditions.
Description:	Specify sets of completion code, reason code, API name, object name and process name. Alert information is not put to alert file if condition matches specified set of parameter values. Up to 100 PutAlertSuppress parameters can be specified.
Valid values:	Syntax is: `PutAlertSuppress = `*compcode*, *reason*, *apiname*, *qname*, *processname*

Where:

compcode	Completion code. One of 0, 1, 2, asterisk (*). (* = any)
reason	4-digit decimal value or asterisk (*). (* = any)
apiname	API name or asterisk (*). (* = any)
qname	Queue name. Asterisk (*) character can be specified at the last byte of queue name as generic specification.
processname	Process name. Asterisk (*) character can be specified at the last byte of process name as generic specification.

Default value:	(none)	
Example:	PutAlertSuppress=2,2085,*,QTEST*,*	Alert for error with CompCode=2 and Reason=2085 is suppressed for queues beginning with QTEST.

ShowChannelAttributes

Abstract:	Whether channel attributes are to be put to trace file.
Description:	Specify whether channel attribute values are to be put to trace file or not in starting channel.
Valid values:	“yes”, “no”
Default value:	“yes”

ShowDataConvOnGet

Abstract:	Whether message data before data conversion is to be put to trace file.
Description:	Specify whether message data before data conversion in MQGET process is to be put to trace file or not.
Valid values:	“yes”, “no”
Default value:	“no”

ShowHexCharCcsid

Abstract: CCSID for character image of hexadecimal data

Description: Specify CCSID to be used for character representation of hexadecimal data. CCSIDs for multi byte character set (MBCS) are supported so that byte string of a character is not divided into multiple lines.

This parameter is ignored when “ShowHexCharImage=no” is specified.

For displaying and printing all characters correctly, devices and programs must be selected appropriately. Supplemental environment setting such as locale and font might be required.

Valid values: Numeric value.
The following values are supported. All other values are treated as default value.

CCSID	Description
0	7-bit ASCII
-1	Automatic CCSID selection regarding locale
819	ISO 8859-1 ASCII
437	USA PC
850	Latin 1 PC
932, 942, 943	Japanese PC
954, 5050, 33722	Japanese EUC
949, 1363	Korean PC
970	Korean EUC
1381	Simplified Chinese PC (IBM GB)
1386	Simplified Chinese PC (GBK)
5488	Simplified Chinese PC (GB18030)
1383	Simplified Chinese EUC
950	Traditional Chinese PC (Big 5)
1370	Traditional Chinese PC
964	Traditional Chinese EUC
1208	UTF-8

On Windows, value is set using system default if value -1 (automatic CCSID selection) is specified.

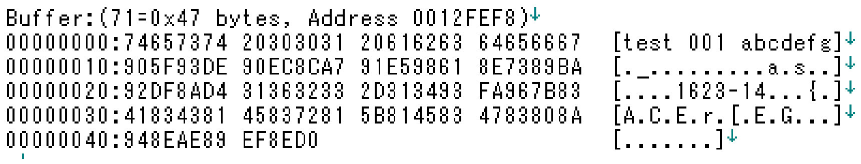
On UNIX, value is set regarding locale setting if value -1 (automatic CCSID selection) is specified.

Locale	Default CCSID	Description
en_US	819	ISO 8859-1 ASCII (Latin 1)
Ja_JP Ja_JP.IBM-932 Ja_JP.IBM-943 ja_JP.PCK ja_JP.SJIS	943 (932, 942)	Japanese PC
ja_JP ja_JP.eucJP ja_JP.eucjp ja_JP.IBM-eucJP ja_JP.ujis japanese japanese.euc ja	954 (5050, 33722)	Japanese EUC

ko_KR ko_KR.eucKR ko_KR.euckr ko_KR.IBM-eucKR Korean korean.euc ko	970	Korean EUC
Zh_CN Zh_CN.GBK zh_CN.gb2312 zh_CN.gbk zh_CN.hp15CN	1386	Simplified Chinese PC (GBK)
zh_CN.gb18030	5488	Simplified Chinese PC (GB18030)
zh_CN zh_CN.IBM-eucCN	1383	Simplified Chinese EUC
Zh_TW.big5 Zh_TW zh_TW.BIG5 zh_TW.big5 zh_TW.ccdc	950	Traditional Chinese PC (Big 5)
zh_TW zh_TW.eucTW zh_TW.euctw zh_TW.IBM-eucTW	964	Traditional Chinese EUC
(any).UTF-8 *(any)*.utf8	1208	UTF-8

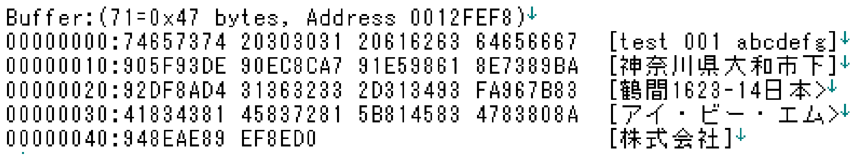
Default value: 0 (7-bit ASCII)

Sample output: ShowHexCharCcsid=0 (default)



```
Buffer:(71=0x47 bytes, Address 0012FEF8)
00000000:74657374 20303031 20616263 64656667  [test 001 abcdefg]
00000010:905F93DE 90EC8CA7 91E59861 8E7389BA  [._.........a.s..]
00000020:92DF8AD4 31363233 2D313493 FA967B83  [....1623-14...{.]
00000030:41834381 45837281 5B814583 4783808A  [A.C.E.r.[.E.G...]
00000040:948EAE89 EF8ED0                      [.......]
```

ShowHexCharCcsid=943 (Japanese PC)



```
Buffer:(71=0x47 bytes, Address 0012FEF8)
00000000:74657374 20303031 20616263 64656667  [test 001 abcdefg]
00000010:905F93DE 90EC8CA7 91E59861 8E7389BA  [神奈川県大和市下]
00000020:92DF8AD4 31363233 2D313493 FA967B83  [鶴間1623-14日本>
00000030:41834381 45837281 5B814583 4783808A  [アイ・ビー・エム>
00000040:948EAE89 EF8ED0                      [株式会社]
```

ShowHexCharImage

Abstract:	Whether character image is to be put to trace file.
Description:	If “ShowHexCharImage=yes” is specified, character image for each hexadecimal expression is put to trace file, in addition.
Valid values:	“yes”, “no”
Default value:	“yes”
Sample output:	ShowHexCharImage=yes (default)

```
|--- Seq # : 5 ----------------------------------------
[2006/05/01 17:22:16.625]0000020422-0949362359
<MQOPEN
(BEFORE
Hconn:0x009434A0
OD:(336=0x150 bytes, Address 00D3F2B0)
00000000:4F442020 03000000 01000000 53595354  [OD  ........SYST]
00000010:454D2E41 444D494E 2E535441 54495354  [EM.ADMIN.STATIST]
00000020:4943532E 51554555 45202020 20202020  [ICS.QUEUE       ]
00000030:20202020 20202020 20202020 00000000  [            ....]
00000040:00000000 00000000 00000000 00000000  [................]
00000050-0000005F same as above
00000060:00000000 00000000 00000000 414D512E  [............AMQ.]
00000070:2A000000 00000000 00000000 00000000  [*...............]
00000080:00000000 00000000 00000000 00000000  [................]
00000090-0000014F same as above
```

ShowHexCharImage=no

```
|--- Seq # : 5 ----------------------------------------
[2006/05/01 17:22:16.625]0000020422-0949362359
<MQOPEN
(BEFORE
Hconn:0x009434A0
OD:(336=0x150 bytes, Address 00D3F2B0)
00000000:4F442020 03000000 01000000 53595354
00000010:454D2E41 444D494E 2E535441 54495354
00000020:4943532E 51554555 45202020 20202020
00000030:20202020 20202020 20202020 00000000
00000040:00000000 00000000 00000000 00000000
00000050-0000005F same as above
00000060:00000000 00000000 00000000 414D512E
00000070:2A000000 00000000 00000000 00000000
00000080:00000000 00000000 00000000 00000000
00000090-0000014F same as above
```

ShowMaxLenChannel / ShowMaxDataLengthChannel

Abstract:	Maximum message data length to be put to trace file for channel data.
Description:	If maximum message data length in trace file for channel is to be limited, it can be done by specifying this parameter.
Valid values:	Numeric value. Unit is in byte. Specify value -1 for showing all data without truncating.
Default value:	-1 (all data is to be put to trace file without truncating)

ShowMaxLenCharv

Abstract:	Maximum MQCHARV data length to be put to trace file
Description:	If maximum message data length in trace file for variable lenth data (MQCHARV data) is to be limited, it can be done by specifying this parameter.
Valid values:	Numeric value. Unit is in byte. Specify value -1 for showing all data without truncating.
Default value:	-1 (all data is to be put to trace file without truncating)

Sample output:

```
Variable-SelectionString:(10640=x2990 bytes, Address 0x00B2C200)
00000000:20202020 20202020 514D3120 20202020  [        QM1     ]
00000010:20202020 20202020 20202020 20202020  [                ]
00000020-0000002F same as above
00000030:20202020 20202020 38000000 00000001  [        8.......]
00000040:00000024 00000001 00000002 00000001  [...$............]
00000050:00000001 00000000 00000000 00000001  [................]
00000060:00000005 00000014 000003E9 00000001  [................]
00000070:000003F1 000004EE 00000002 00000000  [................]
00000080:00000000 00000000 00000000 00000000  [................]
00000090-0000030F same as above
00000310:00000000 00                          [.....]
  (Shown only leading 789 byte(s) ...)      <== When value is 789
```

ShowMaxLenCharvString

Abstract:	Maximum MQCHARV data length to be put to trace file in string format
Description:	If maximum message data length in trace file for variable lenth data (MQCHARV data) in string format is to be limited, it can be done by specifying this parameter.
Valid values:	Numeric value. Unit is in byte. Specify value -1 for showing all data without truncating.
Default value:	400

Sample output:

```
|> SelectionString        :
|>   VSPtr                : (null)
|>   VSOffset             : 400
|>   VSBufSize            : 10240
|>   VSLength             : 0
|>   VSCCSID              : 1208 (UTF-8 with IBM PUA)
|>   Value (in string)    :
|>           QM1                                      8.........$........................
|>    ...........................................................................................
|>    ...........................................................................................
|>    ...........................................................................................
|>    ...........................
|>     (Shown only leading 400 byte(s) ...)
```

ShowMaxLenGet / ShowMaxDataLengthGet

Abstract:	Maximum message data length to be put to trace file for MQGET.
Description:	If maximum message data length in trace file for MQGET is to be limited, it can be done by specifying this parameter.
Valid values:	Numeric value. Unit is in byte. Specify value -1 for showing all data without truncating.
Default value:	-1 (all data is to be put to trace file without truncating)

ShowMaxLenParsedContent

Abstract:	Maximum message data length to be parsed and put to trace file
Description:	If maximum message data length to be parsed is to be limited, it can be done by specifying this parameter. This parameter is ignored if “ShowParsedMsgContent = no” is specified.
Valid values:	Numeric value. Unit is in byte. Specify value -1 for showing all data without truncating.
Default value:	-1 (all data is to be put to trace file without truncating)

ShowMaxLenPut / ShowMaxDataLengthPut

Abstract:	Maximum message data length to be put to trace file for MQPUT and MQPUT1.
Description:	If maximum message data length in trace file for MQPUT and MQPUT1 is to be limited, it can be done by specifying this parameter.
Valid values:	Numeric value. Unit is in byte. Specify value -1 for showing all data without truncating.
Default value:	-1 (all data is to be put to trace file without truncating)

ShowMqVersion

Abstract:	Whether version information of WebSphere MQ is to be shown or not.
Description:	Specify whether version information of WebSphere MQ is to be shown or not. If value “yes” is set, information equivalent to the output of dspmqver command is shown in the process profile. (Refer to page 62 for example.)
Valid values:	“yes”, “no”
Default value:	“yes”

ShowParsedData

Abstract:	Whether parsed data is to be shown or not.
Description:	If “ShowParsedData=yes” is specified, attribute values are shown in addition to graphic hexadecimal output.
Valid values:	“yes”, “no”
Default value:	“yes”
Sample output:	ShowParsedData=yes (default)

```
|--- Seq # : 5 ---------------------------------------
[2006/05/01 17:22:16.625]0000020422-0949362359
<MQOPEN
(BEFORE
Hconn:0x009434A0
OD:(336=0x150 bytes, Address 00D3F2B0)
00000000:4F442020 03000000 01000000 53595354  [OD  ........SYST]
00000010:454D2E41 444D494E 2E535441 54495354  [EM.ADMIN.STATIST]
00000020:4943532E 51554555 45202020 20202020  [ICS.QUEUE       ]
00000030:20202020 20202020 20202020 00000000  [            ....]
00000040:00000000 00000000 00000000 00000000  [................]
00000050-0000005F same as above
00000060:00000000 00000000 00000000 414D512E  [............AMQ.]
00000070:2A000000 00000000 00000000 00000000  [*...............]
00000080:00000000 00000000 00000000 00000000  [................]
00000090-0000014F same as above
|> Version                : 3
|> ObjectType             : QUEUE
|> ObjectName             : SYSTEM.ADMIN.STATISTICS.QUEUE
|> ObjectQMgrName         :
|> DynamicQName           : AMQ.*
|> AlternateUserId        :
|> RecsPresent            : 0
|> KnownDestCount         : 0
|> UnknownDestCount       : 0
|> InvalidDestCount       : 0
|> AlternateSecurityId    : (null)
|> ResolvedQName          :
|> ResolvedQMgrName       :
| ObjectName:SYSTEM.ADMIN.STATISTICS.QUEUE
Options:0x00000810
|> Options                : OUTPUT / SET_ALL_CONTEXT
```

ShowParsedData=no

```
|--- Seq # : 5 ---------------------------------------
[2006/05/01 17:22:16.625]0000020422-0949362359
<MQOPEN
(BEFORE
Hconn:0x009434A0
OD:(336=0x150 bytes, Address 00D3F2B0)
00000000:4F442020 03000000 01000000 53595354  [OD  ........SYST]
00000010:454D2E41 444D494E 2E535441 54495354  [EM.ADMIN.STATIST]
00000020:4943532E 51554555 45202020 20202020  [ICS.QUEUE       ]
00000030:20202020 20202020 20202020 00000000  [            ....]
00000040:00000000 00000000 00000000 00000000  [................]
00000050-0000005F same as above
00000060:00000000 00000000 00000000 414D512E  [............AMQ.]
00000070:2A000000 00000000 00000000 00000000  [*...............]
00000080:00000000 00000000 00000000 00000000  [................]
00000090-0000014F same as above
| ObjectName:SYSTEM.ADMIN.STATISTICS.QUEUE
Options:0x00000810
```

ShowParsedMsgContent

Abstract: Whether message content is to be shown in parsed format or not.

Description: If “ShowParsedMsgContent=yes” is specified, message text is shown in parsed format.

The following formats are supported in current release.

- Dead letter header (MQDEAD)
- Embedded PCF (MQHEPCF)
- Event (MQEVENT)
- Message descriptor extension (MQHMDE)
- Programmable Command Format (MQPCF, MQADMIN)
- Rules and formatting header (MQHRF)
- Rules and formatting header 2 (MQHRF2)
- Trigger message (MQTRIG)
- Transmission queue header (MQXMIT)
- Distribution header (MQHDIST)
- Reference message header (MQHREF)
- Extended markup language (XML)

For extended markup language (XML), data is assumed as XML format only when all of the following conditions are met.

- First byte is “<”.
- Last nonblank byte is “>”.
- All “<” characters and “>” characters are completely paired.

Valid values: “yes”, “no”

Default value: “yes”

Sample output: Trigger message (MQTRIG)

```
DataLength:684
Buffer:(684=0x2AC bytes, Address 0012F3BC)
00000000:544D2020 01000000 54524947 47455251  [TM  ....TRIGGERQ]
00000010:20202020 20202020 20202020 20202020  [                ]
00000020-0000002F same as above
00000030:20202020 20202020 50524F43 32202020  [        PROC2   ]
00000040:20202020 20202020 20202020 20202020  [                ]
00000050-0000009F same as above
000000A0:20202020 20202020 0B000000 653A5C6D  [        ....e:¥m]
000000B0:71746F6F 6C735C6D 6973635C 62617466  [qtools¥misc¥batf]
000000C0:696C6531 2E626174 20202020 20202020  [ile1.bat        ]
000000D0:20202020 20202020 20202020 20202020  [                ]
000000E0-0000021F same as above
00000220:20202020 20202020 20202020 66697273  [            firs]
00000230:74207365 636F6E64 20746869 72642020  [t second third  ]
00000240:20202020 20202020 20202020 20202020  [                ]
00000250-0000029F same as above
000002A0:20202020 20202020 20202020           [            ]
 |> offset 0 (=0x0) - MQTM (Trigger message)
 |> Version                    : 1
 |> QName                      : TRIGGERQ
 |> ProcessName                : PROC2
 |> TriggerData                :
 |> ApplType                   : Windows
 |> ApplId                     : e:¥mqtools¥misc¥batfile1.bat
 |> EnvData                    :
 |> UserData                   : first second third
```

(continue to next page)

Sample output:
(continued)

Programmable Command Format (MQPCF)

```
BufferLength:96
Buffer:(96=0x60 bytes, Address 003FDBD8)
00000000:01000000 24000000 01000000 0D000000  [....$...........]
00000010:01000000 01000000 00000000 00000000  [................]
00000020:03000000 04000000 18000000 E0070000  [................]
00000030:00000000 01000000 2A000000 03000000  [........*.......]
00000040:10000000 14000000 01000000 05000000  [................]
00000050:14000000 EA030000 01000000 F1030000  [................]
 |> ---------- offset 0 (=0x0) - MQCFH (PCF header and parameters)
 |> Type                      : COMMAND
 |> StrucLength               :     36
 |> Version                   :      1
 |> Command                   :     13 (INQUIRE_Q)
 |> MsgSeqNumber              :      1
 |> Control                   : LAST
 |> CompCode                  :      0
 |> Reason                    :      0
 |> ParameterCount            :      3
 |> PCF parm --------------- : --- 1 --- offset 36 (=0x24)
 |> PCF parm - Type           : STRING
 |> PCF parm - Length         :     24
 |> PCF parm - Parameter      :   2016 (Q_NAME)
 |> PCF parm - CodedCharSetId: default / queue_manager
 |> PCF parm - Length         :      1
 |> PCF parm - Value          : *
 |> PCF parm --------------- : --- 2 --- offset 60 (=0x3C)
 |> PCF parm - Type           : INTEGER
 |> PCF parm - Length         :     16
 |> PCF parm - Parameter      :     20 (Q_TYPE)
 |> PCF parm - Value          :      1 (local)
 |> PCF parm --------------- : --- 3 --- offset 76 (=0x4C)
 |> PCF parm - Type           : INTEGER_LIST
 |> PCF parm - Length         :     20
 |> PCF parm - Parameter      :   1002 (CF_Q_ATTRS)
 |> PCF parm - Count          :      1
 |> PCF parm - Value          :   1009 (CF_ALL)
 -
```

Rules and formatting header 2 (MQHRF2)

```
BufferLength:198
Buffer:(198=0xC6 bytes, Address 00BD0020)
00000000:52464820 02000000 84000000 22020000  [RFH ........″...]
00000010:BA130000 4D515354 52202020 00000000  [....MQSTR   ....]
00000020:B8040000 5C000000 3C6D6364 3E3C4D73  [....¥...<mcd><Ms]
00000030:643E4D52 4D3C2F4D 73643E3C 5365743E  [d>MRM</Msd><Set>]
00000040:444D5051 4F434330 37393939 393C2F53  [DMPQOCC079999</S]
00000050:65743E3C 54797065 3E4D5347 5F4D4958  [et><Type>MSG_MIX]
00000060:45444341 53455F49 443C2F54 7970653E  [EDCASE_ID</Type>]
00000070:3C466D74 3E435746 3C2F466D 743E3C2F  [<Fmt>CWF</Fmt></]
00000080:6D63643E 4F73616D 7520496E 6F756520  [mcd>Osamu Inoue ]
00000090:536F7461 20496E6F 75658140 814082A0  [Sota Inoue.@.@..]
000000A0:82A282A4 20205361 746F6520 82A682A6  [....  Satoe ....]
000000B0:82A682A6 8140B0B3 B2A9B3C2 B3C9F2CD  [.....@..........]
000000C0:F2F4E9DA D9F8                        [......]
 |> ---------- offset 0 (=0x0) - MQHRF2 (Rules and formatting header 2)
 |> Version                   : 2
 |> StrucLength               : 132
 |> Encoding                  : 546 (Native encoding) INTEGER_REVERSED / DECIM
 |> CodedCharSetId            : 5050
 |> Format                    : MQSTR
 |> Flags                     : 0
 |> NameValueCCSID            : 1208
 |> ---------- offset 40 (=0x28) - XML data
 |> <mcd>
 |>   <Msd>
 |>     MRM
 |>   </Msd>
 |>   <Set>
 |>     DMPQOCC079999
 |>   </Set>
 |>   <Type>
 |>     MSG_MIXEDCASE_ID
 |>   </Type>
 |>   <Fmt>
 |>     CWF
 |>   </Fmt>
 |> </mcd>
```

Sample output: (continued)

Extended markup language (XML)

```
DataLength:430
Buffer:(430=0x1AE bytes, Address 009E51B4)
00000000:3C42726F 6B657220 75756964 3D226535  [<Broker uuid="e5]
00000010:33333234 39362D30 3930312D 30303030  [332496-0901-0000]
00000020:2D303038 302D6437 39616664 31333037  [-0080-d79afd1307]
00000030:6264223E 3C457865 63757469 6F6E4772  [bd"><ExecutionGr]
00000040:6F757043 6F6D706C 6574696F 6E436F64  [oupCompletionCod]
00000050:65207575 69643D22 30646331 34353961  [e uuid="0dc1459a]
00000060:2D303930 312D3030 30302D30 3038302D  [-0901-0000-0080-]
00000070:61393461 30643865 30633931 22207265  [a94a0d8e0c91" re]
00000080:73756C74 3D227375 63636573 73223E3C  [sult="success"><]
00000090:4C6F6745 6E747279 20636174 616C6F67  [LogEntry catalog]
000000A0:3D224249 50763630 3022206E 756D6265  [="BIPv600" numbe]
000000B0:723D2234 30343022 3E3C496E 73657274  [r="4040"><Insert]
000000C0:20747970 653D2273 7472696E 67222074  [ type="string" t]
000000D0:6578743D 22454730 31222F3E 3C496E73  [ext="EG01"/><Ins]
000000E0:65727420 74797065 3D227374 72696E67  [ert type="string]
000000F0:22207465 78743D22 30646331 34353961  [" text="0dc1459a]
00000100:2D303930 312D3030 30302D30 3038302D  [-0901-0000-0080-]
00000110:61393461 30643865 30633931 222F3E3C  [a94a0d8e0c91"/><]
00000120:2F4C6F67 456E7472 793E3C2F 45786563  [/LogEntry></Exec]
00000130:7574696F 6E47726F 7570436F 6D706C65  [utionGroupComple]
00000140:74696F6E 436F6465 3E3C5374 61727452  [tionCode><StartR]
00000150:6573706F 6E73653E 3C457865 63757469  [esponse><Executi]
00000160:6F6E4772 6F757020 75756964 3D223064  [onGroup uuid="0d]
00000170:63313435 39612D30 3930312D 30303030  [c1459a-0901-0000]
00000180:2D303038 302D6139 34613064 38653063  [-0080-a94a0d8e0c]
00000190:3931222F 3E3C2F53 74617274 52657370  [91"/></StartResp]
000001A0:6F6E7365 3E3C2F42 726F6B65 723E      [onse></Broker>]
|> ---------- offset 0 (=0x0) - XML data
|> <Broker uuid="e5332496-0901-0000-0080-d79afd1307bd">
|>   <ExecutionGroupCompletionCode uuid="0dc1459a-0901-0000-0080-a94a0d8e0c91"
result="success">
|>     <LogEntry catalog="BIPv600" number="4040">
|>       <Insert type="string" text="EG01"/>
|>       <Insert type="string" text="0dc1459a-0901-0000-0080-a94a0d8e0c91"/>
|>     </LogEntry>
|>   </ExecutionGroupCompletionCode>
|>   <StartResponse>
|>     <ExecutionGroup uuid="0dc1459a-0901-0000-0080-a94a0d8e0c91"/>
|>   </StartResponse>
|> </Broker>
-
```

ShowPerformanceIntervalCount

Abstract:	API call interval to show performance summary data.
Description:	If positive number is specified, performance summary is shown after calling every specified count of MQ API calls.
Valid values:	Numeric value. Unit is in API call count.
Default value:	0 (only last of trace data)

ShowPerformanceIntervalTime

Abstract:	Minimum time interval to show performance summary data.
Description:	If positive number is specified, performance summary is shown when specified second has passed from previous summary data.
Valid values:	Numeric value. Unit is in second.
Default value:	0 (only last of trace data)

ShowPerformanceOnly

Abstract:	Whether only performance summary data is to be shown or not.
Description:	If value “yes” is specified, only performance summary data is shown without data for each API call.
Valid values:	“yes”, “no”
Default value:	“no”
Sample output:	Refer to page 97 in “Appendix C. Sample Trace Data”.

ShowProcessEnvValue

Abstract:	Whether process environment values are to be shown or not.
Description:	If value “yes” is specified, environment variables for current thread are shown to trace file.
Valid values:	“yes”, “no”
Default value:	“no”

ShowSummaryCorrelId

Abstract:	Whether correlation id is to be shown in summary trace or not.
Description:	If value “yes” is specified, correlation id is shown to summary trace file. This parameter is for summary trace. It is ignored if “ShowSummaryOnly=no” or “ShowPerformanceOnly=yes” is specified.
Valid values:	“yes”, “no”
Default value:	“yes”

ShowSummaryDataSuppress

Abstract: Suppress summary data in particular conditions.

Description: Specify sets of completion code, reason code, API name and object name. Summary data record is not put to trace file if condition matches specified set of parameter values.

Up to 100 ShowSummaryDataSuppress parameters can be specified.

This parameter is effective only when “ShowSummaryOnly=yes” is specified.

Valid values: Syntax is:

```
  ShowSummaryDataSuppress = compcode, reason, apiname, qname
```

Where:

compcode	Completion code. One of 0, 1, 2, asterisk (*). (* = any)
reason	4-digit decimal value or asterisk (*). (* = any)
apiname	API name or asterisk (*). (* = any)
qname	Queue name. Asterisk (*) character can be specified at the last byte of queue name as generic specification.

Default value: (none)

Example: ShowSummaryDataSuppress=2,2033,MQGET,QTEST*

(Summary record is not put to trace file when CompCode=2, Reason=2033 and first five bytes of queue name is QTEST for MQGET calls.)

Sample output:

ShowSummaryDataSuppress=2,*,MQPUT,*
ShowSummaryDataSuppress=*,*,MQDISC,*

```
   Time                                          API      CC Reason  Time(sec)  ObjectName
 1 [2006/10/14 23:33:38.687]0000000041-0512362890  MQCONNX   0     0      0.001743
 2 [2006/10/14 23:33:38.703]0000000041-0512363471  MQOPEN    0     0      0.000105  Q8
 4 [2006/10/14 23:33:38.703]0000000041-0512372793  MQCLOSE   0     0      0.000067  Q8
```

Default (no ShowSummarySuppress parameter is specified)

```
   Time                                          API      CC Reason  Time(sec)  ObjectName
 1 [2006/10/14 23:33:38.687]0000000041-0512362890  MQCONNX   0     0      0.001743
 2 [2006/10/14 23:33:38.703]0000000041-0512363471  MQOPEN    0     0      0.000105  Q8
 3 [2006/10/14 23:33:38.703]0000000041-0512364310  MQPUT     2 2051      0.000199  Q8    (2051 = MQRC_
 4 [2006/10/14 23:33:38.703]0000000041-0512372793  MQCLOSE   0     0      0.000067  Q8
 5 [2006/10/14 23:33:38.703]0000000041-0512373273  MQDISC    0     0      0.000101
```

ShowSummaryOnly

Abstract:	Whether only summary data is to be shown or not.
Description:	If value “yes” is specified, only summary data for each MQ API call is shown in an output record. If value “no” is specified or this parameter is not specified, detail data is shown.
Valid values:	“yes”, “no”
Default value:	“no”
Sample output:	Refer to page 98 and page 100 in “Appendix C. Sample Trace Data”.

ShowXaCalls

Abstract:	Whether calls for XA transaction are to be traced or not
Description:	If value “yes” is specified, XA calls are traced in additiona to MQ API calls. This parameter specification is effective on MQ V7.0.1 and subsequent releases.
Valid values:	“yes”, “no”
Default value:	“no”

TargetProcess

Abstract:	Target process name to be traced.
Description:	If MQ API call only for specific process name is to be traced, it can be done by specifying TargetProcess parameters. Up to 100 TargetProcess parameters can be specified. Asterisk (*) character can be specified at the last byte of queue name as generic specification. Parameter values are case insensitive. Lowercase characters and uppercase characters are not distinguished. If TargetProcess parameter is not specified, it is assumed that all MQ application programs are to be traced. On the other hand, if at least one TargetProcess parameter is specified, all MQ applications that do not match specified process name are not traced.
Valid values:	Any character string values
Default value:	(none - all MQ application programs are to be traced).

TargetQueue

Abstract:	Target queue and process type to be traced
Description:	If API calls only for specific MQ queues are to be traced, it can be done by specifying TargetQueue parameters. MQOPEN and MQCLOSE are traced only for specified MQ queues. MQ API calls independent of specific queues such as MQCONN, MQDISC and MQCMIT are always traced regardless of this parameter specification. Up to 100 TargetQueue parameters can be specified. If at least one TargetQueue parameter is specified, all API calls for MQ queues not matching the specification are not traced.
Valid values:	Syntax is: `TargetQueue = ` *`queue_name`* ` [,put][,get][,inq][,set]` Asterisk (*) character can be specified at the last byte of queue name as generic specification.
Default value:	(none - all MQ queues and all of MQPUT/MQPUT1/MQGET/MQINQ /MQSET calls are to be traced).

Examples:

`TargetQueue = Q1`	(All MQ API calls for queue Q1 are to be traced.)
`TargetQueue = Q2,put`	(MQPUT/MQPUT1 calls for queue Q2 are to be traced.)
`TargetQueue = Q3*,get,inq,set`	(MQGET/MQINQ/MQSET calls for queues whose name begins with 'Q3' are to be traced.)
`TargetQueue = (queue_manager)`	(MQINQ/MQSET calls for object type MQOT_Q_MGR are to be traced.)

WriteAfterEachApi

Abstract:	Whether trace data is to be put to file right after processing for each API.
Description:	If value "yes" is specified, file output operation is performed when whole trace data for an API call is put to buffer even if buffer is not full. This parameter is effective only when "WriteStream=no" and "WriteCache=yes" are specified.
Valid values:	"yes", "no"
Default value:	"yes"

WriteBufferSize

Abstract:	Size of buffer where trace data is cached before putting to file.
Description:	API exit module caches all data to be put to trace file at each MQ API call and performs file output operation only when buffer becomes full. When MQDISC API is called, all remaining data in the buffer is put to trace file. Specified size of memory area is allocated in each MQ thread when MQCONN or MQCONNX API is called. Allocated memory area is freed when MQDISC API is called. If specified value is less than minimum size, value is set to minimum size. This parameter is effective only when “WriteStream=no” and “WriteCache=yes” are specified.
Valid values:	Numeric value. Unit is in byte. Minimum size is 5000 (5KB).
Default value:	10000 (10KB)

WriteCache

Abstract:	Whether write cache is enabled or disabled.
Description:	When value “yes” is specified, API exit module caches all data to be put to trace file at each MQ API call and performs file output operation only when buffer becomes full When value “no” is specified, all data is immediately put to trace file at each MQ API call without using write buffer. This parameter is effective only when “WriteStream=no” is specified.
Valid values:	“yes”, “no”
Default value:	“yes”

WriteInterval

Abstract:	Time interval for file output operation.
Description:	If specified time interval has been passed from previous API call, file output operation is to be performed even if buffer is not full. This parameter is effective only when “WriteStream=no” is specified.
Valid values:	Numeric value. Unit is in second. If value 0 is specified, this parameter is ignored.
Default value:	30

WriteStream

Abstract:	Whether trace data is to be put in I/O stream mode or low level I/O mode.
Description:	When “WriteStream=no” is specified, trace data is put using low level I/O functions, open, write and close. Actual write timing is controlled by API exit module regarding WriteCache parameter, WriteBufferSize parameter and WriteInterval parameter. When “WriteStream=yes” is specified, trace data is put using I/O stream functions, fopen, fputs and fclose. WriteCache parameter, WriteBufferSize parameter and WriteInterval parameter are ignored. Though performance might become better, actual write timing might be delayed because actual write timing is controlled by operating system.
Valid values:	“yes”, “no”
Default value:	“no”

How to check process parameters at execution time

Parameter file (qx.def file) referred at execution time and all parameters referred in the file can be identified by checking values shown at the beginning of trace file.

```
|-----------------------------------------------------------
|Program information
|  Program name       : WebSphere MQ API Exit (SupportPac MA0W)
|  Version            : 7.00
|  Provided by        : oinoue@jp.ibm.com
|Program attributes
|  Addressing mode    : 32-bit
|  Threading          : Common to single thread and multi thread
|  Process level      : 700
|WebSphere MQ
|  Version            : 7.0.0.0
|  CMVC level         : p000-L080118
|  Build type         : IKAP - (Production)
|Operating environment
|  Operating system   : Windows Server 2003
|  Major version      : 5
|  Minor version      : 2
|  Service level      : Service Pack 2
|  Build number       : 3790
|  Computer name      : RS480M2
|API Exit context
|  Exit name          : ma0w
|  Function name      : qxEntry
|  API caller type    : EXTERNAL
|  Queue manager      : QMV7
|  Environment        : OTHER
|  User id            : iselmq
|  Connection name    :
|  Appl/Channel name  : Q¥testerformq_v7_mqm_r32.exe
|  Appl/Channel type  : Windows
|  Program name       : testerformq_v7_mqm_r32.exe
|  Program description: Application program
|  Process id         : 2852
|  Thread id          : 2
|  Process id (real)  : 2852
|  Thread id  (real)  : 1952
|Process parameters
|  Parameter file     : E:¥ma0w¥tracedata¥qx.def
|  Check update req   : 30
|  Show summary only  : no
|  Show perf only     : no
|  Perf intvl (time)  : 0
|  Perf intvl (count) : 0
|  Show parsed data   : yes
|  Show parsed content: yes
|  Max out line length: 100
|  Show hex char image: yes
|  Show hex char ccsid: 943 (Japanese PC - new SJIS)
|  Show data conv     : no
|  Show correl id     : yes
|  Show channel attrs : yes
|  Show max len MQPUT : (all)
|  Show max len MQGET : (all)
|  Show max len chl   : (all)
|  Show max len parsed: (all)
|  Count only selected: no
|  Target proc name   : tester*
|  Alert file (error) : yes
|  Alert file (warn)  : no
|  Alert file (info)  : no
|  Max file size      : 100,000,000
|  Compress command   : (none)
|  Compress threshold : 1,000,000
|  Create empty file  : no
|  Write stream mode  : no
|  Write cache        : yes
|  Write after call   : yes
|  Write buffer size  : 10,000
|  Write interval     : 30
|  (Debug flag)       : no
|-----------------------------------------------------------
```

GUI for parameter file manipulation

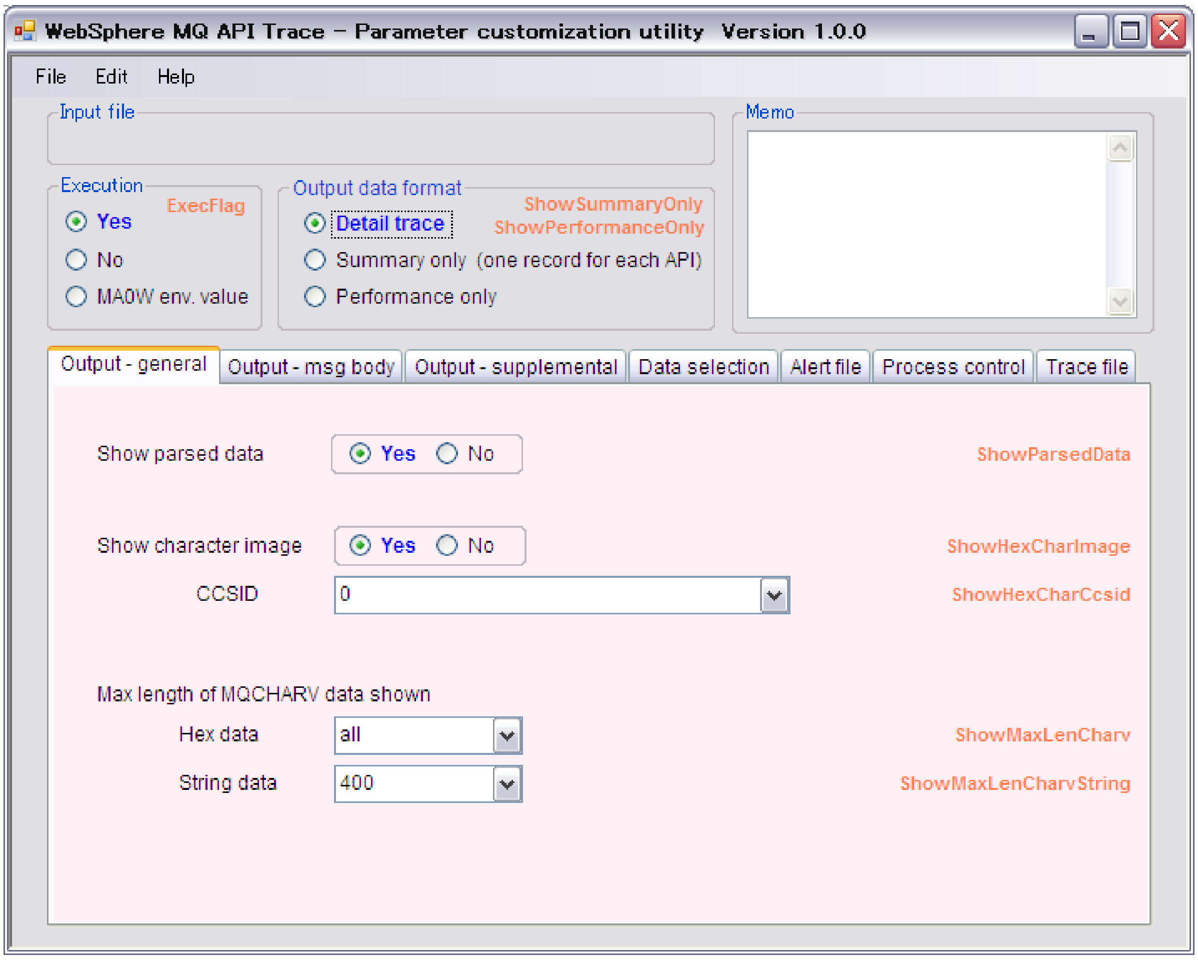
Graphical user interface for manipulating parameter file (qx.def file) is provided on Windows.
Though it runs on only Windows, generated files can also be used on UNIX operating environments.

Note: Microsoft .NET Framework is required for the execution of this user interface.

Command syntax

```
qxgui   [ parameter_file ]
```

Example of GUI window



WebSphere MQ API Trace – Parameter customization utility Version 1.0.0

File Edit Help

Input file

Memo

Execution — ExecFlag: [✓ Yes] No / MA0W env. value

Output data format — ShowSummaryOnly, ShowPerformanceOnly: [✓ Detail trace] / Summary only (one record for each API) / Performance only

Output - general | Output - msg body | Output - supplemental | Data selection | Alert file | Process control | Trace file

Show parsed data: [✓ Yes] No — ShowParsedData

Show character image: [✓ Yes] No — ShowHexCharImage

CCSID: 0 — ShowHexCharCcsid

Max length of MQCHARV data shown

Hex data: all — ShowMaxLenCharv

String data: 400 — ShowMaxLenCharvString

Chapter 6. Enabling/Disabling Trace Function

Enabling/Disabling Trace Function Dynamically

Trace function can be enabled and disabled dynamically after starting queue manager. In addition, trace options can be updated dynamically when trace function is active.

Enabling trace function

When trace function is disabled initially by specifying either of:

- ExecFlag
- TargetProcess
- ExcludeProcess

parameters in parameter file or disabled dynamically, trace function can be enabled using “qxset enable” command. Command syntax is as follows:

```
qxset  enable  trace_dir   target_process_id  [  target_thread_id ]
qxset  enable  trace_dir   all
```

Parameters:

trace_dir is directory name where trace data is to be put.

target_process_id is target process id to be enabled. If value “all” is specified as target process id, request is to be applied to all MQ processes putting trace data to the specified directory.

target_thread_id is target thread id to be enabled. It can be either actual thread id or thread id set in the context of MQ execution. If it is not specified, request is applied to all threads in the specified process.

Disabling trace function

Active trace function can be disabled using “qxset disable” command.
Command syntax is as follows:

```
qxset  disable  trace_dir   target_process_id  [  target_thread_id ]
qxset  disable  trace_dir   all
```

Parameters are same as “Enabling trace function” specified in this chapter.

Renewing trace options

Trace option can be updated using “qxset renew” command. When this function is invoked, the following functions are performed.

- Active trace file is closed.
- Parameter file is read and latest parameter values in the file are applied for further processing.

Command syntax is as follows:

```
qxset  renew  trace_dir   target_process_id  [  target_thread_id ]
qxset  renew  trace_dir   all
```

Parameters are same as “Enabling trace function” specified in this chapter.

Minimum time interval for changing trace option

To minimize performance degradation, minimum time interval to check invocation of qxset command by API exit modules can be customized by setting CheckUpdateRequestInterval parameter in parameter file. If trace option is to be changed in short time frame, value must be updated appropriately.

Refer to “CheckUpdateRequestInterval” in page 42 for more information.

How to identify target process id and thread id

When trace function is active, one of the easiest methods to identify process id and thread id is to find name of trace file. (Refer to “File name” in page 81.) They can also be found in process parameters information in trace file. (Refer to "How to check process parameters at execution time” in page 62.)

Therefore, if you want to find target process ids and thread ids to be traced in easy way, it is recommended to set CreateEmptyFile=yes to parameter file so that process profile information of MQ processes is to be put to trace file even if trace function is initially disabled by specifying either of ExecFlag=no parameter, TargetProcess parameters and ExcludeProcess parameters.

If CreateEmptyFile=no is specified in parameter file, trace file is not generated at all for the processes not matching trace condition regarding TargetProcess parameters and ExcludeProcess parameters.

If CreateEmptyFile=no is specified with ExecFlag=no, trace file is not generated.

In the cases, target process to activate trace function must be identified using general methods such as UNIX ps command, Windows task manager and so on.

Enabling and disabling trace for new processes

Methods described in this chapter are for enabling and disabling tracing for active MQ processes.

All invocations of qxset command are ignored by MQ processes started after invoking it. For the processes, initial trace options can be controlled by editing parameter file just before starting processes since parameter file is read at the startup of each MQ process.

For example, if whole trace is initially disabled by specifying ExecFlag=no parameter in parameter file and trace data just for specific MQ application program to be started soon is to be got, it can be done just by updating parameter file specifying ExecFlag=yes. Editing parameter file is no effect for the other active MQ processes unless qxset command is invoked.

Sample operation for changing trace options dynamically

Trace all MQ processes during specific time frame

(Step 1) Set parameter file

Specify "ExecFlag=no" in parameter file before starting queue manager.
If only trace files including actual trace data are to be generated, specify "CreateEmptyFile=no" in addition.

If trace option is to be changed in short time frame, it is recommended to modify CheckUpdateRequestInterval parameter to smaller value so that update request is to be checked more frequently by API exit.

(Step 2) Start queue manager

(Step 3) When trace is to be enabled, update ExecFlag value in parameter file to "ExecFlag=yes" and run qxset command to apply updated process parameter value to MQ processes.

Run "`qxset renew` *`<trace_dir>`* `all`" command.

(Step 4) When trace is to be disabled again, update ExecFlag value in parameter file to "ExecFlag=no" and run qxset command to apply updated process parameter value to MQ processes.

Run "`qxset renew` *`<trace_dir>`* `all`" command.

Note: Actual enabling and disabling process is performed at the timing of first API call in each MQ processes/threads after invoking qxset command. Therefore, if MQ API call is not performed in target processes/threads after invoking qxset command, trace status remains unchanged.

Trace particular active MQ process during specific time frame

(Step 1) Set parameter file

Specify "ExecFlag=no" in parameter file before starting queue manager.
If only trace files including actual trace data are to be generated, specify "CreateEmptyFile=no" in addition.

(Step 2) Start queue manager

(Step 3) Enable trace when it is to be enabled

Run "`qxset enable` *`<trace_dir>`* *`<target_process_id>`*" command.

(Step 4) Disable trace when it is to be disabled

Run "`qxset disable` *`<trace_dir>`* *`<target_process_id>`*" command.

Note: Actual enabling and disabling process is performed at the timing of first API call in each MQ processes/threads after invoking qxset command. Therefore, if MQ API call is not performed in target processes/threads after invoking qxset command, trace status remains unchanged.

Enabling/Disabling Trace Function for Each Process

If target processes to be traced often vary, it is recommended to exploit “ExecFlag=env” parameter setting in parameter file. (Refer to page 44 for more information.)
When this parameter is set, trace function is activated for processes running with non-null MA0W environment value.

Example in korn shell on UNIX:

```
export MA0W=y
(run MQ application)
```

Example in command prompt on Windows

```
set MA0W=y
(run MQ application)
```

If process names to be traced do not vary so often, it is useful to exploit TargetProcess parameter in parameter file. (Refer to page 58 for more information.)

Chapter 7. Utilities

Reformat API trace utility (common to API exit and Channel exit)

Showing parsed attributes with hexadecimal data representation makes viewing easier in looking into trace data. Character representation with hexadecimal data also makes viewing easier.
ShowParsedData, ShowParsedMsgContent and ShowHexCharImage parameter in the definition file (qx.def file) provide processing options for the requirements.

Some users might want to run MQ application programs with the API trace utility disabling them intentionally regarding performance degradation. (Refer to “Chapter 8. Performance” in page 72 for more information.)
In the case, user may want to look into trace data with parsed information and/or character data information in addition to original data generated by the API trace utility after executing MQ application. Reformat API trace utility is provided to add parsed data and character representation after getting trace data file.

Command syntax

32-bit module

```
qxreform32 input_trace_file  [ -c ccsid ] [ -x length ] [ -o output_file ] [ -nc ]
                                 [ -v show_max_len_charv_string ]
```

64-bit module

```
qxreform64 input_trace_file  [ -c ccsid ] [ -x length ] [ -o output_file ] [ -nc ]
                                 [ -v show_max_len_charv_string ]
```

Note that program qxreform32 should be used for 32-bit trace file. For 64-bit trace file, program qxreform64 should be used.

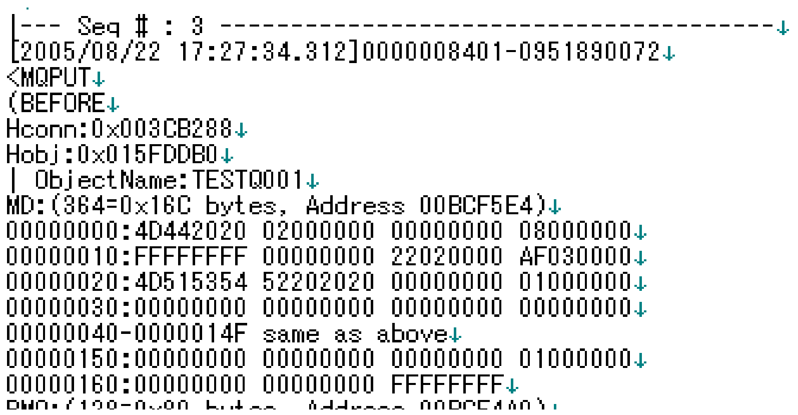
Parameters

input_trace_file	Input trace file to be reformatted
-c *ccsid*	CCSID value referred for formatting. For the supported CCSID values and default values (automatic setting), refer to “ShowHexCharCcsid” in page 47
-x *length*	Maximum line length in formatting XML data. It is equivalent to MaxParsedOutputLineLength parameter specified in page 44.
-v *length*	Maximum length of variable length data (MQCHARV data) show in string format. It is equivalent to ShowMaxLenCharvString parameter specified in page 50.
-o *output_file*	Name of output file.
-nc	Message content is not parsed and formatted.

Trace file which already includes character data and/or parsed attribute data can be specified as input file of the utility. It is useful when user set ShowHexCharCcsid value and/or ShowParsedMsgContent value in parameter file improperly in getting trace data, for example.

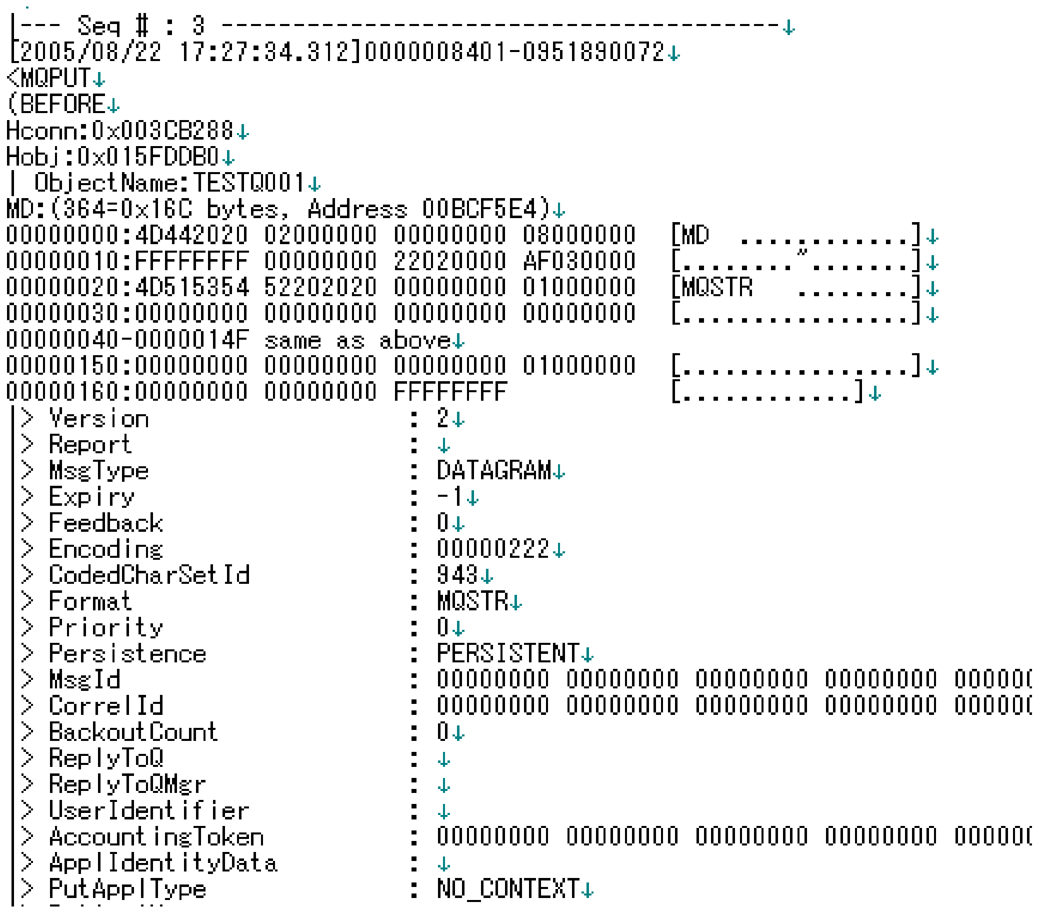
Sample input/output

Sample input file:



```
|--- Seq # : 3 ---------------------------------------↓
[2005/08/22 17:27:34.312]0000008401-0951890072↓
<MQPUT↓
(BEFORE↓
Hconn:0x003CB288↓
Hobj:0x015FDDB0↓
| ObjectName:TESTQ001↓
MD:(364=0x16C bytes, Address 00BCF5E4)↓
00000000:4D442020 02000000 00000000 08000000↓
00000010:FFFFFFFF 00000000 22020000 AF030000↓
00000020:4D515354 52202020 00000000 01000000↓
00000030:00000000 00000000 00000000 00000000↓
00000040-0000014F same as above↓
00000150:00000000 00000000 00000000 01000000↓
00000160:00000000 00000000 FFFFFFFF↓
```

Sample output file:



```
|--- Seq # : 3 ---------------------------------------↓
[2005/08/22 17:27:34.312]0000008401-0951890072↓
<MQPUT↓
(BEFORE↓
Hconn:0x003CB288↓
Hobj:0x015FDDB0↓
| ObjectName:TESTQ001↓
MD:(364=0x16C bytes, Address 00BCF5E4)↓
00000000:4D442020 02000000 00000000 08000000  [MD  ............]↓
00000010:FFFFFFFF 00000000 22020000 AF030000  [........".......]↓
00000020:4D515354 52202020 00000000 01000000  [MQSTR   ........]↓
00000030:00000000 00000000 00000000 00000000  [................]↓
00000040-0000014F same as above↓
00000150:00000000 00000000 00000000 01000000  [................]↓
00000160:00000000 00000000 FFFFFFFF           [............]↓
|> Version                    : 2↓
|> Report                     : ↓
|> MsgType                    : DATAGRAM↓
|> Expiry                     : -1↓
|> Feedback                   : 0↓
|> Encoding                   : 00000222↓
|> CodedCharSetId             : 943↓
|> Format                     : MQSTR↓
|> Priority                   : 0↓
|> Persistence                : PERSISTENT↓
|> MsgId                      : 00000000 00000000 00000000 00000000 000000
|> CorrelId                   : 00000000 00000000 00000000 00000000 000000
|> BackoutCount               : 0↓
|> ReplyToQ                   : ↓
|> ReplyToQMgr                : ↓
|> UserIdentifier             : ↓
|> AccountingToken            : 00000000 00000000 00000000 00000000 000000
|> ApplIdentityData           : ↓
|> PutApplType                : NO_CONTEXT↓
```

Summary data generation utility (for API exit)

Utility program to generate summary of API call sequence information from detail API call data file is provided. It can be used for finding MQ API call sequence at a glance. Output data format of the utility is similar to output of API exit modules when “ShowSummaryOnly=Yes” is specified in parameter file. (Refer to “(API Exit) with ShowSummaryOnly=yes parameter” in page 98.)

The utility can also be used to merge trace data in multiple trace files into a summary data. Both summary trace files (trace data with ShowSummaryOnly=Yes parameter) and detail trace files (trace data with ShowSummaryOnly=no parameter) can be mixed in a file list. It is useful to check usage of a queue manager by multiple MQ applications in time sequence.

Usage information of each queue is reported by the utility in addition. Refer to sample output in the section “(API Exit) Sample output by Summary Data Generation Utility” in page 107.

Command syntax

```
qxsumapi  input_file  [ -o output_file ]
```

Parameters

input_file	One of the followings: ● A trace file generated by API exit module ● List of trace files (see below) When first byte of input file is “|”, input file is assumed as trace file generated by API exit module. Otherwise, it is assumed as list of trace files.
-o *output_file*	Output file. Default value is <input_file_without_extension>_sumutil.txt . For example, if name of input file is “trace_file_list.txt”, output file name becomes “trace_file_list_sumutil.txt”.

When trace files are listed in input file, specify only one file name in each line.
If first nonblank character in a line is asterisk (‘*’) or pound (‘#’), it is ignored. Blank lines are also ignored.

Example of input file:

```
qx_QMGR001_testpgm.exe_1216_1_001.txt
qx_QMGR001_testpgm.exe_1216_1_002.txt
# qx_QMGR001_testpgm.exe_1216_1_003.txt  -- to be ignored --
qx_QMGR001_utilitypgm.exe_2345_1_001.txt
```

Summary data generation utility (for channel exit)

Utility program to generate summary of transferred data via MQ channels from detail channel exit data file is provided. Output data format of the utility is similar to output of API exit modules when “ShowSummaryOnly=Yes” is specified in parameter file.

Command syntax

```
qxsumchl  input_file  [ -o output_file ]
```

Parameters

input_file	A trace file generated by channel exit module
output_file	Output file. Default value is <input_file_without_extension>_sumutil.txt . For example, if name of input file is “trace_file_list.txt”, output file name becomes “trace_file_list_sumutil.txt”.

Message browsing utility

The utility is now provided as SupportPac MA96.
Visit the following URL for more information.

```
http://www-01.ibm.com/support/docview.wss?rs=171&uid=swg24021783&loc=en_US&cs=utf-8&lang=en
```

Chapter 8. Performance

In developing API trace utility, effort to minimize performance degradation has been made. However, performance degradation with more system resource consumption is inevitable when MQ API exit interface is used. In the chapter, actual performance data is shown as examples. Note that performance data varies depending on processing environment. All data in the chapter is just examples.

Performance data regarding contents of trace data

Performance varies significantly depending on trace options. Refer to the following data regarding API trace options.

Test environment

● CPU	: pSeries 690 (7040-681) Power 4, 1.3GHz x 8 way
● Memory	: 16.0GB
● OS	: AIX 5.3 + ML 03
● MQ version	: V6.0.1
● MQ message	: Persistent, 200 bytes, text string data
● API call sequence	: MQCONNX - MQOPEN - ((MQPUT 10 times - MQCMIT) * 10,000 times) - MQCLOSE - MQDISC

Performance data

Process performance depends mainly on data volume. The following parameters are major factor regarding output data volume.

● ShowPerformanceOnly	: Summary of performance data only
● ShowSummaryOnly	: One simple output record or all parameter data for each API call
● ShowParsedData	: Whether attributes are put in parsed format in addition to hexadecimal representation
● ShowParsedMsgContet	: Whether message content is to be shown in parsed format or not
● ShowHexCharImage	: Whether character image is to be put to trace data in addition to hexadecimal representation

Set parameters regarding your objectives in getting trace data.

Process parameter setting	Elapsed Time (second)	Data volume (MB)	Note
No API exit definition	84	0	
With API Exit definition ShowPerformanceOnly=yes	85	0.004	(Example: page 97)

With API Exit definition
ShowPerformanceOnly=yes
ShowPerformanceIntervalTime=5
	85	0.059	

With API Exit definition ShowSummaryOnly=yes	86	20	Suitable method for performance measurement (Example: page 98)

With API Exit definition
ShowParsedData=no
ShowParsedMsgContent=no
ShowHexCharImage=no
	91	330	Suitable method for performance sensitive problem determination
			(See *1 below.)

With API Exit definition
ShowParsedData=no
ShowParsedMsgContent=no
	113	443	

With API Exit definition ShowHexCharImage=no	121	692	
With API Exit definition (default)	124	804	Fully formatted output. (Example: page 100)

(*1) : If full trace data is to be got with minimum performance degradation at execution time, it is recommended to run with ShowParsedData=no and ShowHexCharImage=no options and use Reformat API trace utility specified in page 68 after getting trace data.

Performance data regarding options for file I/O

The following parameters are major factors regarding options for file I/O.

● WriteStream	:	Whether trace data is to be put in I/O stream mode or low level I/O mode.
● WriteCache	:	Whether write cache is enabled or disabled.
● WriteAfterEachApi	:	Whether trace data is to be put to file right after processing each API.
● WriteBufferSize	:	Size of buffer where trace data is cached before putting to file.

Consider to modify parameter values if you want to use the utility for performance measurement or problem determination for performance sensitive problems.

Process parameter setting	**Elapsed Time (second)**	**Remark**
No API exit definition	84	

ShowSummaryOnly=no		
With API Exit definition WriteStream=yes	122	Write delay might occur.
With API Exit definition WriteCache=no	282	Many I/O requests. Not recommended.

With API Exit definition
WriteAfterEachApi=no
WriteBufferSize=10000 (10KB – default)
	122	Write delay might occur.

With API Exit definition WriteBufferSize=10000 (10KB – default)	124	Default setting

With API Exit definition
WriteAfterEachApi=no
WriteBufferSize=500000 (500KB)
	123	Write delay might occur.

With API Exit definition WriteBufferSize=500000 (500KB)	123	

ShowSummaryOnly=yes		
With API Exit definition WriteStream=yes	85	Write delay might occur.
With API Exit definition WriteCache=no	86	Many I/O requests. Not recommended.

With API Exit definition
WriteAfterEachApi=no
WriteBufferSize=10000 (10KB - default)
	85	Write delay might occur.

With API Exit definition WriteBufferSize=10000 (10KB - default)	86	

Memory usage by API exit

Memory usage was optimized after shipping V4.0 of API trace utility so that MQ API calls can be traced with much less overhead in heavy transaction environment. Memory usage in each MQ thread has been decreased by more than 600KB.

Method used to identify memory usage

Memory usage information

- Memory usage shown in Windows task manager on Windows
- SZ value of “ps –elf” command on UNIX

Calculation of memory usage by API trace
(memory size with API exit definition) – (memory size without API exit definition)

Sample memory usage on each operating environment

The following chart shows memory usage of API exit in a process. Unit is Kilobytes (KB).
In multi thread application, MQCONNX and subsequent MQ API calls were performed concurrently in threads of a process.

For example, only 4.7MB is used by API exit even if MQ API calls are performed concurrently in 50 threads of a process on AIX.

Operating environment	**Single thread MQ application (KB)**	**Multi thread MQ application (KB)**					
Thread count	**1**	**1**	**2**	**3**	**5**	**10**	**50**
Windows	412	412	526	624	832	1,348	5,576
AIX	148	144	240	340	520	1,016	4,748
Solaris (SPARC)	384	392	472	568	744	1,192	4,776
Solaris (x86-64)	452	476	568	660	844	1,376	5,056
HP-UX (PA-RISC)	48	64	128	192	320	640	2,240
HP-UX (IA-64)	36	80	144	208	336	624	2,336
Linux (x86-64)	368	376	376	516	656	804	3,304
Linux (PowerPC)	512	512	512	668	804	952	3,472

Chapter 9. Problem Determination

Error at the startup of MQ queue manager and applications

If error occurs in queue manager startup or in application startup, check MQ log records.

On UNIX systems, MQ log files are in /var/mqm/qmgrs/<*qmgr_name*>/errors directory. File contents in /var/mqm/errors directory is also to be checked.
On Windows, log records can be checked using Windows event viewer.

If you find error message “AMQ7214: The module for API Exit 'qxEntry' could not be loaded.”, it might be access permission error. To solve the problem:

- Check access permission of API trace utility modules.
- Check access permission of directory where modules reside.
- Check access permission of parent directories of module directory.
- Check ownership of API trace utility modules.
- Check symbolic link to API trace utility modules.

Refer to “Step 3. Define API exit (queue manager side)
Define API exit modules (AIX, Solaris, HP-UX and Linux)” for more information.

Error at the startup of MQ channel

If error occurs in channel startup, check MQ log records.
On UNIX systems, MQ log files are in /var/mqm/qmgrs/<*qmgr_name*>/errors directory. File contents in /var/mqm/errors directory is also to be checked.
On Windows, log records can be checked using Windows event viewer.

If you find error message indicating dynamic load error such as AMQ6175 in MQ log file, check the followings:

- Check access permission of channel trace utility modules.
- Check access permission of directory where modules reside.
- Check access permission of parent directories of module directory.
- Check ownership of channel trace utility modules.
- Check suffix of module names defined for channel exits. (“32_r”, “64_r”, “32”, “64”) It must be defined regarding addressing mode and threading condition.
- Check uppercase and lowercase character in exit definition. If exit is defined using MQSC commands without enclosing by single quote, all characters are defined in uppercase. It causes channel startup error.

Refer to “Define channel exit modules (AIX, HP-UX, Linux)”, “Define channel exit modules (Solaris)” and “Define channel exit modules (Windows)” for more information.

Error after MQ queue manager startup

If error occurs in tracing, error messages are to be put to the following files. Find messages in the files for problem determination.

- qx_apiexit_error_<pid>_<tid>.txt (API exit)
- qx_chlexit_error_<pid>_<tid>.txt (Channel exit)

Directory where files reside depends on process timing and/or processing directory of each process.

On UNIX systems, it might be in one of the following directories:

- Directory where trace files reside.
- /var/mqm directory
- Directory in which MQ application programs are executed.

On Windows system, it might be in one of the following directories:

- Directory where trace files reside.
- Installation directory of WebSphere MQ package.
  (C:¥Program files¥IBM¥WebSphere MQ)
- “bin” subdirectory of installation directory of WebSphere MQ package.
  (C:¥Program files¥IBM¥WebSphere MQ¥bin)
- Directory in which MQ application programs are executed.
- Windows directory
  (C:¥Windows¥System32)

Error in parsing and formatting MQ messages

Error might occur in parsing and formatting various MQ messages. If error occurs unexpectedly in API exit or channel exit, reason code 2374 (MQRC_API_EXIT_ERROR) is returned to MQ application putting error messages such as AMQ7216 to MQ log in /var/mqm/qmgrs/<*qmgr_name*>/errors or /var/mqm/errors directory on Unix and event log on Windows.

If error occurs in parsing and formatting MQ messages, it can be avoided by specifying the following parameters in parameter file (qx.def file).

- ShowParsedData = no
- ShowParsedMsgContent = no

If error still persists, specification of “WriteCache = no” parameter might be helpful to identify the location where error occurred. (This parameter specification makes API exit slowdown. Do not specify it under normal condition.)

If error does not occur by specifying these parameters, please try to run reformat API trace utility (specified in page 68) for the trace files generated by API exit. The same error occurred in API exit would occur again.

I appreciate if you could contact to the address specified in page 77 with trace files when you find error.

Chapter 10. Support

License agreement

The API trace utility is provided as-is basis. You have to use the utility at your own responsibility.
Positioning of the utility is equivalent to that of WebSphere MQ category 2 SupportPacs.

Refer to http://www-306.ibm.com/software/integration/support/supportpacs/cat2license.pdf and files in license directory of installation package for the detail of license agreement.

Your feedback comments to me

Your feedback comments such as proposal for functional enhancement, usability improvement, problem reporting and so on are highly appreciated. Please do not hesitate to send them.

But please understand that formal position of the utility is as described above. Response to your comments is to be sent depending on the author’s availability without any commitment.

Please send your comments to the following e-mail address either using ma0w_your_comment_to_me.txt file in the package or free form.

e-mail :	oinoue@jp.ibm.com
IBM internal :	Osamu 2 Inoue/Japan/IBM

Support for new API attributes

New attributes might be introduced by WebSphere MQ development in future. Some existing attributes might not be formatted correctly since it is not realistic to verify “all” MQ attributes before shipping the SupportPac.

If you find missing attribute in formatting parsed data and/or incorrect attribute display, please send information to the mail address specified above.

If unknown attributes by the utility were detected, the following message is shown after summary information in trace file.

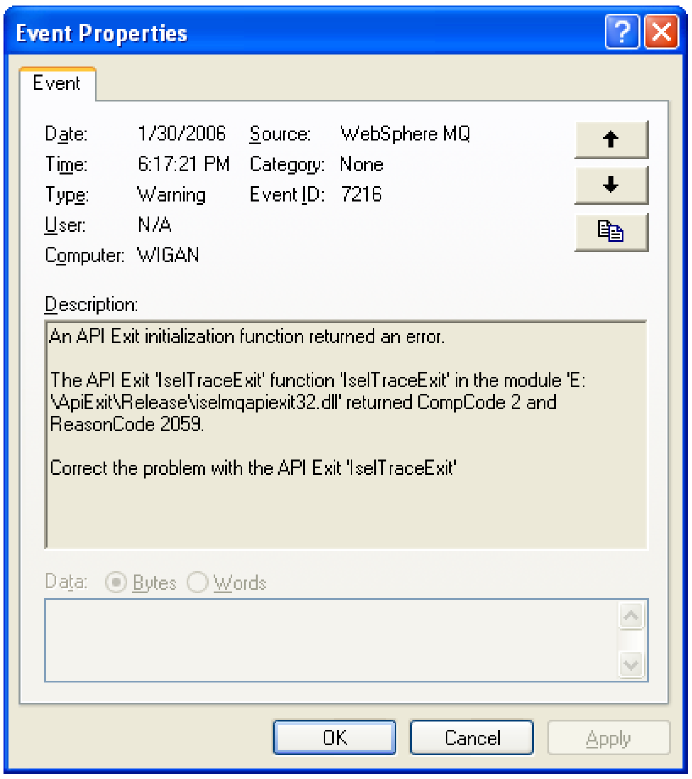
```
|*****************************************************************
|**
|**   Unknown MQ attribute exists.
|**   Please check MQ attribute settings in your program or
|**   report to oinoue@jp.ibm.com with this trace data.
|**   (New attributes might have been supported by WebSphere MQ.)
|**
|*****************************************************************
```

Known problems

Warning in Windows event log

When MQ queue manager is terminated using MQ Explorer, the following warning message indicating CompCode 2 and ReasonCode 2059 (MQRC_Q_MGR_NOT_AVAILABLE) or 2161 (MQRC_Q_MGR_QUIESCING) is shown in Windows event viewer. (There is no functional impact except for monitoring application events.)

This event log does not appear when MQ queue manger is terminated using endmqm command.



Date:	1/30/2006	Source:	WebSphere MQ
Time:	6:17:21 PM	Category:	None
Type:	Warning	Event ID:	7216
User:	N/A		
Computer:	WIGAN		

Description:

An API Exit initialization function returned an error.

The API Exit 'IselTraceExit' function 'IselTraceExit' in the module 'E:\ApiExit\Release\iselmqapiexit32.dll' returned CompCode 2 and ReasonCode 2059.

Correct the problem with the API Exit 'IselTraceExit'

Data: (● Bytes) ○ Words

API exit error when invalid pointer value is passed by MQ application program

If invalid pointer value is set as MQ API parameter by MQ application programs, error is reported as API exit error by MQ though cause of error is in MQ application program. It is because API exit simply tries to access memory area passed by application without error handing routine and it causes memory access error (SIGSEGV on UNIX, Access violation (GPF) on Windows).

The following information is reported by MQ.

Common error directory (/var/mqm/errors on UNIX, <*install_dir* >¥errors on Windows)

- Log file

  Example of AMQERRnn.LOG file:

```
[07/09/26 04:39:40] - Process(237956.1) User(mqm) Program(mqputerror)
AMQ6109: An internal WebSphere MQ error has occurred.

EXPLANATION:
An error has been detected, and the WebSphere MQ error recording routine has
been called.
ACTION:
Use the standard facilities supplied with your system to record the problem
identifier, and to save the generated output files. Contact your IBM support
center.  Do not discard these files until the problem has been resolved.
----- amqxfdcx.c : 771 --------------------------------------------------------


[07/09/26 04:39:40] - Process(237956.1) User(mqm) Program(mqputerror)
AMQ6184: An internal WebSphere MQ error has occurred on queue manager QM_MAOW.

EXPLANATION:
An error has been detected, and the WebSphere MQ error recording routine has
been called. The failing process is process 237956.
ACTION:
Use the standard facilities supplied with your system to record the problem
identifier, and to save the generated output files. Contact your IBM support
center.  Do not discard these files until the problem has been resolved.
----- amqxfdcx.c : 810 --------------------------------------------------------
```

- FDC file

  FDC file is generated by MQ in addition to log file.

Error directory for each queue manager (/var/mqm/qmgrs/<*qmgr_name* >/errors on UNIX, <*install_dir*>¥<*qmgr_name* >¥errors on Windows)

- Log file

  Example of AMQERRnn.LOG file:

```
07/09/26 13:39:40 - Process(237956.1) User(mqm) Program(mqputerror)
AMQ7216: An API Exit initialization function returned an error.

EXPLANATION:
The API Exit 'qxEntry' function 'qxEntry' in the module'qx1api' returned CompCode 2 and ReasonCode 2374.

ACTION:
Correct the problem with the API Exit 'qxEntry'
```

In the case, error in MQ application program must be fixed so that valid pointer value is passed to MQ API calls. Find cause of error in MQ application program by checking trace files generated by API exit.

The following is an example when invalid MQPUT buffer pointer was set and passed by MQ application program.

```
|--- Seq # : 3 ---------------------------------------
[2007/09/26 13:39:40.304219]
<MQPUT
(BEFORE
Hconn:0x200424F8
Hobj:0x00000002
| ObjectName:Q1
MD:(324=0x144 bytes, Address 0x2FF229A0)
00000000:4D442020 00000001 00000000 00000008  [MD  ............]
00000010:FFFFFFFF 00000000 00000111 00000000  [................]
00000020:20202020 20202020 FFFFFFFF 00000002  [        ........]
00000030:00000000 00000000 00000000 00000000  [................]
00000040-0000013F same as above
00000140:00000000                             [....]
|> Version                    : 1
|> Report                     : (none)
|> MsgType                    : DATAGRAM
|> PutTime                    :
|> ApplOriginData             :

      . . .

PMO:(128=0x80 bytes, Address 0x2FF22B10)
00000000:504D4F20 00000001 00000000 FFFFFFFF  [PMO ............]
00000010:00000000 00000000 00000000 00000000  [................]
00000020-0000006F same as above
00000070:00000000 00000000 00000000 00000000  [................]
|> Version                    : 1
|> Options                    : (none)
|> Timeout                    : -1

      . . .

|> ResolvedQMgrName           :
BufferLength:13
Buffer:(13=0xD bytes, Address 0x3AD64F00)       <== Expected hexadecimal value is not present.
-                                               <== It can be guessed that passed buffer address
[2007/09/26 13:39:40.397946]                        0x3AD64F00 might be incorrect.
<MQPUT
(AFTER
CompCode:2                                      <==
Reason:2374                                     <== Reported as API exit error, unfortunately.
| Reason:MQRC_API_EXIT_ERROR                    <==
| API process time:    0.093727 second
    . . .
```

Appendix A. Specification of Output Trace file

File specification of trace file to be generated is specified in this section.

File specification of trace file

Directory

Value specified with Data parameter of API exit definition.

Refer to "Step 2. Prepare directory for trace data" in page 25 and "Step 3. Define API exit (queue manager side) Define API exit modules (AIX, Solaris, HP-UX and Linux)" in page 25 for more information.

File name

File name of trace files are to be set with the following naming convention.

API exit: when ShowSummaryOnly=yes and ShowSummaryOnly=yes options are not specified:
    qx_*<qmgr_name>*_*<program_name>*_*<process_id>*_*<thread_id>*_*<seq_number>*

API exit: when ShowSummaryOnly=yes option is specified:
    qx_*<qmgr_name>*_*<program_name>*_*<process_id>*_*<thread_id>*_*<seq_number>*_summary

API exit: when ShowPerformanceOnly=yes option is specified:
    qx_*<qmgr_name>*_*<program_name>*_*<process_id>*_*<thread_id>*_*<seq_number>*_perf

Channel exit: when ShowSummaryOnly=yes option and ShowSummaryOnly=yes options are not specified:
    qx_*<qmgr_name>*_channel_*<chl_type>*_*<chl_name>*_*<process_id>*_*<thread_id>*_*<seq_number>*

Channel exit: when ShowSummaryOnly=yes option is specified:
    qx_*<qmgr_name>*_channel_*<chl_type>*_*<chl_name>*_*<process_id>*_*<thread_id>*_*<seq_number>*_summary

Channel exit: when ShowPerformanceOnly=yes option is specified:
    qx_*<qmgr_name>*_channel_*<chl_type>*_*<chl_name>*_*<process_id>*_*<thread_id>*_*<seq_number>*_perf

File extension

File extension varies depending on the processing status

File extension	Processing status
.t	Now opened and writing trace data
.txt	Write complete and closed.
.txt.txt[.*xxx*]	Write complete, closed and compressed. File suffix is set by compression program.
.txt.f	Trace is disabled or reset dynamically by qxset command. (Refer to "Chapter 6. Enabling/Disabling Trace Function" in page 64 for more information about qxset command.)
.txt.f.txt [.*xxx*]	Trace is disabled or reset dynamically by qxset command and then compressed. File suffix is set by compression program. (Refer to "Chapter 6. Enabling/Disabling Trace Function" in page 64 for more information about qxset command.)

File specification of alert file

Directory

Value specified with Data parameter of API exit definition.

Refer to “Step 2. Prepare directory for trace data” in page 25 and “Step 3. Define API exit (queue manager side) Define API exit modules (AIX, Solaris, HP-UX and Linux)” in page 25 for more information.

File name

File name of trace files are to be set with the following naming convention.

```
alert<qmgr_name>_<program_name>_<process_id>_<thread_id>_<seq_number>.txt
```

Description of call summary data in trace files

Categories of API call results

Results of API calls are categorized into the following four categories.

Results	API CompCode	API Reason Code
Success	0	(any)
Info	(any)	2033 (MQRC_NO_MSG_AVAILABLE) 2059 (MQRC_Q_MGR_NOT_AVAILABLE) 2107 (MQRC_XWAIT_CANCELED) 2161 (MQRC_Q_MGR_QUIESCING) 2437 (MQRC_NO_RETAINED_MSG) 2471 (MQRC_PROPERTY_NOT_AVAILABLE)
Warning	1	(any)
Fail	2	(any code except for “info” result)

Description of process time

Average process time, minimum process time and maximum process time for each API call type are shown in call summary data. Unit is in second.

Example

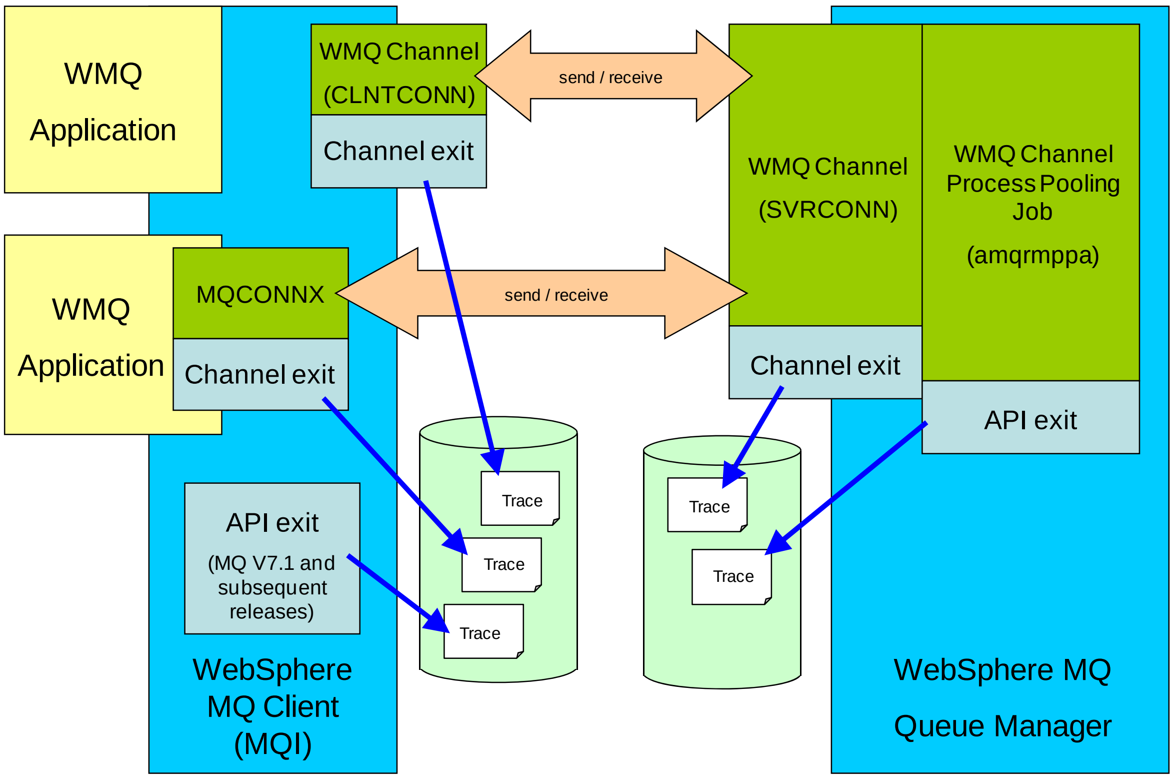
```
------------------------------------------------------------------------------------------
Call summary   [2005/10/24 23:57:12.669663]
                                                            -------- Process time (second) -------
  API       Success     Info  Warning     Fail    Total     Average      Minimum      Maximum
  ------- -------- -------- -------- -------- --------  ------------ ------------ ------------
  MQCONN         1        0        0        0        1     0.003140     0.003140     0.003140
  MQCONNX        0        0        0        0        0
  MQDISC         1        0        0        0        1     0.000352     0.000352     0.000352
  MQOPEN         1        0        0        0        1     0.000969     0.000969     0.000969
  MQCLOSE        1        0        0        0        1     0.000315     0.000315     0.000315
  MQPUT1         0        0        0        0        0
  MQPUT          3        0        0        0        3     0.000778     0.000595     0.000919
  MQGET          0        0        0        0        0
  MQINQ          0        0        0        0        0
  MQSET          0        0        0        0        0
  MQBEGIN        0        0        0        0        0
  MQCMIT         0        0        0        0        0
  MQBACK         0        0        0        0        0
  ------- -------- -------- -------- -------- --------  ------------ ------------ ------------
  Total          7        0        0        0        7
------------------------------------------------------------------------------------------
```

Appendix B. Hints and Tips

Getting trace data for MQ client program

When MQ application program calls MQ application programming interface (API) locally, API calls can be traced simply using API exit interface. On the other hand, when MQ application programs call MQ API remotely via MQ channel, more consideration is needed since API exit interface is not provided at MQ client side if installed MQ version is 7.0 or older.

Methods for getting trace data of API calls invoked by MQ client programs remotely are introduced in this section. The following chart depicts summary for getting trace for MQ client programs.



Refer to “Summary of methods for getting WebSphere MQ API trace” in page 18 for the comparison of methods.

Getting trace using API exit interface (MQ server side)

API calls invoked by MQ client program via MQ channels are managed by process named “amqrmppa” (channel process pooling job) at MQ queue manager (MQ server) side. Therefore API calls performed by MQ client can be identified by finding API trace for process “amqrmppa”.

If multiple client programs and/or MQ channels are working at the same time, multiple amqrmppa processes are generated by MQ queue manager. In the case, target trace file must be identified regarding process timing, API call sequence, message content and so on.

Note that MQINQ calls performed by amqrmppa process are included in trace data in addition to API calls invoked by MQ client program.

Getting trace using channel exit interface (MQ server side)

API calls invoked by MQ client program via MQ channels can be traced by setting channel trace to server connection (SVRCONN) channel.

Getting trace using API exit interface (MQ client side)

API exit interface at client side is newly provided by WebSphere MQ V7.1. Exit interface can be defined in mqclient.ini file.

Refer to “Step 4. Define API exit (client side)” in page 30 for more information.

Getting trace using channel exit interface (MQ client side)

API calls invoked by MQ client program via MQ channels can be traced by setting channel trace to client connection (CLNTCONN) channel.

Refer to “WebSphere MQ Client” document for more information about client connection channel.

Getting trace by modifying client program (MQ client side)

If client program invokes MQCONNX call to connect to queue manager, channel exit can be set in MQCONNX parameter.

The following is an example of setting channel exit in C program.

```
MQCNO     cno = {MQCNO_DEFAULT};
MQCD      cd  = {MQCD_CLIENT_CONN_DEFAULT};
char      exitname[MQ_EXIT_NAME_LENGTH];
char      exitdata[MQ_EXIT_DATA_LENGTH];

    . . .
    . . .

cno.Version = MQCNO_VERSION_2;  /* or higher */
cno.ClientConnPtr = &cd;

if (  _exit_is_to_be_enabled_  )
{
   strncpy(exitname,  _somewhere_  ,  MQ_EXIT_NAME_LENGTH);
   strncpy(exitdata,  _somewhere_  ,  MQ_EXIT_DATA_LENGTH);

   cd.ExitNameLength       = MQ_EXIT_NAME_LENGTH;
   cd.ExitDataLength       = MQ_EXIT_DATA_LENGTH;
   cd.SendExitsDefined     = 1;
   cd.SendExitPtr          = exitname;
   cd.SendUserDataPtr      = exitdata;
   cd.ReceiveExitsDefined = 1;
   cd.ReceiveExitPtr       = exitname;
   cd.ReceiveUserDataPtr   = exitdata;
}

MQCONNX(qmgr_name,  &cno,  &hconn,  &CompCode,  &Reason);
```

Value to be set in variable “exitname” in sample program above is equivalent to that for SENDEXIT/RCVEXIT parameters in MQSC DEFINE/ALTER CHANNEL command.

Value to be set in variable “exitdata” in sample program above is equivalent to that for SENDDATA/RCVDATA parameters in MQSC DEFINE/ALTER CHANNEL command.

Refer to channel exit definition parts of “Chapter 3. Installation” for more information about value settings to “exitname” and “exitdata”.

Performance measurement

Performance of MQ API call executions can be measured using the utility. It is recommended to set ShowPerformanceOnly=yes or ShowSummaryOnly=yes in parameter file (qx.def file) for minimizing performance degradation.

Average, minimum and maximum process time for each type of MQ API

Average process time, minimum process time and maximum process time for each API call type are shown in summary data at the bottom of trace data.

```
-------------------------------------------------------------------------------------------
Call summary   [2005/10/24 23:57:12.669663]
                                                       -------- Process time (second) -------
  API       Success     Info  Warning     Fail    Total     Average      Minimum      Maximum
  ------- -------- -------- -------- -------- -------- ------------ ------------ ------------
  MQCONN         1        0        0        0        1     0.003140     0.003140     0.003140
  MQCONNX        0        0        0        0        0
  MQDISC         1        0        0        0        1     0.000352     0.000352     0.000352
  MQOPEN         1        0        0        0        1     0.000969     0.000969     0.000969
  MQCLOSE        1        0        0        0        1     0.000315     0.000315     0.000315
  MQPUT1         0        0        0        0        0
  MQPUT          3        0        0        0        3     0.000778     0.000595     0.000919
  MQGET          0        0        0        0        0
  MQINQ          0        0        0        0        0
  MQSET          0        0        0        0        0
  MQBEGIN        0        0        0        0        0
  MQCMIT         0        0        0        0        0
  MQBACK         0        0        0        0        0
  ------- -------- -------- -------- -------- -------- ------------ ------------ ------------
  Total          7        0        0        0        7
-------------------------------------------------------------------------------------------
```

Process time for each MQ API call

When ShowSummaryOnly=yes and ShowPerformanceOnly=no parameters are specified, process time for each MQ API call can be put to trace file with minimum performance degradation. (Average, minimum and maximum process time for each type of MQ API call specified above are also included.)

```
   Time                         API      CC Reason  Time(sec)  ObjectName                    MsgId          MsgLen
 1 [2005/11/03 20:19:07.421553] MQCONNX 0      0   0.188269
 2 [2005/11/03 20:19:07.425598] MQOPEN  0      0   0.003825
 3 [2005/11/03 20:19:07.427455] MQINQ   0      0   0.001771
 4 [2005/11/03 20:19:07.428277] MQOPEN  0      0   0.000747  SYSTEM.DEFAULT.NAMELIST
 5 [2005/11/03 20:19:07.428739] MQCLOSE 0      0   0.000388  SYSTEM.DEFAULT.NAMELIST
 6 [2005/11/03 20:19:07.429523] MQOPEN  0      0   0.000715  SYSTEM.CLUSTER.COMMAND.QUEUE
 7 [2005/11/03 20:19:07.429844] MQINQ   0      0   0.000255  SYSTEM.CLUSTER.COMMAND.QUEUE
 8 [2005/11/03 20:19:07.438300] MQOPEN  0      0   0.008398  SYSTEM.CLUSTER.TRANSMIT.QUEUE
 9 [2005/11/03 20:19:07.438843] MQOPEN  0      0   0.000306  SYSTEM.CLUSTER.REPOSITORY.QUEUE
10 [2005/11/03 20:19:07.439481] MQGET   0      0   0.000545  SYSTEM.CLUSTER.REPOSITORY.QUEUE  414D512051...   12320
11 [2005/11/03 20:19:07.439985] MQOPEN  0      0   0.000178  SYSTEM.CLUSTER.TRANSMIT.QUEUE
12 [2005/11/03 20:19:07.440243] MQGET   2 2033   0.000141  SYSTEM.CLUSTER.TRANSMIT.QUEUE     (2033 = MQRC_NO_MSG_AVAILABLE)
13 [2005/11/03 20:19:07.440480] MQCLOSE 0      0   0.000164  SYSTEM.CLUSTER.TRANSMIT.QUEUE
14 [2005/11/03 20:19:07.440676] MQINQ   0      0   0.000127
15 [2005/11/03 20:19:07.441010] MQGET   2 2033   0.000272  SYSTEM.CLUSTER.COMMAND.QUEUE      (2033 = MQRC_NO_MSG_AVAILABLE)
```

Checking usage and performance of each queue

Usage of each queue can be checked using summary data generation utility specified in page 70. API call count, result and process time are summarized for each queue. For MQPUT, MQPUT1 and MQGET, message length data is also reported.

Sample output of MQGET call and MQINQ call:

```
| MQGET
|                                                              --- Process time (second) --- - Message length (byte) -
| Queue                         Success   Info  Warning  Fail  Total  Average   Minimum   Maximum   Average Minimum Maximum
| ------------------------------ ------ ------ ------ ------ ------ --------- --------- --------- ------- ------ ------
| (null)                              0      0      0      0      0
| SYSTEM.ADMIN.COMMAND.QUEUE          0      2      0      0      2 11.245175  8.568275 13.922074
| SYSTEM.ADMIN.STATISTICS.QUEUE       0      0      0      0      0
| SYSTEM.AUTH.DATA.QUEUE            112      2      0      0    114  0.000188  0.000017  0.000609     185      4    188
| SYSTEM.CHANNEL.INITQ                0      2      0      0      2 11.190440  8.542740 13.838139
| SYSTEM.CLUSTER.COMMAND.QUEUE        2      2      0      0      4 10.757440  0.000175 23.991372      36     36     36
| SYSTEM.CLUSTER.REPOSITORY.QUEUE     2      4      2      0      8  0.000741  0.000032  0.004348   12316  12316  12316
| SYSTEM.CLUSTER.TRANSMIT.QUEUE       0      2      0      0      2  0.000119  0.000031  0.000207
| SYSTEM.DEFAULT.NAMELIST             0      0      0      0      0
| SYSTEM.PENDING.DATA.QUEUE           0      2      0      0      2  0.000960  0.000699  0.001222
| ------------------------------ ------ ------ ------ ------ ------ --------- --------- --------- ------- ------ ------
| All                               116     16      2      0    134  0.656197  0.000017 23.991372     593      4  12316

| MQINQ
|                                                              --- Process time (second) ---
| Queue                         Success   Info  Warning  Fail  Total  Average   Minimum   Maximum
| ------------------------------ ------ ------ ------ ------ ------ --------- --------- ---------
| (null)                             12      0      0      0     12  0.000213  0.000091  0.000786
| SYSTEM.ADMIN.COMMAND.QUEUE          8      0      0      0      8  0.000260  0.000095  0.001157
| SYSTEM.ADMIN.STATISTICS.QUEUE       0      0      0      0      0
| SYSTEM.AUTH.DATA.QUEUE              0      0      0      0      0
| SYSTEM.CHANNEL.INITQ                0      0      0      0      0
| SYSTEM.CLUSTER.COMMAND.QUEUE        2      0      0      0      2  0.000171  0.000153  0.000190
| SYSTEM.CLUSTER.REPOSITORY.QUEUE     0      0      0      0      0
| SYSTEM.CLUSTER.TRANSMIT.QUEUE       0      0      0      0      0
| SYSTEM.DEFAULT.NAMELIST             0      0      0      0      0
| SYSTEM.PENDING.DATA.QUEUE           0      0      0      0      0
| ------------------------------ ------ ------ ------ ------ ------ --------- --------- ---------
| All                                22      0      0      0     22  0.000226  0.000091  0.001157
```

Another sample output is shown in the section “(API Exit) Sample output by Summary Data Generation Utility” in page 107.

For getting better report, it is recommended to specify ShowSummaryOnly=Yes parameter in getting trace data to minimize performance degradation if detail trace report is not needed.

Inaccuracy for process time of each API call on Windows

On Windows environment, you might sometimes find inaccuracy for process time of each API call because of the limitation of Windows timer resolution. API exit utility calculates process time using performance counter (CPU counter) value in addition to time indication (millisecond) so that performance information becomes more precise.

In detail trace, API process time regarding performance counter in microsecond and process time regarding Windows timer in millisecond are shown respectively on Windows.

```
[2006/04/11 10:52:14.015]0000001497-1334875850
<MQOPEN
(BEFORE

  …

[2006/04/11 10:52:14.031]0000001497-1353965750
<MQOPEN
(AFTER
CompCode:0
Reason:0
| API process time:    0.005378  (0.016) second    Performance Counter: 19089900
```

On UNIX systems, process time is simply calculated using process time since time data can be got in microsecond.

```
[2006/08/03 13:16:55.436607]
<MQOPEN
(BEFORE

  …

[2006/08/03 13:16:55.439060]
<MQOPEN
(AFTER
CompCode:0
Reason:0
| API process time:    0.002453 second
```

Monitoring channel traffic in detail

Detail information about channel traffic can be got using channel exit in addition to channel status information shown using MQSC command.

Example:

```
       Time                         API                 Segmentation    DataLen CC Reason MsgId       Queue  Queue manager
     1 [2006/05/11 17:17:49.280867]  CHANNEL_MSG_EXIT       none        50000428  -    -   414D512051..  Q1          QM2
     2 [2006/05/11 17:17:49.280989]  CHANNEL_SEND_EXIT      first          32750  -    -   414D512051..  Q1          QM2
     3 [2006/05/11 17:17:49.281179]  CHANNEL_SEND_EXIT      middle         32750  -    -
     4 [2006/05/11 17:17:49.431942]  CHANNEL_SEND_EXIT      middle         32750  -    -
     5 [2006/05/11 17:17:49.631971]  CHANNEL_SEND_EXIT      middle         32750  -    -
     6 [2006/05/11 17:17:49.832010]  CHANNEL_SEND_EXIT      middle         32750  -    -
     7 [2006/05/11 17:17:49.832097]  CHANNEL_SEND_EXIT      middle         32750  -    -
     8 [2006/05/11 17:17:49.832204]  CHANNEL_SEND_EXIT      middle         32750  -    -
     9 [2006/05/11 17:17:49.832289]  CHANNEL_SEND_EXIT      middle         32750  -    -
    10 [2006/05/11 17:17:49.832370]  CHANNEL_SEND_EXIT      middle         32750  -    -
    11 [2006/05/11 17:17:49.832453]  CHANNEL_SEND_EXIT      middle         32750  -    -

         ........

  1525 [2006/05/11 17:17:50.041443]  CHANNEL_SEND_EXIT      middle         32750  -    -
  1526 [2006/05/11 17:17:50.041581]  CHANNEL_SEND_EXIT      middle         32750  -    -
  1527 [2006/05/11 17:17:50.041718]  CHANNEL_SEND_EXIT      middle         32750  -    -
  1528 [2006/05/11 17:17:50.041853]  CHANNEL_SEND_EXIT      middle         32750  -    -
  1529 [2006/05/11 17:17:50.041988]  CHANNEL_SEND_EXIT      middle         32750  -    -
  1530 [2006/05/11 17:17:50.048600]  CHANNEL_SEND_EXIT      last           31820  -    -
  1531 [2006/05/11 17:17:52.957220]  CHANNEL_RCV_EXIT       none              28  -    -
  1532 [2006/05/11 17:22:52.967873]  CHANNEL_SEND_EXIT      (others)          28  -    -
  1533 [2006/05/11 17:22:52.968431]  CHANNEL_RCV_EXIT       none              28  -    -
  1534 [2006/05/11 17:27:52.977817]  CHANNEL_SEND_EXIT      (others)          28  -    -
  1535 [2006/05/11 17:27:52.978183]  CHANNEL_RCV_EXIT       none              28  -    -
  1536 [2006/05/11 17:32:52.987858]  CHANNEL_SEND_EXIT      (others)          28  -    -
  1537 [2006/05/11 17:32:52.988242]  CHANNEL_RCV_EXIT       none              28  -    -
  1538 [2006/05/11 17:37:52.997841]  CHANNEL_SEND_EXIT      (others)          28  -    -
  1539 [2006/05/11 17:37:52.998241]  CHANNEL_RCV_EXIT       none              28  -    -
```

Monitoring status of API calls separating time frame

Summary information of API calls in each time frame can be checked by specifying ShowPerformanceIntervalTime parameter and/or ShowPerformanceIntervalCount parameter in parameter file. If no other trace information is needed, ShowPerformanceOnly=yes parameter setting is also recommended. Note that actual interval time may be longer than parameter setting because timing of API calls depends on MQ applications.

The following example is trace data when ShowPerformanceIntervalTime=180 is specified.

```
---------------------------------------------------------------------------------------------
Call summary - [2006/01/19 14:02:16.755181]
                                                           -------- Process time (second) -------
  API       Success      Info  Warning     Fail     Total     Average       Minimum       Maximum
  ------- -------- -------- -------- -------- -------- ------------ ------------ ------------
  MQCONN         0         0        0        0        0
  MQCONNX        1         0        0        0        1    0.003788      0.003788      0.003788
  MQDISC         0         0        0        0        0
  MQOPEN         1         0        0        0        1    0.000456      0.000456      0.000456
  MQCLOSE        0         0        0        0        0
  MQPUT1         0         0        0        0        0
  MQPUT          0         0        0        0        0
  MQGET          0         9        0        0        9   20.013358     20.002742     20.019803
  MQINQ         10         0        0        0       10    0.000569      0.000103      0.004245
  MQSET          0         0        0        0        0
  MQBEGIN        0         0        0        0        0
  MQCMIT         0         0        0        0        0
  MQBACK         0         0        0        0        0
  ------- -------- -------- -------- -------- -------- ------------ ------------ ------------
  Total         12         9        0        0       21
---------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------
Split summary (1)
  start [2006/01/19 13:59:16.582911]
  end   [2006/01/19 14:02:16.755181]
  duration         [00:03:00.172270]
                                                           -------- Process time (second) -------
  API       Success      Info  Warning     Fail     Total     Average       Minimum       Maximum
  ------- -------- -------- -------- -------- -------- ------------ ------------ ------------
  MQCONN         0         0        0        0        0
  MQCONNX        1         0        0        0        1    0.003788      0.003788      0.003788
  MQDISC         0         0        0        0        0
  MQOPEN         1         0        0        0        1    0.000456      0.000456      0.000456
  MQCLOSE        0         0        0        0        0
  MQPUT1         0         0        0        0        0
  MQPUT          0         0        0        0        0
  MQGET          0         9        0        0        9   20.013358     20.002742     20.019803
  MQINQ         10         0        0        0       10    0.000569      0.000103      0.004245
  MQSET          0         0        0        0        0
  MQBEGIN        0         0        0        0        0
  MQCMIT         0         0        0        0        0
  MQBACK         0         0        0        0        0
  ------- -------- -------- -------- -------- -------- ------------ ------------ ------------
  Total         12         9        0        0       21
---------------------------------------------------------------------------------------------

---------------------------------------------------------------------------------------------
Call summary - [2006/01/19 14:05:16.855132]
                                                           -------- Process time (second) -------
  API       Success      Info  Warning     Fail     Total     Average       Minimum       Maximum
  ------- -------- -------- -------- -------- -------- ------------ ------------ ------------
  MQCONN         0         0        0        0        0
  MQCONNX        1         0        0        0        1    0.003788      0.003788      0.003788
  MQDISC         0         0        0        0        0
  MQOPEN         1         0        0        0        1    0.000456      0.000456      0.000456
  MQCLOSE        0         0        0        0        0
  MQPUT1         0         0        0        0        0
  MQPUT          0         0        0        0        0
  MQGET          0        18        0        0       18   20.012036     20.001978     20.019803
  MQINQ         19         0        0        0       19    0.000372      0.000103      0.004245
  MQSET          0         0        0        0        0
  MQBEGIN        0         0        0        0        0
  MQCMIT         0         0        0        0        0
  MQBACK         0         0        0        0        0
  ------- -------- -------- -------- -------- -------- ------------ ------------ ------------
  Total         21        18        0        0       39
---------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------
Split summary (2)
  start [2006/01/19 14:02:16.755181]
  end   [2006/01/19 14:05:16.855132]
  duration         [00:03:00.099951]
```

```
|                                                        -------- Process time (second) -------
|  API      Success     Info  Warning     Fail    Total      Average      Minimum      Maximum
|  ------- -------- -------- -------- -------- -------- ------------ ------------ ------------
|  MQCONN         0        0        0        0        0
|  MQCONNX        0        0        0        0        0
|  MQDISC         0        0        0        0        0
|  MQOPEN         0        0        0        0        0
|  MQCLOSE        0        0        0        0        0
|  MQPUT1         0        0        0        0        0
|  MQPUT          0        0        0        0        0
|  MQGET          0        9        0        0        9    20.010714    20.001978    20.019612
|  MQINQ          9        0        0        0        9     0.000153     0.000106     0.000202
|  MQSET          0        0        0        0        0
|  MQBEGIN        0        0        0        0        0
|  MQCMIT         0        0        0        0        0
|  MQBACK         0        0        0        0        0
|  ------- -------- -------- -------- -------- -------- ------------ ------------ ------------
|  Total          9        9        0        0       18
|---------------------------------------------------------------------------------------------
-
[2006/01/19 14:06:08.465809]
<TERM
(CONNECTION
-
|---------------------------------------------------------------------------------------------
|Call summary - [2006/01/19 14:06:08.465921]
|                                                        -------- Process time (second) -------
|  API      Success     Info  Warning     Fail    Total      Average      Minimum      Maximum
|  ------- -------- -------- -------- -------- -------- ------------ ------------ ------------
|  MQCONN         0        0        0        0        0
|  MQCONNX        1        0        0        0        1     0.003788     0.003788     0.003788
|  MQDISC         1        0        0        0        1     0.000545     0.000545     0.000545
|  MQOPEN         1        0        0        0        1     0.000456     0.000456     0.000456
|  MQCLOSE        1        0        0        0        1     0.003884     0.003884     0.003884
|  MQPUT1         0        0        0        0        0
|  MQPUT          0        0        0        0        0
|  MQGET          0       21        0        0       21    19.608205    11.526817    20.019803
|  MQINQ         22        0        0        0       22     0.000344     0.000103     0.004245
|  MQSET          0        0        0        0        0
|  MQBEGIN        0        0        0        0        0
|  MQCMIT         0        0        0        0        0
|  MQBACK         0        0        0        0        0
|  ------- -------- -------- -------- -------- -------- ------------ ------------ ------------
|  Total         26       21        0        0       47
|---------------------------------------------------------------------------------------------
|---------------------------------------------------------------------------------------------
|Split summary (3)
|  start [2006/01/19 14:05:16.855132]
|  end   [2006/01/19 14:06:08.465921]
|  duration        [00:00:51.610789]
|                                                        -------- Process time (second) -------
|  API      Success     Info  Warning     Fail    Total      Average      Minimum      Maximum
|  ------- -------- -------- -------- -------- -------- ------------ ------------ ------------
|  MQCONN         0        0        0        0        0
|  MQCONNX        0        0        0        0        0
|  MQDISC         1        0        0        0        1     0.000545     0.000545     0.000545
|  MQOPEN         0        0        0        0        0
|  MQCLOSE        1        0        0        0        1     0.003884     0.003884     0.003884
|  MQPUT1         0        0        0        0        0
|  MQPUT          0        0        0        0        0
|  MQGET          0        3        0        0        3    17.185221    11.526817    20.019398
|  MQINQ          3        0        0        0        3     0.000165     0.000120     0.000221
|  MQSET          0        0        0        0        0
|  MQBEGIN        0        0        0        0        0
|  MQCMIT         0        0        0        0        0
|  MQBACK         0        0        0        0        0
|  ------- -------- -------- -------- -------- -------- ------------ ------------ ------------
|  Total          5        3        0        0        8
|---------------------------------------------------------------------------------------------
```

Reducing output data volume

When API trace is enabled in heavy transaction environment, data volume becomes huge. Hints to reduce output data volume are specified in this section.

Choosing output data format

Data volume significantly depends on output data format. The following process parameters are for choosing basic output data format.

Related parameter	Description	Page
ShowPerformanceOnly	Whether only performance data of API calls is to be put to output trace file.	56
ShowSummaryOnly	Whether only summary data for each API call is to be put to output trace file or detail off each API call is to be put.	58

For detail trace, the following parameter settings are related to output data volume.

Related parameter	Description	Page
ShowParsedData	Whether parsed data is to be put to trace file in addition to representation in graphic hexadecimal character.	51
ShowParsedMsgContent	Whether message content is to be put to trace file in parsed format.	53
ShowHexCharImage	Whether character image of each output byte is to be put to trace file in addition to graphic hexadecimal character.	49
ShowMaxLenParsedContent	Maximum message data length to be parsed and put to trace file.	51

Missing data by specifying value “no” to the parameters specified above in parameter file can be added to trace file using reformat API trace utility after getting trace data. (Refer to Reformat API trace utility in page 68 for more information.)

Eliminating output data regarding target processes and queues

If MQ API calls only for specific processes and/or queues are to be monitored, output data volume can be reduced by specifying the following parameters in parameter file.

Related parameter	Description	Page
TargetProcess	Target process name to be traced.	58
ExcludeProcess	Process name not to be traced.	43
TargetQueue	Target queue name to be traced. Process type (MQGET, MQPUT, MQINQ and MQSET) also can be specified in option.	59

Eliminating maximum message length to be traced

If main objective in getting trace is to verify API attributes, tracing content of MQ messages in MQPUT, MQPUT1 and MQGET calls might not be important. If so, data volume can be reduced by specifying the following process parameters. If message size is large, specifications of the parameters are effective.

Related parameter	**Description**	**Page**
ShowMaxLenChannel	Maximum message data length to be put to trace file for channel trace	49
ShowMaxLenGet	Maximum message data length to be put to trace file for MQGET	51
ShowMaxLenPut	Maximum message data length to be put to trace file for MQPUT	51
ShowMaxLenParsedContent	Maximum message data length to be parsed and put to trace file.	51

Exploiting Trace Data Compression Function

Data compression program can be invoked optionally to minimize disk space usage. The invocation can be specified in parameter file. (Refer to CompressCommand parameter in page 42 for more information.)

Compression command is invoked asynchronously after closing each trace file. Therefore process time for particular MQ API call does not become long even if compression command is invoked.

The following is sample process result using compression command.

Test environment

- CPU : pSeries 610 (Power 3-II 450MHz x 1 way)
- OS : AIX 5.2 + ML 06
- MQ : V6.0
- MQ Message : persistent
- MQ Message length : 1M bytes, text string data
- API call sequence : MQCONNX – MQOPEN – MQPUT 1000 times – MQCLOSE - MQDISC

Test result

Compression method	Elapsed time (second)	Total data volume (MB)	Data volume of each file.
No compression	235	361	100MB x 3 files 61MB x 1 file
compress command (file extension is .Z)	261	51	14MB x 3 files 9MB x 1 file
gzip command (file extension is .gz)	254	44	12MB x 3 files 8MB x 1 file

Tracing processes for specific message

Processes for specific message can be identified regarding message id information in API trace data. It can be done with minimum performance degradation using summary trace function (ShowSummaryOnly=yes in parameter file).

It might be useful when the following information is to be identified, for example.

- Program that stored message to MQ queue
- Program that retrieved message from MQ queue
- Waiting time in MQ queue
- Routing information of MQ messages

Example 1:

Appl_A

```
  Time                         API    CC Reason Time(sec)  ObjectName  MsgId
  [2005/09/20 21:25:34.791821] MQPUT   0    0   0.000381  Q1           414D5120514D3120202020202020202020872AAC422000000C
```

Appl_B

```
  Time                         API    CC Reason Time(sec)  ObjectName  MsgId
  [2005/09/20 21:25:35.174543] MQGET   0    0   0.000259  Q1           414D5120514D3120202020202020202020872AAC422000000C
```

In the case, message was stored by Appl_A and retrieved by Appl_B. Message was in Q1 for about 0.383 second.
(35.174543 – 34.791821 = 0.382722)

Example 2:

Sending application at sending queue manager

```
  Time                         API   CC Reason  Time(sec) ObjectName MsgId
  [2006/05/15 12:23:03.398039] MQPUT 0   0      0.010842   Q2        414D5120514D31202020202020202020B1F3674420001204
```

Sender channel at sending queue manager

```
  Time                        API         Segmentation DataLen CC Reason MsgId                         Queue   Queue manager
  [2006/05/15 12:23:03.872357] CHANNEL_SEND_EXIT  none   690 - - 414D5120514D31202020202020202020B1F3674420001204 Q2 QM2
```

Receiver channel at destination queue manager

```
  Time                        API         Segmentation DataLen CC Reason MsgId                         Queue   Queue manager
  [2006/05/15 12:23:03.879655] CHANNEL_RCV_EXIT   none   690 - - 414D5120514D31202020202020202020B1F3674420001204 Q2 QM2
```

Receiving application at destination queue manager

```
  Time                         API   CC Reason  Time(sec) ObjectName MsgId
  [2006/05/15 12:23:50.402018] MQGET 0   0      0.000116   Q2        414D5120514D31202020202020202020B1F3674420001204
```

How to identify addressing mode and threading condition of MQ applications

Checking compile/link options

One of the most basic methods to identify addressing mode and threading condition of executable MQ application programs is to find compile options and link options specified in making them.

Refer to “Building a WebSphere MQ application“ section of “WebSphere MQ Application Programming Guide” document for more information.

Checking process information part of API trace data

If API exit is already defined to queue manager and operational, addressing mode can be found in the module attributes part of process information data in API trace file. Threading information is also shown for AIX, HP-UX and Linux, in addition.

Example:

```
|----------------------------------------------------------
|Program information
|  Program name       : WebSphere MQ API Exit (SupportPac MA0W)
|  Version            : 7.41
|  Provided by        : oinoue@jp.ibm.com
|Module attributes
|  Addressing mode    : 64-bit                  <== 64-bit MQ application
|  Threading          : Multi thread (_r)       <== Multi thread
|Operating environment
|  Operating system   : AIX
```

Using ldd command (for UNIX)

You might want to identify addressing mode and threading condition of MQ applications before running them. For exit definitions of CLNTCONN channels, information of addressing mode and threading condition is required prior to exit definition to choose appropriate exit module.

On UNIX, ldd command can be used to identify addressing mode and threading condition of MQ application programs. (Of course, it is assumed that linking libraries were chosen properly in making MQ application programs.)

The following charts show summary of MQ libraries to be loaded regarding addressing mode, server/client and threading condition on each operating environment. (Subsequent MQ libraries exist in the output of ldd command.) Refer to “Building a WebSphere MQ application“ section of “WebSphere MQ Application Programming Guide” document for more information.

AIX			
32 bit mode	Server	Single thread	/usr/mqm/lib/libmqm.a(libmqm.o)
		Multi thread	/usr/mqm/lib/libmqm_r.a(libmqm_r.o)
	Client (MQI)	Single thread	/usr/mqm/lib/libmqic.a(mqic.o)
		Multi thread	/usr/mqm/lib/libmqic_r.a(mqic_r.o)
64 bit mode	Server	Single thread	/usr/mqm/lib64/libmqm.a(libmqm.o)
		Multi thread	/usr/mqm/lib64/libmqm_r.a(libmqm_r.o)
	Client (MQI)	Single thread	/usr/mqm/lib64/libmqic.a(mqic.o)
		Multi thread	/usr/mqm/lib64/libmqic_r.a(mqic_r.o)

Solaris			
32 bit mode	Server	Single thread	/opt/mqm/lib/libmqm.so
		Multi thread	
	Client (MQI)	Single thread	/opt/mqm/lib/libmqic.so
		Multi thread	
64 bit mode	Server	Single thread	/opt/mqm/lib64/libmqm.so
		Multi thread	
	Client (MQI)	Single thread	/opt/mqm/lib64/libmqic.so
		Multi thread	

HP-UX (PA-RISC)			
32 bit mode	Server	Single thread	/opt/mqm/lib/libmqm.sl
		Multi thread	/opt/mqm/lib/libmqm_r.sl
	Client (MQI)	Single thread	/opt/mqm/lib/libmqic.sl
		Multi thread	/opt/mqm/lib/libmqic_r.sl
64 bit mode	Server	Single thread	/opt/mqm/lib64/libmqm.sl
		Multi thread	/opt/mqm/lib64/libmqm_r.sl
	Client (MQI)	Single thread	/opt/mqm/lib64/libmqic.sl
		Multi thread	/opt/mqm/lib64/libmqic_r.sl

Linux, HP-UX (IA-64)			
32 bit mode	Server	Single thread	/opt/mqm/lib/libmqm.so
		Multi thread	/opt/mqm/lib/libmqm_r.so
	Client (MQI)	Single thread	/opt/mqm/lib/libmqic.so
		Multi thread	/opt/mqm/lib/libmqic_r.so
64 bit mode	Server	Single thread	/opt/mqm/lib64/libmqm.so
		Multi thread	/opt/mqm/lib64/libmqm_r.so
	Client (MQI)	Single thread	/opt/mqm/lib64/libmqic.so
		Multi thread	/opt/mqm/lib64/libmqic_r.so

Example (on AIX):

```
# ldd mq_appl_001
mq_appl_001 needs:
         /usr/lib/libc.a(shr.o)
              ...
         /usr/mqm/lib/libmqic.a(mqic.o)         <== 32-bit MQ client (MQI) program, single thread
         /unix
         /usr/lib/libcrypt.a(shr.o)
         /usr/mqm/lib/libmqmcs.a(shr.o)
              ...

# ldd mq_appl_999
mq_appl_999 needs:
         /usr/lib/libc.a(shr_64.o)
              ...
         /usr/mqm/lib64/libmqic_r.a(mqic_r.o)   <== 64-bit MQ client (MQI) program, multi thread
         /unix
         /usr/lib/libcrypt.a(shr_64.o)
         /usr/mqm/lib64/libmqmcs_r.a(shr.o)
              ...
```

Appendix C. Sample Trace Data

(API Exit) with ShowPerformanceOnly=yes parameter

```
|-----------------------------------------------------------
|Program information
|  Program name       : WebSphere MQ API Exit (SupportPac MA0W)

    ........

|Process parameters
|  Parameter file     : E:¥cpp¥Apiexit¥tracedata¥qx.def
|  Show max MQPUT len : (all)
|  Show max MQGET len : (all)
|  Show max chl len   : (all)
|  Show data conv     : no
|  Show summary only  : no
|  Show parsed data   : yes
|  Show parsed XML    : no
|  Show hex char image: yes
|  Show hex char ccsid: 0
|  Show perf only     : yes
|  Perf intvl (time)  : 0
|  Perf intvl (count) : 0
|  Max file size      : 100,000,000
|  Max XML line length: 100
|  Compress command   : gzip $
|  Compress threshold : 1,000,000
|  Create empty file  : no
|  Write buffer size  : 50,000
|  Write cache        : Enabled
|  Write interval     : 30
|  Force write        : no
|  Check update req   : 1
|  Exclude proc name  : amqzdmaa
|  Exclude proc name  : amqzfuma
|   (Debug flag)      : off
|-----------------------------------------------------------
-
[2005/11/16 01:30:51.156]0000000003-1843794397
<INIT
(CONNECTION
-
[2005/11/16 01:32:20.937]0000000003-2165148715
<TERM
(CONNECTION
-
|--------------------------------------------------------------------------------------------
|Call summary   [2005/11/16 01:32:20.937]0000000003-2165148777
|                                                       -------- Process time (second) -------
|  API      Success      Info  Warning     Fail    Total     Average      Minimum      Maximum
|  ------- -------- -------- -------- -------- -------- ------------ ------------ ------------
|  MQCONN         0        0        0        0        0
|  MQCONNX        1        0        0        0        1     1.689837     1.689837     1.689837
|  MQDISC         1        0        0        0        1     0.000103     0.000103     0.000103
|  MQOPEN         7        0        0        0        7     0.001306     0.000052     0.003306
|  MQCLOSE        5        0        0        0        5     0.001384     0.000039     0.006373
|  MQPUT1         0        0        0        0        0
|  MQPUT          1        0        0        0        1     0.000365     0.000365     0.000365
|  MQGET          2        4        1        0        7    12.521350     0.000045    87.648579
|  MQINQ          5        0        0        0        5     0.000129     0.000039     0.000465
|  MQSET          0        0        0        0        0
|  MQBEGIN        0        0        0        0        0
|  MQCMIT         4        0        0        0        4     0.002339     0.000029     0.006658
|  MQBACK         0        0        0        0        0
|  ------- -------- -------- -------- -------- -------- ------------ ------------ ------------
|  Total         26        4        1        0       31
|--------------------------------------------------------------------------------------------
```

(API Exit) with ShowSummaryOnly=yes parameter

```
|-----------------------------------------------------------
|Program information
|  Program name       : WebSphere MQ API Exit (SupportPac MA0W)
|  Version            : 4.60
|  Provided by        : oinoue@jp.ibm.com
|Module attributes
|  Addressing mode    : 64-bit
|  Threading          : Multi thread (_r)
|Operating environment
|  Operating system   : AIX
|  Release            : 2
|  Version            : 5
|  Node name          : liverpool
|  Machine            : 0006D1BA4C00
|API Exit context
|  Exit name          : qxEntrySummary
|  API caller type    : MQXACT_INTERNAL
|  Queue manager      : QM1
|  Environment        : MQXE_OTHER
|  User id            : apiexit
|  Connection name    :
|  Appl/Channel name  : amqrrmfa
|  Appl/Channel type  : MQAT_UNKNOWN
|  Program name       : amqrrmfa
|  Program description: WMQ Repository process (for clusters)
|  Process id         : 368892
|  Thread id          : 1
|  Process id (real)  : 368892
|  Thread id  (real)  : 1
|Process parameters
|  Parameter file     : /home/apiexit/tracedata/sum/qx.def
|  Show max MQPUT len : (all)
|  Show max MQGET len : (all)
|  Show max chl len   : (all)
|  Show data conv     : no
|  Show summary only  : yes
|  Show parsed data   : yes
|  Show hex char image: yes
|  Show hex char ccsid: 0
|  Show perf only     : no
|  Perf intvl (time)  : 0
|  Perf intvl (count) : 0
|  Max file size      : 100,000,000
|  Compress command   : (none)
|  Compress threshold : 1,000,000
|  Create empty file  : no
|  Write buffer size  : 50,000
|  Write cache        : Enabled
|  Write interval     : 30
|  Force write        : no
|  (Debug flag)       : off
|-----------------------------------------------------------
-
|-----------------------------------------------------------------------------------------
|Call summary   [2005/10/24 22:48:51.662734]
|                                                          -------- Process time (second) -------
|  API      Success     Info  Warning     Fail    Total     Average      Minimum      Maximum
|  ------- -------- -------- -------- -------- -------- ------------ ------------ ------------
|  MQCONN         0        0        0        0        0
|  MQCONNX        0        0        0        0        0
|  MQDISC         0        0        0        0        0
|  MQOPEN         0        0        0        0        0
|  MQCLOSE        0        0        0        0        0
|  MQPUT1         0        0        0        0        0
|  MQPUT          0        0        0        0        0
|  MQGET          0        0        0        0        0
|  MQINQ          0        0        0        0        0
|  MQSET          0        0        0        0        0
|  MQBEGIN        0        0        0        0        0
|  MQCMIT         0        0        0        0        0
|  MQBACK         0        0        0        0        0
|  ------- -------- -------- -------- -------- -------- ------------ ------------ ------------
|  Total          0        0        0        0        0
|-----------------------------------------------------------------------------------------
-
[2005/10/24 22:48:51.662808]
<INIT
(CONNECTION
   Time                            API      CC Reason  Time(sec)  ObjectName                          MsgId
 1 [2005/10/24 22:48:52.948335]  MQCONNX   0    0      1.285385
 2 [2005/10/24 22:48:52.950297]  MQOPEN    0    0      0.001446
 3 [2005/10/24 22:48:52.951922]  MQINQ     0    0      0.001229
 4 [2005/10/24 22:48:52.953141]  MQOPEN    0    0      0.000835  SYSTEM.DEFAULT.NAMELIST
 5 [2005/10/24 22:48:52.954911]  MQCLOSE   0    0      0.001408  SYSTEM.DEFAULT.NAMELIST
 6 [2005/10/24 22:48:52.957905]  MQOPEN   0    0       0.002713  SYSTEM.CLUSTER.COMMAND.QUEUE
 7 [2005/10/24 22:48:52.988318]  MQINQ    0    0       0.009964  SYSTEM.CLUSTER.COMMAND.QUEUE
 8 [2005/10/24 22:48:52.989895]  MQOPEN   0    0       0.001100  SYSTEM.CLUSTER.TRANSMIT.QUEUE
 9 [2005/10/24 22:48:52.990791]  MQOPEN   0    0       0.000211  SYSTEM.CLUSTER.REPOSITORY.QUEUE
10 [2005/10/24 22:48:52.991914]  MQGET    0    0       0.000516  SYSTEM.CLUSTER.REPOSITORY.QUEUE      414D5120514D31202020
11 [2005/10/24 22:48:52.995186]  MQOPEN   0    0       0.000248  SYSTEM.CLUSTER.TRANSMIT.QUEUE
12 [2005/10/24 22:48:52.996026]  MQGET    2 2033       0.000143  SYSTEM.CLUSTER.TRANSMIT.QUEUE        (2033 = MQRC_NO_MSG
13 [2005/10/24 22:48:52.996591]  MQCLOSE  0    0       0.000179  SYSTEM.CLUSTER.TRANSMIT.QUEUE
14 [2005/10/24 22:48:52.996887]  MQINQ    0    0       0.000132
15 [2005/10/24 22:48:52.997585]  MQGET    2 2033       0.000252  SYSTEM.CLUSTER.COMMAND.QUEUE         (2033 = MQRC_NO_MSG
16 [2005/10/24 22:48:52.998369]  MQOPEN   0    0       0.000172
17 [2005/10/24 22:48:52.998870]  MQINQ    0    0       0.000127
18 [2005/10/24 22:48:52.999681]  MQGET    1 2079      0.000250  SYSTEM.CLUSTER.REPOSITORY.QUEUE      414D5120514D31202020
19 [2005/10/24 22:48:53.008298]  MQPUT    0    0       0.004930  SYSTEM.CLUSTER.REPOSITORY.QUEUE      414D5120514D31202020
20 [2005/10/24 22:48:53.009011]  MQCLOSE  0    0       0.000142
21 [2005/10/24 22:48:53.048517]  MQCMIT   0    0       0.032514
22 [2005/10/24 22:48:53.048857]  MQINQ    0    0       0.000152
23 [2005/10/24 22:48:53.049888]  MQCMIT   0    0       0.000107
24 [2005/10/24 22:49:24.240014]  MQGET    0    0      31.189702  SYSTEM.CLUSTER.COMMAND.QUEUE         414D5120514D31202020
25 [2005/10/24 22:49:24.244356]  MQCMIT   0    0       0.003469
26 [2005/10/24 22:49:24.245269]  MQGET    2 2033       0.000390  SYSTEM.CLUSTER.REPOSITORY.QUEUE      (2033 = MQRC_NO_MSG_
27 [2005/10/24 22:49:24.246124]  MQGET    2 2033       0.000148  SYSTEM.CLUSTER.REPOSITORY.QUEUE      (2033 = MQRC_NO_MSG_
28 [2005/10/24 22:49:24.246749]  MQCMIT   0    0       0.000122
29 [2005/10/24 22:49:24.247054]  MQCLOSE  0    0       0.000185  SYSTEM.CLUSTER.REPOSITORY.QUEUE
30 [2005/10/24 22:49:24.247402]  MQCLOSE  0    0       0.000152  SYSTEM.CLUSTER.COMMAND.QUEUE
31 [2005/10/24 22:49:24.248031]  MQDISC   0    0       0.000484
-
[2005/10/24 22:49:24.248110]
<TERM
(CONNECTION
-
|--------------------------------------------------------------------------------------------
|Call summary   [2005/10/24 22:49:24.248132]
|                                                         -------- Process time (second) -------
|  API      Success     Info  Warning     Fail    Total     Average      Minimum      Maximum
|  ------- -------- -------- -------- -------- -------- ------------ ------------ ------------
|  MQCONN         0        0        0        0        0
|  MQCONNX        1        0        0        0        1    1.285385     1.285385     1.285385
|  MQDISC         1        0        0        0        1    0.000484     0.000484     0.000484
|  MQOPEN         7        0        0        0        7    0.000961     0.000172     0.002713
|  MQCLOSE        5        0        0        0        5    0.000413     0.000142     0.001408
|  MQPUT1         0        0        0        0        0
|  MQPUT          1        0        0        0        1    0.004930     0.004930     0.004930
|  MQGET          2        4        1        0        7    4.455914     0.000143    31.189702
|  MQINQ          5        0        0        0        5    0.002321     0.000127     0.009964
|  MQSET          0        0        0        0        0
|  MQBEGIN        0        0        0        0        0
|  MQCMIT         4        0        0        0        4    0.009053     0.000107     0.032514
|  MQBACK         0        0        0        0        0
|  ------- -------- -------- -------- -------- -------- ------------ ------------ ------------
|  Total         26        4        1        0       31
|--------------------------------------------------------------------------------------------
```

(API Exit) with default parameter values

```
|--------------------------------------------------------------
|Program information
|  Program name       : WebSphere MQ API Exit (SupportPac MAOW)
|  Version            : 6.01
|  Provided by        : oinoue@jp.ibm.com
|Module attributes
|  Addressing mode    : 32-bit
|  Threading          : Common to single thread and multi thread
|Operating environment
|  Operating system   : Windows XP
|  Major version      : 5
|  Minor version      : 1
|  Service level      : Service Pack 2
|  Build number       : 2600
|  Computer name      : OINOUEM
|API Exit context
|  Exit name          : qxEntry
|  API caller type    : EXTERNAL
|  Queue manager      : QM1
|  Environment        : MCA
|  User id            : oinoue
|  Connection name    :
|  Appl/Channel name  : s¥IBM¥WMQ60¥bin¥amqrmppa.exe
|  Appl/Channel type  : Windows
|  Program name       : amqrmppa.exe
|  Program description: WMQ Channel process pooling job
|  Process id         : 2468
|  Thread id          : 2
|  Process id (real)  : 2468
|  Thread id  (real)  : 2828
|Process parameters
|  Parameter file     : E:¥cpp¥ApiExit¥tracedata¥qx.def
|  Check update req   : 30
|  Show summary only  : no
|  Show perf only     : no
|  Perf intvl (time)  : 0
|  Perf intvl (count) : 0
|  Show parsed data   : yes
|  Show parsed content: yes
|  Max out line length: 100
|  Show hex char image: yes
|  Show hex char ccsid: 0
|  Show data conv     : no
|  Show max len MQPUT : (all)
|  Show max len MQGET : (all)
|  Show max len chl   : (all)
|  Show max len parsed: (all)
|  Count only selected: no
|  Alert file (error) : yes
|  Alert file (warn)  : no
|  Suppress alert     : 2, 2085, MQOPEN , SYSTEM.BROKER.INTER.BROKER.COMMUNICATIONS, DataFlowEngine
|  Suppress alert     : 2, 2038, MQINQ  , SYSTEM.BROKER.ADMIN.REPLY, DataFlowEngine
|  Max file size      : 100,000,000
|  Compress command   : (none)
|  Compress threshold : 1,000,000
|  Create empty file  : yes
|  Write stream mode  : no
|  Write cache        : yes
|  Write after call   : yes
|  Write buffer size  : 10,000
|  Write interval     : 30
|  (Debug flag)       : off
|--------------------------------------------------------------
-
[2006/12/27 17:22:14.437]0000000024-0103611786
<INIT
(CONNECTION
-
|--- Seq # : 1 ---------------------------------------
[2006/12/27 17:22:14.437]0000000024-0103611905
<MQCONNX
(BEFORE
QMgrName:QM1
CNO:(12=0xC bytes, Address 00C3E8E0)
00000000:434E4F20 01000000 00010000           [CNO ........]
|> Version                  : 1
|> Options                  : STANDARD_BINDING / SHARED_BINDING
-
[2006/12/27 17:22:14.578]0000000024-0104131274
<MQCONNX
(AFTER
CompCode:0
Reason:0
```

```
| API process time:    0.141375  (0.141) second    Performance Counter: 519369
Hconn:0x00ABB7B8
CNO:(12=0xC bytes, Address 00C3E8E0)
00000000:434E4F20 01000000 00010000           [CNO ........]
|> Version                  : 1
|> Options                  : STANDARD_BINDING / SHARED_BINDING
-
|--- Seq # : 2 ---------------------------------------
[2006/12/27 17:22:14.578]0000000024-0104131754
<MQOPEN
(BEFORE
Hconn:0x00ABB7B8
OD:(336=0x150 bytes, Address 00C3EEA8)
00000000:4F442020 03000000 05000000 00000000  [OD  ............]
00000010:00000000 00000000 00000000 00000000  [................]
00000020-0000002F same as above
00000030:00000000 00000000 00000000 514D3120  [............QM1 ]
00000040:20202020 20202020 20202020 20202020  [                ]
00000050-0000005F same as above
00000060:20202020 20202020 20202020 414D512E  [            AMQ.]
00000070:2A000000 00000000 00000000 00000000  [*...............]
00000080:00000000 00000000 00000000 00000000  [................]
00000090-0000014F same as above
|> Version                  : 3
|> ObjectType               : Q_MGR
|> ObjectName               :
|> ObjectQMgrName           : QM1
|> DynamicQName             : AMQ.*
|> AlternateUserId          :
|> --- OD version 2----------
|> RecsPresent              : 0
|> KnownDestCount           : 0
|> UnknownDestCount         : 0
|> InvalidDestCount         : 0
|> --- OD version 3----------
|> AlternateSecurityId      : (null)
|> ResolvedQName            :
|> ResolvedQMgrName         :
| ObjectName:(queue_manager)
Options:0x00000020
|> Options                  : INQUIRE
-
[2006/12/27 17:22:14.578]0000000024-0104133296
<MQOPEN
(AFTER
CompCode:0
Reason:0
| API process time:    0.000419  (0.000) second    Performance Counter: 1542
OD:(336=0x150 bytes, Address 00C3EEA8)
00000000:4F442020 03000000 05000000 00000000  [OD  ............]
00000010:00000000 00000000 00000000 00000000  [................]
00000020-0000002F same as above
00000030:00000000 00000000 00000000 514D3120  [............QM1 ]
00000040:20202020 20202020 20202020 20202020  [                ]
00000050-0000005F same as above
00000060:20202020 20202020 20202020 414D512E  [            AMQ.]
00000070:2A000000 00000000 00000000 00000000  [*...............]
00000080:00000000 00000000 00000000 00000000  [................]
00000090-0000009F same as above
000000A0:00000000 00000000 00000000 01000000  [................]
000000B0:00000000 00000000 00000000 00000000  [................]
000000C0-000000EF same as above
000000F0:20202020 20202020 20202020 20202020  [                ]
00000100-0000014F same as above
|> Version                  : 3
|> ObjectType               : Q_MGR
|> ObjectName               :
|> ObjectQMgrName           : QM1
|> DynamicQName             : AMQ.*
|> AlternateUserId          :
|> --- OD version 2----------
|> RecsPresent              : 0
|> KnownDestCount           : 1
|> UnknownDestCount         : 0
|> InvalidDestCount         : 0
|> --- OD version 3----------
|> AlternateSecurityId      : (null)
|> ResolvedQName            :
|> ResolvedQMgrName         :
| ObjectName:(queue_manager)
Hobj:0x003EFBF0
-
|--- Seq # : 3 ---------------------------------------
[2006/12/27 17:22:14.578]0000000024-0104134026
<MQINQ
(BEFORE
Hconn:0x00ABB7B8
Hobj:0x003EFBF0
| ObjectName:(queue_manager)
SelectorCount:14
```

```
Selectors:(56=0x38 bytes, Address 00C3E9F0)
00000000:02000000 22000000 37000000 38000000  [....″...7...8...]
00000010:21000000 8A000000 89000000 7C000000  [!...........|...]
00000020:7A000000 82000000 81000000 0D000000  [z...............]
00000030:DF070000 EA070000                    [........]
|> CODED_CHAR_SET_ID        :
|> DIST_LISTS               :
|> CHANNEL_AUTO_DEF         :
|> CHANNEL_AUTO_DEF_EVENT   :
|> MAX_UNCOMMITTED_MSGS     :
|> ACTIVITY_RECORDING       :
|> TRACE_ROUTE_RECORDING    :
|> MONITORING_AUTO_CLUSSDR  :
|> MONITORING_CHANNEL       :
|> STATISTICS_AUTO_CLUSSDR  :
|> STATISTICS_CHANNEL       :
|> MAX_MSG_LENGTH           :
|> Q_MGR_NAME               :
|> CHANNEL_AUTO_DEF_EXIT    :
IntAttrCount:12
CharAttrLength:176
-
[2006/12/27 17:22:14.578]0000000024-0104134546
<MQINQ
(AFTER
CompCode:0
Reason:0
| API process time:    0.000141  (0.000) second    Performance Counter: 520
IntAttrs:(48=0x30 bytes, Address 00C3E9C0)
00000000:A4030000 01000000 00000000 01000000  [.;..............]
00000010:10270000 02000000 02000000 FDFFFFFF  [.'..............]
00000020:00000000 FDFFFFFF 00000000 00004000  [..............@.]
|> CODED_CHAR_SET_ID        : 932
|> DIST_LISTS               : supported
|> CHANNEL_AUTO_DEF         : disabled
|> CHANNEL_AUTO_DEF_EVENT   : enabled
|> MAX_UNCOMMITTED_MSGS     : 10000
|> ACTIVITY_RECORDING       : message
|> TRACE_ROUTE_RECORDING    : message
|> MONITORING_AUTO_CLUSSDR  : queue manager
|> MONITORING_CHANNEL       : off / disabled
|> STATISTICS_AUTO_CLUSSDR  : queue manager
|> STATISTICS_CHANNEL       : off / disabled
|> MAX_MSG_LENGTH           : 4194304
CharAttrs:(176=0xB0 bytes, Address 00C3ECB8)
00000000:514D3120 20202020 20202020 20202020  [QM1             ]
00000010:20202020 20202020 20202020 20202020  [                ]
00000020-000000AF same as above
|> Q_MGR_NAME               : QM1
|> CHANNEL_AUTO_DEF_EXIT    :
-
|--- Seq # : 4 ---------------------------------------
[2006/12/27 17:22:14.578]0000000024-0104135433
<MQCLOSE
(BEFORE
Hconn:0x00ABB7B8
Hobj:0x003EFBF0
| ObjectName:(queue_manager)
Options:0x00000000
|> Options                  : (none)
-
[2006/12/27 17:22:14.578]0000000024-0104135890
<MQCLOSE
(AFTER
CompCode:0
Reason:0
| API process time:    0.000124  (0.000) second    Performance Counter: 457
Hobj:0xFFFFFFFF
-
|--- Seq # : 5 ---------------------------------------
[2006/12/27 17:22:15.125]0000000024-0106067236
<MQOPEN
(BEFORE
Hconn:0x00ABB7B8
OD:(168=0xA8 bytes, Address 00AB2DA0)
00000000:4F442020 01000000 01000000 51310000  [OD  ........Q1..]
00000010:00000000 00000000 00000000 00000000  [................]
00000020-0000005F same as above
00000060:00000000 00000000 00000000 414D512E  [............AMQ.]
00000070:2A000000 00000000 00000000 00000000  [*...............]
00000080:00000000 00000000 00000000 00000000  [................]
00000090-0000009F same as above
000000A0:00000000 00000000                    [........]
|> Version                  : 1
|> ObjectType               : QUEUE
|> ObjectName               : Q1
|> ObjectQMgrName           :
|> DynamicQName             : AMQ.*
|> AlternateUserId          :
| ObjectName:Q1
```

```
Options:0x00002010
|> Options                  : OUTPUT / FAIL_IF_QUIESCING
-
[2006/12/27 17:22:15.125]0000000024-0106087226
<MQOPEN
(AFTER
CompCode:0
Reason:0
| API process time:    0.005523  (0.000) second    Performance Counter: 19990
OD:(168=0xA8 bytes, Address 00AB2DA0)
00000000:4F442020 01000000 01000000 51310000  [OD  ........Q1..]
00000010:00000000 00000000 00000000 00000000  [................]
00000020-0000005F same as above
00000060:00000000 00000000 00000000 414D512E  [............AMQ.]
00000070:2A000000 00000000 00000000 00000000  [*...............]
00000080:00000000 00000000 00000000 00000000  [................]
00000090-0000009F same as above
000000A0:00000000 00000000                    [........]
|> Version                  : 1
|> ObjectType               : QUEUE
|> ObjectName               : Q1
|> ObjectQMgrName           :
|> DynamicQName             : AMQ.*
|> AlternateUserId          :
| ObjectName:Q1
Hobj:0x003EFBF0
-
|--- Seq # : 6 ----------------------------------------
[2006/12/27 17:22:15.125]0000000024-0106092993
<MQPUT
(BEFORE
Hconn:0x00ABB7B8
Hobj:0x003EFBF0
| ObjectName:Q1
MD:(364=0x16C bytes, Address 00AB2DA0)
00000000:4D442020 02000000 00000000 08000000  [MD  ............]
00000010:FFFFFFFF 00000000 22020000 AF030000  [........".......]
00000020:4D514852 46322020 00000000 01000000  [MQHRF2  ........]
00000030:00000000 00000000 00000000 00000000  [................]
00000040-0000005F same as above
00000060:00000000 51524550 4C592020 20202020  [....QREPLY      ]
00000070:20202020 20202020 20202020 20202020  [                ]
00000080-0000008F same as above
00000090:20202020 00000000 00000000 00000000  [    ............]
000000A0:00000000 00000000 00000000 00000000  [................]
000000B0-0000014F same as above
00000150:00000000 00000000 00000000 01000000  [................]
00000160:00000000 00000000 FFFFFFFF           [............]
|> Version                  : 2
|> Report                   : (none)
|> MsgType                  : DATAGRAM
|> Expiry                   : -1
|> Feedback                 : 0
|> Encoding                 : 546 (Native encoding) INTEGER_REVERSED / DECIMAL_REVERSED / FLOAT_IEEE_REVERSED
|> CodedCharSetId           : 943
|> Format                   : MQHRF2
|> Priority                 : 0
|> Persistence              : PERSISTENT
|> MsgId                    : (null)
|> CorrelId                 : (null)
|> BackoutCount             : 0
|> ReplyToQ                 : QREPLY
|> ReplyToQMgr              :
|> UserIdentifier           :
|> AccountingToken          : (null)
|> ApplIdentityData         :
|> PutApplType              : no context
|> PutApplName              :
|> PutDate                  :
|> PutTime                  :
|> ApplOriginData           :
|> --- MD version 2----------
|> GroupId                  : (null)
|> MsgSeqNumber             : 1
|> Offset                   : 0
|> MsgFlags                 : (none)
|> OriginalLength           : -1
PMO:(128=0x80 bytes, Address 00AB2F0C)
00000000:504D4F20 01000000 42800000 FFFFFFFF  [PMO ....B.......]
00000010:00000000 00000000 00000000 00000000  [................]
00000020-0000007F same as above
|> Version                  : 1
|> Options                  : SYNCPOINT / NEW_MSG_ID / LOGICAL_ORDER
|> Timeout                  : -1
|> Context                  : 00000000
|> KnownDestCount           : 0
|> UnknownDestCount         : 0
|> InvalidDestCount         : 0
|> ResolvedQName            :
|> ResolvedQMgrName         :
```

```
BufferLength:198
Buffer:(198=0xC6 bytes, Address 00AB2F90)
00000000:52464820 02000000 84000000 22020000  [RFH ........″...]
00000010:BA130000 4D515354 52202020 00000000  [....MQSTR   ....]
00000020:B8040000 5C000000 3C6D6364 3E3C4D73  [....¥...<mcd><Ms]
00000030:643E4D52 4D3C2F4D 73643E3C 5365743E  [d>MRM</Msd><Set>]
00000040:444D5051 4F434330 37393939 393C2F53  [DMPQOCC079999</S]
00000050:65743E3C 54797065 3E4D5347 5F4D4958  [et><Type>MSG_MIX]
00000060:45444341 53455F49 443C2F54 7970653E  [EDCASE_ID</Type>]
00000070:3C466D74 3E435746 3C2F466D 743E3C2F  [<Fmt>CWF</Fmt></]
00000080:6D63643E 4F73616D 7520496E 6F756520  [mcd>Osamu Inoue ]
00000090:536F7461 20496E6F 75658140 814082A0  [Sota Inoue.@.@..]
000000A0:82A282A4 20205361 746F6520 82A682A6  [....  Satoe ....]
000000B0:82A682A6 8140B0B3 B2A9B3C2 B3C9F2CD  [.....@..........]
000000C0:F2F4E9DA D9F8                        [......]
 |> ---------- offset 0 (=0x0) - MQHRF2 (Rules and formatting header 2)
 |> Version                  : 2
 |> StrucLength              : 132
 |> Encoding                 : 546 (Native encoding) INTEGER_REVERSED / DECIMAL_REVERSED / FLOAT_IEEE_REVERSED
 |> CodedCharSetId           : 5050
 |> Format                   : MQSTR
 |> Flags                    : 0
 |> NameValueCCSID           : 1208
 |> ---------- offset 40 (=0x28) - XML data
 |> <mcd>
 |>   <Msd>
 |>     MRM
 |>   </Msd>
 |>   <Set>
 |>     DMPQOCC079999
 |>   </Set>
 |>   <Type>
 |>     MSG_MIXEDCASE_ID
 |>   </Type>
 |>   <Fmt>
 |>     CWF
 |>   </Fmt>
 |> </mcd>
-
[2006/12/27 17:22:15.296]0000000024-0106659283
<MQPUT
(AFTER
CompCode:0
Reason:0
| API process time:    0.158235  (0.171) second    Performance Counter: 566290
MD:(364=0x16C bytes, Address 00AB2DA0)
00000000:4D442020 02000000 00000000 08000000  [MD  ............]
00000010:FFFFFFFF 00000000 22020000 AF030000  [........″.......]
00000020:4D514852 46322020 00000000 01000000  [MQHRF2  ........]
00000030:414D5120 514D3120 20202020 20202020  [AMQ QM1         ]
00000040:E32C9245 20000902 00000000 00000000  [.,.E ...........]
00000050:00000000 00000000 00000000 00000000  [................]
00000060:00000000 51524550 4C592020 20202020  [....QREPLY      ]
00000070:20202020 20202020 20202020 20202020  [                ]
00000080-0000008F same as above
00000090:20202020 00000000 00000000 00000000  [    ............]
000000A0:00000000 00000000 00000000 00000000  [................]
000000B0-000000BF same as above
000000C0:00000000 6F696E6F 75652020 20202020  [....oinoue      ]
000000D0:16010515 000000BE 043E324B 2CBC1A82  [.........>2K,...]
000000E0:8BA628EB 03000000 00000000 0000000B  [..(.............]
000000F0:20202020 20202020 20202020 20202020  [                ]
00000100-0000010F same as above
00000110:0B000000 756E5C69 73656C5F 6D716C6F  [....un¥isel_mqlo]
00000120:6E677275 6E5F6D71 69633332 2E657865  [ngrun_mqic32.exe]
00000130:32303036 31323237 30383232 31353132  [2006122708221512]
00000140:20202020 00000000 00000000 00000000  [    ............]
00000150:00000000 00000000 00000000 01000000  [................]
00000160:00000000 00000000 FFFFFFFF           [............]
 |> Version                  : 2
 |> Report                   : (none)
 |> MsgType                  : DATAGRAM
 |> Expiry                   : -1
 |> Feedback                 : 0
 |> Encoding                 : 546 (Native encoding) INTEGER_REVERSED / DECIMAL_REVERSED / FLOAT_IEEE_REVERSED
 |> CodedCharSetId           : 943
 |> Format                   : MQHRF2
 |> Priority                 : 0
 |> Persistence              : PERSISTENT
 |> MsgId                    : 414D5120 514D3120 20202020 20202020 E32C9245 20000902  [AMQ QM1         .,.E ...]
 |> CorrelId                 : (null)
 |> BackoutCount             : 0
 |> ReplyToQ                 : QREPLY
 |> ReplyToQMgr              :
 |> UserIdentifier           : oinoue
 |> AccountingToken          : 16010515 000000BE 043E324B 2CBC1A82 8BA628EB 03000000 00000000 0000000B
 |> ApplIdentityData         :
 |> PutApplType              : Windows
 |> PutApplName              : un¥isel_mqlongrun_mqic32.exe
 |> PutDate                  : 20061227
 |> PutTime                  : 08221512
 |> ApplOriginData           :
 |> --- MD version 2----------
 |> GroupId                  : (null)
 |> MsgSeqNumber             : 1
 |> Offset                   : 0
 |> MsgFlags                 : (none)
 |> OriginalLength           : -1
 PMO:(128=0x80 bytes, Address 00AB2F0C)
 00000000:504D4F20 01000000 42800000 FFFFFFFF  [PMO ....B.......]
 00000010:00000000 01000000 00000000 00000000  [................]
 00000020:51312020 20202020 20202020 20202020  [Q1              ]
 00000030:20202020 20202020 20202020 20202020  [                ]
 00000040-0000004F same as above
 00000050:514D3120 20202020 20202020 20202020  [QM1             ]
 00000060:20202020 20202020 20202020 20202020  [                ]
 00000070-0000007F same as above
 |> Version                  : 1
 |> Options                  : SYNCPOINT / NEW_MSG_ID / LOGICAL_ORDER
 |> Timeout                  : -1
 |> Context                  : 00000000
 |> KnownDestCount           : 1
 |> UnknownDestCount         : 0
 |> InvalidDestCount         : 0
 |> ResolvedQName            : Q1
 |> ResolvedQMgrName         : QM1
 -
 |--- Seq # : 7 ---------------------------------------
 [2006/12/27 17:22:15.296]0000000024-0106669849
 <MQCLOSE
 (BEFORE
 Hconn:0x00ABB7B8
 Hobj:0x003EFBF0
 | ObjectName:Q1
 Options:0x00000000
 |> Options                  : (none)
 -
 [2006/12/27 17:22:15.296]0000000024-0106678234
 <MQCLOSE
 (AFTER
 CompCode:0
 Reason:0
 | API process time:    0.002332  (0.000) second    Performance Counter: 8385
 Hobj:0xFFFFFFFF
 -
 |--- Seq # : 8 ---------------------------------------
 [2006/12/27 17:22:15.296]0000000024-0106680276
 <MQCMIT
 (BEFORE
 Hconn:0x00ABB7B8
 -
 [2006/12/27 17:22:15.312]0000000024-0106726357
 <MQCMIT
 (AFTER
 CompCode:0
 Reason:0
 | API process time:    0.012846  (0.016) second    Performance Counter: 46081
 -
 |--- Seq # : 9 ---------------------------------------
 [2006/12/27 17:22:15.328]0000000024-0106786135
 <MQBACK
 (BEFORE
 Hconn:0x00ABB7B8
 -
 [2006/12/27 17:22:15.328]0000000024-0106786492
 <MQBACK
 (AFTER
 CompCode:0
 Reason:0
 | API process time:    0.000099  (0.000) second    Performance Counter: 357
 -
 |--- Seq # : 10 ---------------------------------------
 [2006/12/27 17:22:15.359]0000000024-0106926816
 <MQDISC
 (BEFORE
 Hconn:0x00ABB7B8
 -
 [2006/12/27 17:22:15.359]0000000024-0106927456
 <MQDISC
 (AFTER
 CompCode:0
 Reason:0
 | API process time:    0.000177  (0.000) second    Performance Counter: 640
 Hconn:0xFFFFFFFF
 -
 [2006/12/27 17:22:15.359]0000000024-0106927599
 <TERM
 (CONNECTION
 -
 |-----------------------------------------------------------------------------------------------
```

```
|Call summary - [2006/12/27 17:22:15.359]0000000024-0106927645
|                                                                  -------- Process time (second) -------
|  API       Success     Info  Warning     Fail    Total      Average      Minimum      Maximum
|  ------- -------- -------- -------- -------- -------- ------------ ------------ ------------
|  MQCONN          0        0        0        0        0
|  MQCONNX         1        0        0        0        1     0.141375     0.141375     0.141375
|  MQDISC          1        0        0        0        1     0.000177     0.000177     0.000177
|  MQOPEN          2        0        0        0        2     0.002971     0.000419     0.005523
|  MQCLOSE         2        0        0        0        2     0.001228     0.000124     0.002332
|  MQPUT1          0        0        0        0        0
|  MQPUT           1        0        0        0        1     0.158235     0.158235     0.158235
|  MQGET           0        0        0        0        0
|  MQINQ           1        0        0        0        1     0.000141     0.000141     0.000141
|  MQSET           0        0        0        0        0
|  MQBEGIN         0        0        0        0        0
|  MQCMIT          1        0        0        0        1     0.012846     0.012846     0.012846
|  MQBACK          1        0        0        0        1     0.000099     0.000099     0.000099
|  ------- -------- -------- -------- -------- -------- ------------ ------------ ------------
|  Total          10        0        0        0       10
|--------------------------------------------------------------------------------------------------
|
```

(API Exit) Sample output by Summary Data Generation Utility

Content of input file

```
qx_QM1_strmqm_192700_1_001.txt
qx_QM1_runmqchi_283068_1_001.txt
qx_QM1_amqpcsea_168138_1_001.txt
qx_QM1_amqzmur0_184544_5_001.txt
qx_QM1_amqzfuma_119018_1_001.txt
qx_QM1_amqzdmaa_221608_1_001.txt
qx_QM1_amqrrmfa_123112_1_001.txt
```

Output

```
|---------------------------------------------------------------------------------------------
|
|
|   WebSphere MQ API Exit (SupportPac MA0W)
|   Summary Data Generation Utility (64-bit version) Version 1.12
|
|     Input file : tracefiles.txt
|
|---------------------------------------------------------------------------------------------

1 qx_QM1_strmqm_192700_1_001.txt
2 qx_QM1_runmqchi_283068_1_001.txt
3 qx_QM1_amqpcsea_168138_1_001.txt
4 qx_QM1_amqzmur0_184544_5_001.txt
5 qx_QM1_amqzfuma_119018_1_001.txt
6 qx_QM1_amqzdmaa_221608_1_001.txt
7 qx_QM1_amqrrmfa_123112_1_001.txt

       Time                          API     CC Reason  Time(sec)  ObjectName                  MsgId      MsgLen
5    1 [2006/02/16 13:18:04.536758]  MQCONNX  0    0  0.007757
5    2 [2006/02/16 13:18:04.537732]  MQOPEN   0    0  0.000884 SYSTEM.AUTH.DATA.QUEUE
5    3 [2006/02/16 13:18:04.538128]  MQGET    0    0  0.000217 SYSTEM.AUTH.DATA.QUEUE           414D5120..        4
5    4 [2006/02/16 13:18:04.538535]  MQGET    0    0  0.000203 SYSTEM.AUTH.DATA.QUEUE           414D5120..       88
5    5 [2006/02/16 13:18:04.538963]  MQGET    0    0  0.000206 SYSTEM.AUTH.DATA.QUEUE           414D5120..       88
5    6 [2006/02/16 13:18:04.539384]  MQGET    0    0  0.000204 SYSTEM.AUTH.DATA.QUEUE           414D5120..       88
5    7 [2006/02/16 13:18:04.539805]  MQGET    0    0  0.000203 SYSTEM.AUTH.DATA.QUEUE           414D5120..       88
5    8 [2006/02/16 13:18:04.540224]  MQGET    0    0  0.000202 SYSTEM.AUTH.DATA.QUEUE           414D5120..       88
5    9 [2006/02/16 13:18:04.540629]  MQGET    0    0  0.000186 SYSTEM.AUTH.DATA.QUEUE           414D5120..       88
5   10 [2006/02/16 13:18:04.541065]  MQGET    0    0  0.000203 SYSTEM.AUTH.DATA.QUEUE           414D5120..       88
5   11 [2006/02/16 13:18:04.541483]  MQGET    0    0  0.000203 SYSTEM.AUTH.DATA.QUEUE           414D5120..       88
5   12 [2006/02/16 13:18:04.541898]  MQGET    0    0  0.000201 SYSTEM.AUTH.DATA.QUEUE           414D5120..       88
5   13 [2006/02/16 13:18:04.542315]  MQGET    0    0  0.000202 SYSTEM.AUTH.DATA.QUEUE           414D5120..       88
5   14 [2006/02/16 13:18:04.542732]  MQGET    0    0  0.000201 SYSTEM.AUTH.DATA.QUEUE           414D5120..       88
5   15 [2006/02/16 13:18:04.543210]  MQGET    0    0  0.000205 SYSTEM.AUTH.DATA.QUEUE           414D5120..       88
5   16 [2006/02/16 13:18:04.543639]  MQGET    0    0  0.000205 SYSTEM.AUTH.DATA.QUEUE           414D5120..       88
5   17 [2006/02/16 13:18:04.544082]  MQGET    0    0  0.000202 SYSTEM.AUTH.DATA.QUEUE           414D5120..       88
5   18 [2006/02/16 13:18:04.544528]  MQGET    0    0  0.000204 SYSTEM.AUTH.DATA.QUEUE           414D5120..       88
5   19 [2006/02/16 13:18:04.544972]  MQGET    0    0  0.000202 SYSTEM.AUTH.DATA.QUEUE           414D5120..       88
5   20 [2006/02/16 13:18:04.545742]  MQGET    0    0  0.000211 SYSTEM.AUTH.DATA.QUEUE           414D5120..       88
5   21 [2006/02/16 13:18:04.546185]  MQGET    0    0  0.000203 SYSTEM.AUTH.DATA.QUEUE           414D5120..       88
5   22 [2006/02/16 13:18:04.546612]  MQGET    0    0  0.000187 SYSTEM.AUTH.DATA.QUEUE           414D5120..       88
5   23 [2006/02/16 13:18:04.547100]  MQGET    0    0  0.000204 SYSTEM.AUTH.DATA.QUEUE           414D5120..       88
5   24 [2006/02/16 13:18:04.547541]  MQGET    0    0  0.000203 SYSTEM.AUTH.DATA.QUEUE           414D5120..       88
5   25 [2006/02/16 13:18:04.547985]  MQGET    0    0  0.000204 SYSTEM.AUTH.DATA.QUEUE           414D5120..       88
5   26 [2006/02/16 13:18:04.548430]  MQGET    0    0  0.000206 SYSTEM.AUTH.DATA.QUEUE           414D5120..       88
5   27 [2006/02/16 13:18:04.548875]  MQGET    0    0  0.000204 SYSTEM.AUTH.DATA.QUEUE           414D5120..       88
5   28 [2006/02/16 13:18:04.549319]  MQGET    0    0  0.000204 SYSTEM.AUTH.DATA.QUEUE           414D5120..       88
5   29 [2006/02/16 13:18:04.549747]  MQGET    0    0  0.000188 SYSTEM.AUTH.DATA.QUEUE           414D5120..       88
5   30 [2006/02/16 13:18:04.550187]  MQGET    0    0  0.000203 SYSTEM.AUTH.DATA.QUEUE           414D5120..       88
5   31 [2006/02/16 13:18:04.550647]  MQGET    0    0  0.000205 SYSTEM.AUTH.DATA.QUEUE           414D5120..       88
5   32 [2006/02/16 13:18:04.551093]  MQGET    0    0  0.000204 SYSTEM.AUTH.DATA.QUEUE           414D5120..       88
5   33 [2006/02/16 13:18:04.551539]  MQGET    0    0  0.000204 SYSTEM.AUTH.DATA.QUEUE           414D5120..       88
5   34 [2006/02/16 13:18:04.551984]  MQGET    0    0  0.000203 SYSTEM.AUTH.DATA.QUEUE           414D5120..       88
5   35 [2006/02/16 13:18:04.552412]  MQGET    0    0  0.000187 SYSTEM.AUTH.DATA.QUEUE           414D5120..       88
5   36 [2006/02/16 13:18:04.552856]  MQGET    0    0  0.000205 SYSTEM.AUTH.DATA.QUEUE           414D5120..       88
5   37 [2006/02/16 13:18:04.553600]  MQGET    0    0  0.000504 SYSTEM.AUTH.DATA.QUEUE           414D5120..       88
5   38 [2006/02/16 13:18:04.554053]  MQGET    0    0  0.000205 SYSTEM.AUTH.DATA.QUEUE           414D5120..       88
5   39 [2006/02/16 13:18:04.554498]  MQGET    0    0  0.000205 SYSTEM.AUTH.DATA.QUEUE           414D5120..       88
5   40 [2006/02/16 13:18:04.554944]  MQGET    0    0  0.000206 SYSTEM.AUTH.DATA.QUEUE           414D5120..       88
5   41 [2006/02/16 13:18:04.555404]  MQGET    0    0  0.000205 SYSTEM.AUTH.DATA.QUEUE           414D5120..       88
5   42 [2006/02/16 13:18:04.555850]  MQGET    0    0  0.000192 SYSTEM.AUTH.DATA.QUEUE           414D5120..       88
5   43 [2006/02/16 13:18:04.556295]  MQGET    0    0  0.000202 SYSTEM.AUTH.DATA.QUEUE           414D5120..       88
5   44 [2006/02/16 13:18:04.556740]  MQGET    0    0  0.000204 SYSTEM.AUTH.DATA.QUEUE           414D5120..       88
5   45 [2006/02/16 13:18:04.557185]  MQGET    0    0  0.000201 SYSTEM.AUTH.DATA.QUEUE           414D5120..       88
5   46 [2006/02/16 13:18:04.557629]  MQGET    0    0  0.000203 SYSTEM.AUTH.DATA.QUEUE           414D5120..       88
5   47 [2006/02/16 13:18:04.558069]  MQGET    0    0  0.000202 SYSTEM.AUTH.DATA.QUEUE           414D5120..       88
5   48 [2006/02/16 13:18:04.558511]  MQGET    0    0  0.000202 SYSTEM.AUTH.DATA.QUEUE           414D5120..       88
5   49 [2006/02/16 13:18:04.558936]  MQGET    0    0  0.000201 SYSTEM.AUTH.DATA.QUEUE           414D5120..       88
5   50 [2006/02/16 13:18:04.559377]  MQGET    0    0  0.000203 SYSTEM.AUTH.DATA.QUEUE           414D5120..       88
```

```
5  51 [2006/02/16 13:18:04.559818]  MQGET    0    0  0.000203 SYSTEM.AUTH.DATA.QUEUE           414D5120..     88
5  52 [2006/02/16 13:18:04.560258]  MQGET    0    0  0.000203 SYSTEM.AUTH.DATA.QUEUE           414D5120..     88
5  53 [2006/02/16 13:18:04.560714]  MQGET    0    0  0.000204 SYSTEM.AUTH.DATA.QUEUE           414D5120..     88
5  54 [2006/02/16 13:18:04.561389]  MQGET    0    0  0.000436 SYSTEM.AUTH.DATA.QUEUE           414D5120..     88
5  55 [2006/02/16 13:18:04.561825]  MQGET    2 2033  0.000187 SYSTEM.AUTH.DATA.QUEUE         (2033 = MQRC_NO_MSG_AVAILABLE)
7   1 [2006/02/16 13:18:04.599153]  MQCONNX  0    0  0.189519
7   2 [2006/02/16 13:18:04.599478]  MQOPEN   0    0  0.000226
7   3 [2006/02/16 13:18:04.599798]  MQINQ    0    0  0.000174
7   4 [2006/02/16 13:18:04.600077]  MQOPEN   0    0  0.000228 SYSTEM.DEFAULT.NAMELIST
7   5 [2006/02/16 13:18:04.600286]  MQCLOSE  0    0  0.000064 SYSTEM.DEFAULT.NAMELIST
7   6 [2006/02/16 13:18:04.600631]  MQOPEN   0    0  0.000317 SYSTEM.CLUSTER.COMMAND.QUEUE
7   7 [2006/02/16 13:18:04.600864]  MQINQ    0    0  0.000095 SYSTEM.CLUSTER.COMMAND.QUEUE
7   8 [2006/02/16 13:18:04.601373]  MQOPEN   0    0  0.000468 SYSTEM.CLUSTER.TRANSMIT.QUEUE
7   9 [2006/02/16 13:18:04.601770]  MQOPEN   0    0  0.000163 SYSTEM.CLUSTER.REPOSITORY.QUEUE
7  10 [2006/02/16 13:18:04.602245]  MQGET    0    0  0.000351 SYSTEM.CLUSTER.REPOSITORY.QUEUE  414D5120..  12320
7  11 [2006/02/16 13:18:04.603799]  MQOPEN   0    0  0.000162 SYSTEM.CLUSTER.TRANSMIT.QUEUE
7  12 [2006/02/16 13:18:04.604127]  MQGET    2 2033  0.000168 SYSTEM.CLUSTER.TRANSMIT.QUEUE  (2033 = MQRC_NO_MSG_AVAILABLE)
7  13 [2006/02/16 13:18:04.604386]  MQCLOSE  0    0  0.000071 SYSTEM.CLUSTER.TRANSMIT.QUEUE
7  14 [2006/02/16 13:18:04.604494]  MQINQ    0    0  0.000074
7  15 [2006/02/16 13:18:04.604787]  MQGET    2 2033  0.000254 SYSTEM.CLUSTER.COMMAND.QUEUE   (2033 = MQRC_NO_MSG_AVAILABLE)
7  16 [2006/02/16 13:18:04.605076]  MQOPEN   0    0  0.000146
7  17 [2006/02/16 13:18:04.605316]  MQINQ    0    0  0.000103
7  18 [2006/02/16 13:18:04.605711]  MQGET    1 2079  0.000285 SYSTEM.CLUSTER.REPOSITORY.QUEUE  414D5120..  12320 (2079 = MQ
7  19 [2006/02/16 13:18:04.621983]  MQPUT    0    0  0.016061 SYSTEM.CLUSTER.REPOSITORY.QUEUE  414D5120..  12320
7  20 [2006/02/16 13:18:04.622280]  MQCLOSE  0    0  0.000067
4   1 [2006/02/16 13:18:04.630211]  MQCONNX  0    0  0.214731
6   1 [2006/02/16 13:18:04.630305]  MQCONNX  0    0  0.208838
2   1 [2006/02/16 13:18:04.630461]  MQCONNX  0    0  0.005832
4   2 [2006/02/16 13:18:04.630497]  MQOPEN   0    0  0.000180
4   3 [2006/02/16 13:18:04.630746]  MQINQ    0    0  0.000138
6   2 [2006/02/16 13:18:04.630885]  MQOPEN   0    0  0.000482 SYSTEM.PENDING.DATA.QUEUE
4   4 [2006/02/16 13:18:04.630925]  MQINQ    0    0  0.000074
6   3 [2006/02/16 13:18:04.631266]  MQGET    2 2033  0.000259 SYSTEM.PENDING.DATA.QUEUE      (2033 = MQRC_NO_MSG_AVAILABLE)
6   4 [2006/02/16 13:18:04.631545]  MQCMIT   0    0  0.000095
2   2 [2006/02/16 13:18:04.631701]  MQOPEN   0    0  0.000403 SYSTEM.CHANNEL.INITQ
7  21 [2006/02/16 13:18:04.638765]  MQCMIT   0    0  0.013679
7  22 [2006/02/16 13:18:04.638926]  MQINQ    0    0  0.000111
7  23 [2006/02/16 13:18:04.639283]  MQCMIT   0    0  0.000049
2   3 [2006/02/16 13:18:04.653712]  MQSET    0    0  0.021859 SYSTEM.CHANNEL.INITQ
3   1 [2006/02/16 13:18:04.663083]  MQCONNX  0    0  0.002620
3   2 [2006/02/16 13:18:04.663999]  MQOPEN   0    0  0.000379 SYSTEM.ADMIN.COMMAND.QUEUE
3   3 [2006/02/16 13:18:04.664243]  MQINQ    0    0  0.000107 SYSTEM.ADMIN.COMMAND.QUEUE
3   4 [2006/02/16 13:18:04.689785]  MQINQ    0    0  0.000080 SYSTEM.ADMIN.COMMAND.QUEUE
1   1 [2006/02/16 13:18:04.943227]  MQCONNX  0    0  0.001845
1   2 [2006/02/16 13:18:04.943441]  MQDISC   0    0  0.000097
3   5 [2006/02/16 13:18:13.377029]  MQGET    2 2161  8.687199 SYSTEM.ADMIN.COMMAND.QUEUE     (2161 = MQRC_Q_MGR_QUIESCING)
2   4 [2006/02/16 13:18:13.377029]  MQGET    2 2161  8.721121 SYSTEM.CHANNEL.INITQ           (2161 = MQRC_Q_MGR_QUIESCING)
2   5 [2006/02/16 13:18:13.377509]  MQCLOSE  0    0  0.000137 SYSTEM.CHANNEL.INITQ
2   6 [2006/02/16 13:18:13.386671]  MQDISC   0    0  0.000094
3   6 [2006/02/16 13:18:13.421828]  MQCLOSE  0    0  0.000122 SYSTEM.ADMIN.COMMAND.QUEUE
3   7 [2006/02/16 13:18:13.422028]  MQDISC   0    0  0.000096
4   5 [2006/02/16 13:18:23.603075]  MQOPEN   0    0  0.000422 SYSTEM.ADMIN.STATISTICS.QUEUE
4   6 [2006/02/16 13:18:23.603322]  MQCLOSE  0    0  0.000074 SYSTEM.ADMIN.STATISTICS.QUEUE
4   7 [2006/02/16 13:18:23.603401]  MQCLOSE  0    0  0.000051
4   8 [2006/02/16 13:18:23.603498]  MQDISC   0    0  0.000067
6   5 [2006/02/16 13:18:23.713752]  MQBACK   0    0  0.000231
7  24 [2006/02/16 13:18:23.713795]  MQGET    0    0 19.074485 SYSTEM.CLUSTER.COMMAND.QUEUE     414D5120..     36
5  56 [2006/02/16 13:18:23.714031]  MQCLOSE  0    0  0.000098 SYSTEM.AUTH.DATA.QUEUE
6   6 [2006/02/16 13:18:23.714094]  MQDISC   0    0  0.000277
7  25 [2006/02/16 13:18:23.714217]  MQCMIT   0    0  0.000139
5  57 [2006/02/16 13:18:23.714385]  MQDISC   0    0  0.000318
7  26 [2006/02/16 13:18:23.714533]  MQGET    2 2033  0.000244 SYSTEM.CLUSTER.REPOSITORY.QUEUE (2033 = MQRC_NO_MSG_AVAILABLE)
7  27 [2006/02/16 13:18:23.714869]  MQGET    2 2033  0.000190 SYSTEM.CLUSTER.REPOSITORY.QUEUE (2033 = MQRC_NO_MSG_AVAILABLE)
7  28 [2006/02/16 13:18:23.715111]  MQCMIT   0    0  0.000052
7  29 [2006/02/16 13:18:23.715231]  MQCLOSE  0    0  0.000093 SYSTEM.CLUSTER.REPOSITORY.QUEUE
7  30 [2006/02/16 13:18:23.715375]  MQCLOSE  0    0  0.000067 SYSTEM.CLUSTER.COMMAND.QUEUE
7  31 [2006/02/16 13:18:23.715506]  MQDISC   0    0  0.000092

|-----------------------------------------------------------------------------------------------
|                                                               -------- Process time (second) -------
|  API      Success     Info  Warning     Fail    Total     Average       Minimum       Maximum
|  ------- -------- -------- -------- -------- -------- ------------ ------------ ------------
|  MQCONN         0        0        0        0        0
|  MQCONNX        7        0        0        0        7     0.090163     0.001845     0.214731
|  MQDISC         7        0        0        0        7     0.000149     0.000067     0.000318
|  MQOPEN        13        0        0        0       13     0.000343     0.000146     0.000884
|  MQCLOSE       10        0        0        0       10     0.000084     0.000051     0.000137
|  MQPUT1         0        0        0        0        0
|  MQPUT          1        0        0        0        1     0.016061     0.016061     0.016061
|  MQGET         54        8        1        0       63     0.579298     0.000168    19.074485
|  MQINQ          9        0        0        0        9     0.000106     0.000074     0.000174
|  MQSET          1        0        0        0        1     0.021859     0.021859     0.021859
|  MQBEGIN        0        0        0        0        0
|  MQCMIT         5        0        0        0        5     0.002803     0.000049     0.013679
|  MQBACK         1        0        0        0        1     0.000231     0.000231     0.000231
|  ------- -------- -------- -------- -------- -------- ------------ ------------ ------------
|  Total        108        8        1        0      117
|-----------------------------------------------------------------------------------------------
```

```
 MQOPEN
                                                                     ----Process time (second)----
Queue                           Success   Info Warning   Fail  Total   Average   Minimum   Maximum
------------------------------- ------- ------ ------- ------ ------ --------- --------- ---------
(null)                                3      0       0      0      3  0.000184  0.000146  0.000226
SYSTEM.ADMIN.COMMAND.QUEUE            1      0       0      0      1  0.000379  0.000379  0.000379
SYSTEM.ADMIN.STATISTICS.QUEUE         1      0       0      0      1  0.000422  0.000422  0.000422
SYSTEM.AUTH.DATA.QUEUE                1      0       0      0      1  0.000884  0.000884  0.000884
SYSTEM.CHANNEL.INITQ                  1      0       0      0      1  0.000403  0.000403  0.000403
SYSTEM.CLUSTER.COMMAND.QUEUE          1      0       0      0      1  0.000317  0.000317  0.000317
SYSTEM.CLUSTER.REPOSITORY.QUEUE       1      0       0      0      1  0.000163  0.000163  0.000163
SYSTEM.CLUSTER.TRANSMIT.QUEUE         2      0       0      0      2  0.000315  0.000162  0.000468
SYSTEM.DEFAULT.NAMELIST               1      0       0      0      1  0.000228  0.000228  0.000228
SYSTEM.PENDING.DATA.QUEUE             1      0       0      0      1  0.000482  0.000482  0.000482
------------------------------- ------- ------ ------- ------ ------ --------- --------- ---------
All                                  13      0       0      0     13  0.000343  0.000146  0.000884
```

```
 MQCLOSE
                                                                     ----Process time (second)----
Queue                           Success   Info Warning   Fail  Total   Average   Minimum   Maximum
------------------------------- ------- ------ ------- ------ ------ --------- --------- ---------
(null)                                2      0       0      0      2  0.000059  0.000051  0.000067
SYSTEM.ADMIN.COMMAND.QUEUE            1      0       0      0      1  0.000122  0.000122  0.000122
SYSTEM.ADMIN.STATISTICS.QUEUE         1      0       0      0      1  0.000074  0.000074  0.000074
SYSTEM.AUTH.DATA.QUEUE                1      0       0      0      1  0.000098  0.000098  0.000098
SYSTEM.CHANNEL.INITQ                  1      0       0      0      1  0.000137  0.000137  0.000137
SYSTEM.CLUSTER.COMMAND.QUEUE          1      0       0      0      1  0.000067  0.000067  0.000067
SYSTEM.CLUSTER.REPOSITORY.QUEUE       1      0       0      0      1  0.000093  0.000093  0.000093
SYSTEM.CLUSTER.TRANSMIT.QUEUE         1      0       0      0      1  0.000071  0.000071  0.000071
SYSTEM.DEFAULT.NAMELIST               1      0       0      0      1  0.000064  0.000064  0.000064
SYSTEM.PENDING.DATA.QUEUE             0      0       0      0      0
------------------------------- ------- ------ ------- ------ ------ --------- --------- ---------
All                                  10      0       0      0     10  0.000084  0.000051  0.000137
```

```
 MQPUT1
                                                                     ----Process time (second)---- -Message length (byte)-
Queue                           Success   Info Warning   Fail  Total   Average   Minimum   Maximum Average Minimum Maximum
------------------------------- ------- ------ ------- ------ ------ --------- --------- --------- ------- ------- -------
(null)                                0      0       0      0      0
SYSTEM.ADMIN.COMMAND.QUEUE            0      0       0      0      0
SYSTEM.ADMIN.STATISTICS.QUEUE         0      0       0      0      0
SYSTEM.AUTH.DATA.QUEUE                0      0       0      0      0
SYSTEM.CHANNEL.INITQ                  0      0       0      0      0
SYSTEM.CLUSTER.COMMAND.QUEUE          0      0       0      0      0
SYSTEM.CLUSTER.REPOSITORY.QUEUE       0      0       0      0      0
SYSTEM.CLUSTER.TRANSMIT.QUEUE         0      0       0      0      0
SYSTEM.DEFAULT.NAMELIST               0      0       0      0      0
SYSTEM.PENDING.DATA.QUEUE             0      0       0      0      0
------------------------------- ------- ------ ------- ------ ------ --------- --------- --------- ------- ------- -------
All                                   0      0       0      0      0
```

```
 MQPUT
                                                                     ----Process time (second)---- -Message length (byte)-
Queue                           Success   Info Warning   Fail  Total   Average   Minimum   Maximum Average Minimum Maximum
------------------------------- ------- ------ ------- ------ ------ --------- --------- --------- ------- ------- -------
(null)                                0      0       0      0      0
SYSTEM.ADMIN.COMMAND.QUEUE            0      0       0      0      0
SYSTEM.ADMIN.STATISTICS.QUEUE         0      0       0      0      0
SYSTEM.AUTH.DATA.QUEUE                0      0       0      0      0
SYSTEM.CHANNEL.INITQ                  0      0       0      0      0
SYSTEM.CLUSTER.COMMAND.QUEUE          0      0       0      0      0
SYSTEM.CLUSTER.REPOSITORY.QUEUE       1      0       0      0      1  0.016061  0.016061  0.016061   12320   12320   12320
SYSTEM.CLUSTER.TRANSMIT.QUEUE         0      0       0      0      0
SYSTEM.DEFAULT.NAMELIST               0      0       0      0      0
SYSTEM.PENDING.DATA.QUEUE             0      0       0      0      0
------------------------------- ------- ------ ------- ------ ------ --------- --------- --------- ------- ------- -------
All                                   1      0       0      0      1  0.016061  0.016061  0.016061   12320   12320   12320
```

```
 MQGET
                                                                     ----Process time (second)---- -Message length (byte)-
Queue                           Success   Info Warning   Fail  Total   Average   Minimum   Maximum Average Minimum Maximum
------------------------------- ------- ------ ------- ------ ------ --------- --------- --------- ------- ------- -------
(null)                                0      0       0      0      0
SYSTEM.ADMIN.COMMAND.QUEUE            0      1       0      0      1  8.687199  8.687199  8.687199
SYSTEM.ADMIN.STATISTICS.QUEUE         0      0       0      0      0
SYSTEM.AUTH.DATA.QUEUE               52      1       0      0     53  0.000212  0.000186  0.000504      86       4      88
SYSTEM.CHANNEL.INITQ                  0      1       0      0      1  8.721121  8.721121  8.721121
SYSTEM.CLUSTER.COMMAND.QUEUE          1      1       0      0      2  9.537370  0.000254 19.074485      36      36      36
SYSTEM.CLUSTER.REPOSITORY.QUEUE       1      2       1      0      4  0.000267  0.000190  0.000351   12320   12320   12320
SYSTEM.CLUSTER.TRANSMIT.QUEUE         0      1       0      0      1  0.000168  0.000168  0.000168
SYSTEM.DEFAULT.NAMELIST               0      0       0      0      0
SYSTEM.PENDING.DATA.QUEUE             0      1       0      0      1  0.000259  0.000259  0.000259
------------------------------- ------- ------ ------- ------ ------ --------- --------- --------- ------- ------- -------
All                                  54      8       1      0     63  0.579298  0.000168 19.074485     530       4   12320
```

```
  MQINQ
                                                                          ----Process time (second)----
Queue                           Success  Info Warning  Fail  Total  Average   Minimum   Maximum
------------------------------- ------ ------ ------ ------ ------ --------- --------- ---------
(null)                               6      0      0      0      6  0.000112  0.000074  0.000174
SYSTEM.ADMIN.COMMAND.QUEUE           2      0      0      0      2  0.000093  0.000080  0.000107
SYSTEM.ADMIN.STATISTICS.QUEUE        0      0      0      0      0
SYSTEM.AUTH.DATA.QUEUE               0      0      0      0      0
SYSTEM.CHANNEL.INITQ                 0      0      0      0      0
SYSTEM.CLUSTER.COMMAND.QUEUE         1      0      0      0      1  0.000095  0.000095  0.000095
SYSTEM.CLUSTER.REPOSITORY.QUEUE      0      0      0      0      0
SYSTEM.CLUSTER.TRANSMIT.QUEUE        0      0      0      0      0
SYSTEM.DEFAULT.NAMELIST              0      0      0      0      0
SYSTEM.PENDING.DATA.QUEUE            0      0      0      0      0
------------------------------- ------ ------ ------ ------ ------ --------- --------- ---------
All                                  9      0      0      0      9  0.000106  0.000074  0.000174
```

```
  MQSET
                                                                          ----Process time (second)----
Queue                           Success  Info Warning  Fail  Total  Average   Minimum   Maximum
------------------------------- ------ ------ ------ ------ ------ --------- --------- ---------
(null)                               0      0      0      0      0
SYSTEM.ADMIN.COMMAND.QUEUE           0      0      0      0      0
SYSTEM.ADMIN.STATISTICS.QUEUE        0      0      0      0      0
SYSTEM.AUTH.DATA.QUEUE               0      0      0      0      0
SYSTEM.CHANNEL.INITQ                 1      0      0      0      1  0.021859  0.021859  0.021859
SYSTEM.CLUSTER.COMMAND.QUEUE         0      0      0      0      0
SYSTEM.CLUSTER.REPOSITORY.QUEUE      0      0      0      0      0
SYSTEM.CLUSTER.TRANSMIT.QUEUE        0      0      0      0      0
SYSTEM.DEFAULT.NAMELIST              0      0      0      0      0
SYSTEM.PENDING.DATA.QUEUE            0      0      0      0      0
------------------------------- ------ ------ ------ ------ ------ --------- --------- ---------
All                                  1      0      0      0      1  0.021859  0.021859  0.021859
```

(Channel Exit – MQI) with ShowSummaryOnly=yes parameter

```
|----------------------------------------------------------
|Program information
|  Program name       : WEBSPHERE MQ Channel Exit (SupportPac MA0W)
|  Version            : 5.50
|  Provided by        : oinoue@jp.ibm.com
|Module attributes
|  Addressing mode    : 64-bit
|  Threading          : Multi thread (_r)
|Operating environment
|  Operating system   : AIX
|  Version            : 5
|  Release            : 3
|  Node name          : saints
|  Machine            : 00237E8A4C00
|API Exit context
|  Exit name          : qxEntry
|  API caller type    :
|  Queue manager      : QM1
|  Environment        :
|  User id            :
|  Connection name    : 127.0.0.1
|  Appl/Channel name  : CH1EXIT
|  Appl/Channel type  : SVRCONN
|  Program name       : channel_SVRCONN_CH1EXIT
|  Program description:
|  Process id         : 373070
|  Thread id          : 776
|  Process id (real)  : 373070
|  Thread id  (real)  : 776
|Process parameters
|  Parameter file     : /home/apiexit/tracedata/qx.def
|  Show max MQPUT len : (all)
|  Show max MQGET len : (all)
|  Show max chl len   : (all)
|  Show data conv     : no
|  Show summary only  : yes
|  Show parsed data   : yes
|  Show parsed XML    : yes
|  Show hex char image: yes
|  Show hex char ccsid: 0
|  Show perf only     : no
|  Selected call only : no
|  Alert file (error) : yes
|  Alert file (warn)  : no
|  Perf intvl (time)  : 0
|  Perf intvl (count) : 0
|  Max file size      : 100,000,000
|  Max XML line length: 100
|  Compress command   : (none)
|  Compress threshold : 1,000,000
|  Create empty file  : yes
|  Write stream mode  : no
|  Write cache        : yes
|  Write after call   : yes
|  Write buffer size  : 50,000
|  Write interval     : 30
|  Check update req   : 30
|  Suppress alert     : 2, 2085, MQOPEN , SYSTEM.BROKER.INTER.BROKER.COMMUNICATIONS, DataFlowEngine
|  Suppress alert     : 2, 2038, MQINQ  , SYSTEM.BROKER.ADMIN.REPLY, DataFlowEngine
|   (Debug flag)      : off
|----------------------------------------------------------
-
[2006/05/11 17:17:45.917734]
<INIT
(CHANNEL_SEND_EXIT
-
[2006/05/11 17:17:45.917791]
<INIT
(CHANNEL_RCV_EXIT
-


       Time                          API                  Segmentation DataLen CC Reason MsgId        Queue  Queue manager
     1 [2006/05/11 17:17:45.918532]  (N/A)                (others)         160  -    -
     2 [2006/05/11 17:17:45.918621]  MQCONN request       none             304  -    -
     3 [2006/05/11 17:17:45.918761]  MQCONN reply         none             304  0    0
     4 [2006/05/11 17:17:45.918943]  MQOPEN request       none             216  -    -    -              QM2Q1
     5 [2006/05/11 17:17:45.920167]  MQOPEN reply         none             216  0    0    -              QM2Q1
     6 [2006/05/11 17:17:45.920512]  MQPUT request        first          32766  -    -    -              QM2Q1
     7 [2006/05/11 17:17:45.920601]  MQPUT request        middle         32766  -    -
     8 [2006/05/11 17:17:45.920671]  MQPUT request        middle         32766  -    -
```

```
       9 [2006/05/11 17:17:45.920752]  MQPUT request         middle        32766  -    -
      10 [2006/05/11 17:17:45.920819]  MQPUT request         middle        32766  -    -
      11 [2006/05/11 17:17:45.920899]  MQPUT request         middle        32766  -    -

            ........

    1529 [2006/05/11 17:17:46.044886]  MQPUT request         middle        32766  -    -
    1530 [2006/05/11 17:17:46.044976]  MQPUT request         middle        32766  -    -
    1531 [2006/05/11 17:17:46.045043]  MQPUT request         middle        32766  -    -
    1532 [2006/05/11 17:17:46.045126]  MQPUT request         middle        32766  -    -
    1533 [2006/05/11 17:17:46.045189]  MQPUT request         last           9614  -    -
    1534 [2006/05/11 17:17:48.954229]  MQPUT reply           none            540  0    0    414D512051..  QM2Q1
    1535 [2006/05/11 17:17:48.954599]  MQCLOSE request       none             48  -    -    -             QM2Q1
    1536 [2006/05/11 17:17:48.954794]  MQCLOSE reply         none             44  0    0    -             QM2Q1
    1537 [2006/05/11 17:17:48.954930]  MQDISC request        none             44  -    -
    1538 [2006/05/11 17:17:48.961715]  MQDISC reply          none             44  0    0
    1539 [2006/05/11 17:17:48.961805]  (N/A)                 (others)         28  -    -
-
[2006/05/11 17:17:48.962065]
<TERM
(CHANNEL_SEND_EXIT
-
[2006/05/11 17:17:48.962079]
<TERM
(CHANNEL_RCV_EXIT
-
|-------------------------------------------------------------------------------------------
|Call summary - [2006/05/11 17:17:48.962095]
|                                                              ---- Data length (byte) -----
|  API               Success      Info  Warning     Fail    Total   Average   Minimum   Maximum
|  --------------- -------- -------- -------- -------- -------- --------- --------- ---------
|  MQCONN  request                                             1       304       304       304
|  MQCONN  reply           1        0        0        0        1       304       304       304
|  MQDISC  request                                             1        44        44        44
|  MQDISC  reply           1        0        0        0        1        44        44        44
|  MQOPEN  request                                             1       216       216       216
|  MQOPEN  reply           1        0        0        0        1       216       216       216
|  MQCLOSE request                                             1        48        48        48
|  MQCLOSE reply           1        0        0        0        1        44        44        44
|  MQPUT   request                                          1528     32751      9614     32766
|  MQPUT   reply           1        0        0        0        1       540       540       540
|  MQPUT1  request                                             0
|  MQPUT1  reply           0        0        0        0        0
|  MQGET   request                                             0
|  MQGET   reply           0        0        0        0        0
|  MQINQ   request                                             0
|  MQINQ   reply           0        0        0        0        0
|  MQSET   request                                             0
|  MQSET   reply           0        0        0        0        0
|  MQCMIT  request                                             0
|  MQCMIT  reply           0        0        0        0        0
|  MQBACK  request                                             0
|  MQBACK  reply           0        0        0        0        0
|  XA      request                                             0
|  XA      reply           0        0        0        0        0
|  XMIT    send                                                0
|  XMIT    receive                                             2        94        28       160
|  MSG                                                         0
|  others                                                      0
|  --------------- -------- -------- -------- -------- -------- --------- --------- ---------
|  Total                   5        0        0        0     1539
|-------------------------------------------------------------------------------------------
|
```

(Channel Exit – MQI) with default parameter values

```
|-----------------------------------------------------------
|Program information
|  Program name      : WebSphere MQ Channel Exit (SupportPac MA0W)
|  Version           : 6.01
|  Provided by       : oinoue@jp.ibm.com
|Module attributes
|  Addressing mode   : 32-bit
|  Threading         : Common to single thread and multi thread
|Operating environment
|  Operating system  : Windows XP
|  Major version     : 5
|  Minor version     : 1
|  Service level     : Service Pack 2
|  Build number      : 2600
|  Computer name     : OINOUEM
|API Exit context
|  Exit name         : qxEntry
|  API caller type   :
|  Queue manager     : QM1
|  Environment       :
|  User id           :
|  Connection name   :
|  Appl/Channel name  : CH1
|  Appl/Channel type  : SVRCONN
|  Program name      : channel_SVRCONN_CH1
|  Program description:
|  Process id        : 2468
|  Thread id         : 2828
|  Process id (real)  : 2468
|  Thread id  (real)  : 2828
|Process parameters
|  Parameter file    : e:¥cpp¥apiexit¥tracedata¥qx.def
|  Check update req  : 30
|  Show summary only  : no
|  Show perf only    : no
|  Perf intvl (time)  : 0
|  Perf intvl (count) : 0
|  Show parsed data   : yes
|  Show parsed content: yes
|  Max out line length: 100
|  Show hex char image: yes
|  Show hex char ccsid: 0
|  Show data conv     : no
|  Show max len MQPUT : (all)
|  Show max len MQGET : (all)
|  Show max len chl   : (all)
|  Show max len parsed: (all)
|  Count only selected: no
|  Alert file (error) : yes
|  Alert file (warn)  : no
|  Suppress alert     : 2, 2085, MQOPEN , SYSTEM.BROKER.INTER.BROKER.COMMUNICATIONS, DataFlowEngine
|  Suppress alert     : 2, 2038, MQINQ  , SYSTEM.BROKER.ADMIN.REPLY, DataFlowEngine
|  Max file size      : 100,000,000
|  Compress command   : (none)
|  Compress threshold : 1,000,000
|  Create empty file  : yes
|  Write stream mode  : no
|  Write cache        : yes
|  Write after call   : yes
|  Write buffer size  : 10,000
|  Write interval     : 30
|  (Debug flag)       : off
|-----------------------------------------------------------
-
[2006/12/27 17:22:15.078]0000000024-0105881895
<INIT
(CHANNEL_SEND_EXIT
-
[2006/12/27 17:22:15.078]0000000024-0105886008
<INIT
(CHANNEL_RCV_EXIT
-
|--- Seq # : 1 ---------------------------------------
[2006/12/27 17:22:15.078]0000000024-0105896802
<XMIT
(CHANNEL_RCV_EXIT
DataLength        : 160 (=0xA0)
AgentBufferLength : 32766 (=0x7FFE)
API call type     : (N/A) (=0x08)
Segmentation      : (others) (=0x01)
AgentBuffer-TSH  offset 0-27(=0x0-0x1B):(28=0x1C bytes, Address 00AB2D74)
00000000:54534820 000000A0 02080100 00000000  [TSH ............]
00000010:00000000 22020000 A4030000           [....″.......]
```

```
AgentBuffer- offset 28-159(=0x1C-0x9F):(132=0x84 bytes, Address 00AB2D90)
00000000:55494420 4F494E4F 55452020 20202020  [UID OINOUE      ]
00000010:20202020 20202020 20202020 6F696E6F  [            oino]
00000020:75652020 20202020 20202020 20202020  [ue              ]
00000030:20202020 20202020 20202020 20202020  [                ]
00000040-0000004F same as above
00000050:20202020 20202020 20202020 1D010105  [            ....]
00000060:00000000 00051500 0000BE04 3E324B2C  [............>2K,]
00000070:BC1A828B A628EB03 00000000 00000000  [.....(..........]
00000080:00000000                             [....]
-
|--- Seq # : 2 ---------------------------------------
[2006/12/27 17:22:15.125]0000000024-0106062357
<XMIT
(CHANNEL_RCV_EXIT
DataLength        : 304 (=0x130)
AgentBufferLength : 32766 (=0x7FFE)
API call type     : MQCONN request (=0x81)
Segmentation      : none (=0x30)
AgentBuffer-TSH  offset 0-43(=0x0-0x2B):(44=0x2C bytes, Address 00AB2D74)
00000000:54534820 00000130 02813000 00000000  [TSH ...0..0.....]
00000010:00000000 22020000 A4030000 00000130  [....″..........0]
00000020:00000000 00000000 00000000           [............]
AgentBuffer- offset 44-303(=0x2C-0x12F):(260=0x104 bytes, Address 00AB2DA0)
00000000:514D3120 20202020 20202020 20202020  [QM1             ]
00000010:20202020 20202020 20202020 20202020  [                ]
00000020-0000002F same as above
00000030:756E5C69 73656C5F 6D716C6F 6E677275  [un¥isel_mqlongru]
00000040:6E5F6D71 69633332 2E657865 0B000000  [n_mqic32.exe....]
00000050:16010515 000000BE 043E324B 2CBC1A82  [.........>2K,...]
00000060:8BA628EB 03000000 00000000 0000000B  [..(.............]
00000070:03000000 00000000 46434E4F 01000000  [........FCNO....]
00000080:00000000 00000000 00000000 00000000  [................]
00000090-000000FF same as above
00000100:00000000                             [....]
-
|--- Seq # : 3 ---------------------------------------
[2006/12/27 17:22:15.125]0000000024-0106064185
<XMIT
(CHANNEL_SEND_EXIT
DataLength        : 304 (=0x130)
AgentBufferLength : 4000 (=0xFA0)
API call type     : MQCONN reply (=0x91)
Segmentation      : none (=0x30)
CompCode          : 0
Reason            : 0
AgentBuffer-TSH  offset 0-43(=0x0-0x2B):(44=0x2C bytes, Address 00AB2D74)
00000000:54534820 30010000 02913000 00000000  [TSH 0.....0.....]
00000010:00000000 22020000 A4030000 00000130  [....″..........0]
00000020:00000000 00000000 00000000           [............]
AgentBuffer- offset 44-303(=0x2C-0x12F):(260=0x104 bytes, Address 00AB2DA0)
00000000:514D3120 20202020 20202020 20202020  [QM1             ]
00000010:20202020 20202020 20202020 20202020  [                ]
00000020-0000002F same as above
00000030:756E5C69 73656C5F 6D716C6F 6E677275  [un¥isel_mqlongru]
00000040:6E5F6D71 69633332 2E657865 0B000000  [n_mqic32.exe....]
00000050:16010515 000000BE 043E324B 2CBC1A82  [.........>2K,...]
00000060:8BA628EB 03000000 00000000 0000000B  [..(.............]
00000070:03000000 00000000 46434E4F 01000000  [........FCNO....]
00000080:00000000 414D5143 514D3120 20202020  [....AMQCQM1     ]
00000090:20202020 E32C9245 20000901 00000000  [    .,.E .......]
000000A0:00000000 00000000 00000000 00000000  [................]
000000B0-000000FF same as above
00000100:00000000                             [....]
-
|--- Seq # : 4 ---------------------------------------
[2006/12/27 17:22:15.125]0000000024-0106066715
<XMIT
(CHANNEL_RCV_EXIT
DataLength        : 216 (=0xD8)
AgentBufferLength : 32766 (=0x7FFE)
API call type     : MQOPEN request (=0x83)
Segmentation      : none (=0x30)
ObjectName        : Q1
AgentBuffer-TSH  offset 0-43(=0x0-0x2B):(44=0x2C bytes, Address 00AB2D74)
00000000:54534820 000000D8 02833000 00000000  [TSH ......0.....]
00000010:00000000 22020000 A4030000 000000D8  [....″...........]
00000020:00000000 00000000 00000000           [............]
AgentBuffer-OD   offset 44-211(=0x2C-0xD3):(168=0xA8 bytes, Address 00AB2DA0)
00000000:4F442020 01000000 01000000 51310000  [OD  ........Q1..]
00000010:00000000 00000000 00000000 00000000  [................]
00000020-0000005F same as above
00000060:00000000 00000000 00000000 414D512E  [............AMQ.]
00000070:2A000000 00000000 00000000 00000000  [*...............]
00000080:00000000 00000000 00000000 00000000  [................]
00000090-0000009F same as above
000000A0:00000000 00000000                    [........]
 |> Version               : 1
 |> ObjectType            : QUEUE
 |> ObjectName            : Q1
 |> ObjectQMgrName           :
 |> DynamicQName             : AMQ.*
 |> AlternateUserId          :
AgentBuffer- offset 212-215(=0xD4-0xD7):(4=0x4 bytes, Address 00AB2E48)
00000000:10200000                             [. ..]
-
|--- Seq # : 5 ----------------------------------------
[2006/12/27 17:22:15.125]0000000024-0106087678
<XMIT
(CHANNEL_SEND_EXIT
DataLength        : 216 (=0xD8)
AgentBufferLength : 4000 (=0xFA0)
API call type     : MQOPEN reply (=0x93)
Segmentation      : none (=0x30)
ObjectName        : Q1
CompCode          : 0
Reason            : 0
AgentBuffer-TSH  offset 0-43(=0x0-0x2B):(44=0x2C bytes, Address 00AB2D74)
00000000:54534820 D8000000 02933000 00000000  [TSH ......0.....]
00000010:00000000 22020000 A4030000 000000D8  [....″...........]
00000020:00000000 00000000 F0FB3E00           [..........>.]
AgentBuffer-OD   offset 44-211(=0x2C-0xD3):(168=0xA8 bytes, Address 00AB2DA0)
00000000:4F442020 01000000 01000000 51310000  [OD  ........Q1..]
00000010:00000000 00000000 00000000 00000000  [................]
00000020-0000005F same as above
00000060:00000000 00000000 00000000 414D512E  [............AMQ.]
00000070:2A000000 00000000 00000000 00000000  [*...............]
00000080:00000000 00000000 00000000 00000000  [................]
00000090-0000009F same as above
000000A0:00000000 00000000                    [........]
 |> Version                  : 1
 |> ObjectType               : QUEUE
 |> ObjectName               : Q1
 |> ObjectQMgrName           :
 |> DynamicQName             : AMQ.*
 |> AlternateUserId          :
AgentBuffer- offset 212-215(=0xD4-0xD7):(4=0x4 bytes, Address 00AB2E48)
00000000:10200000                             [. ..]
-
|--- Seq # : 6 ----------------------------------------
[2006/12/27 17:22:15.125]0000000024-0106090620
<XMIT
(CHANNEL_RCV_EXIT
DataLength        : 738 (=0x2E2)
AgentBufferLength : 32766 (=0x7FFE)
API call type     : MQPUT request (=0x86)
Segmentation      : none (=0x30)
ObjectName        : Q1
AgentBuffer-TSH  offset 0-43(=0x0-0x2B):(44=0x2C bytes, Address 00AB2D74)
00000000:54534820 000002E2 02863000 00000000  [TSH ......0.....]
00000010:00000000 22020000 A4030000 000002E2  [....″...........]
00000020:00000000 00000000 F0FB3E00           [..........>.]
AgentBuffer-MD   offset 44-407(=0x2C-0x197):(364=0x16C bytes, Address 00AB2DA0)
00000000:4D442020 02000000 00000000 08000000  [MD  ............]
00000010:FFFFFFFF 00000000 22020000 AF030000  [........″.......]
00000020:4D514852 46322020 00000000 01000000  [MQHRF2  ........]
00000030:00000000 00000000 00000000 00000000  [................]
00000040-0000005F same as above
00000060:00000000 51524550 4C592020 20202020  [....QREPLY      ]
00000070:20202020 20202020 20202020 20202020  [                ]
00000080-0000008F same as above
00000090:20202020 00000000 00000000 00000000  [    ............]
000000A0:00000000 00000000 00000000 00000000  [................]
000000B0-0000014F same as above
00000150:00000000 00000000 00000000 01000000  [................]
00000160:00000000 00000000 FFFFFFFF           [............]
 |> Version                  : 2
 |> Report                   : (none)
 |> MsgType                  : DATAGRAM
 |> Expiry                   : -1
 |> Feedback                 : 0
 |> Encoding                 : 546 (Native encoding) INTEGER_REVERSED / DECIMAL_REVERSED / FLOAT_IEEE_REVERSED
 |> CodedCharSetId           : 943
 |> Format                   : MQHRF2
 |> Priority                 : 0
 |> Persistence              : PERSISTENT
 |> MsgId                    : (null)
 |> CorrelId                 : (null)
 |> BackoutCount             : 0
 |> ReplyToQ                 : QREPLY
 |> ReplyToQMgr              :
 |> UserIdentifier           :
 |> AccountingToken          : (null)
 |> ApplIdentityData         :
 |> PutApplType              : no context
 |> PutApplName              :
 |> PutDate                  :
 |> PutTime                  :
 |> ApplOriginData           :
 |> --- MD version 2----------
 |> GroupId                  : (null)
 |> MsgSeqNumber             : 1
 |> Offset                   : 0
 |> MsgFlags                 : (none)
 |> OriginalLength           : -1
 AgentBuffer-PMO  offset 408-535(=0x198-0x217):(128=0x80 bytes, Address 00AB2F0C)
 00000000:504D4F20 01000000 42800000 FFFFFFFF  [PMO ....B.......]
 00000010:00000000 00000000 00000000 00000000  [................]
 00000020-0000007F same as above
 |> Version                  : 1
 |> Options                  : SYNCPOINT / NEW_MSG_ID / LOGICAL_ORDER
 |> Timeout                  : -1
 |> Context                  : 00000000
 |> KnownDestCount           : 0
 |> UnknownDestCount         : 0
 |> InvalidDestCount         : 0
 |> ResolvedQName            :
 |> ResolvedQMgrName         :
 AgentBuffer- offset 536-539(=0x218-0x21B):(4=0x4 bytes, Address 00AB2F8C)
 00000000:C6000000                             [....]
 AgentBuffer-RFH  offset 540-671(=0x21C-0x29F):(132=0x84 bytes, Address 00AB2F90)
 00000000:52464820 02000000 84000000 22020000  [RFH ........"...]
 00000010:BA130000 4D515354 52202020 00000000  [....MQSTR   ....]
 00000020:B8040000 5C000000 3C6D6364 3E3C4D73  [....¥...<mcd><Ms]
 00000030:643E4D52 4D3C2F4D 73643E3C 5365743E  [d>MRM</Msd><Set>]
 00000040:444D5051 4F434330 37393939 393C2F53  [DMPQOCC079999</S]
 00000050:65743E3C 54797065 3E4D5347 5F4D4958  [et><Type>MSG_MIX]
 00000060:45444341 53455F49 443C2F54 7970653E  [EDCASE_ID</Type>]
 00000070:3C466D74 3E435746 3C2F466D 743E3C2F  [<Fmt>CWF</Fmt></]
 00000080:6D63643E                             [mcd>]
 |> ---------- offset 0 (=0x0) - MQHRF2 (Rules and formatting header 2)
 |> Version                  : 2
 |> StrucLength              : 132
 |> Encoding                 : 546 (Native encoding) INTEGER_REVERSED / DECIMAL_REVERSED / FLOAT_IEEE_REVERSED
 |> CodedCharSetId           : 5050
 |> Format                   : MQSTR
 |> Flags                    : 0
 |> NameValueCCSID           : 1208
 |> ---------- offset 40 (=0x28) - XML data
 |> <mcd>
 |>   <Msd>
 |>     MRM
 |>   </Msd>
 |>   <Set>
 |>     DMPQOCC079999
 |>   </Set>
 |>   <Type>
 |>     MSG_MIXEDCASE_ID
 |>   </Type>
 |>   <Fmt>
 |>     CWF
 |>   </Fmt>
 |> </mcd>
 AgentBuffer- offset 672-737(=0x2A0-0x2E1):(66=0x42 bytes, Address 00AB3014)
 00000000:4F73616D 7520496E 6F756520 536F7461  [Osamu Inoue Sota]
 00000010:20496E6F 75658140 814082A0 82A282A4  [ Inoue.@.@......]
 00000020:20205361 746F6520 82A682A6 82A682A6  [  Satoe ........]
 00000030:8140B0B3 B2A9B3C2 B3C9F2CD F2F4E9DA  [.@..............]
 00000040:D9F8                                 [..]
 -
 |--- Seq # : 7 ----------------------------------------
 [2006/12/27 17:22:15.296]0000000024-0106661285
 <XMIT
 (CHANNEL_SEND_EXIT
 DataLength        : 540 (=0x21C)
 AgentBufferLength : 4000 (=0xFA0)
 API call type     : MQPUT reply (=0x96)
 Segmentation      : none (=0x30)
 ObjectName        : Q1
 CompCode          : 0
 Reason            : 0
 AgentBuffer-TSH  offset 0-43(=0x0-0x2B):(44=0x2C bytes, Address 00AB2D74)
 00000000:54534820 1C020000 02963000 00000000  [TSH ......0.....]
 00000010:00000000 22020000 A4030000 0000021C  [...."...........]
 00000020:00000000 00000000 F0FB3E00           [..........>.]
 AgentBuffer-MD   offset 44-407(=0x2C-0x197):(364=0x16C bytes, Address 00AB2DA0)
 00000000:4D442020 02000000 00000000 08000000  [MD  ............]
 00000010:FFFFFFFF 00000000 22020000 AF030000  [........".......]
 00000020:4D514852 46322020 00000000 01000000  [MQHRF2  ........]
 00000030:414D5120 514D3120 20202020 20202020  [AMQ QM1         ]
 00000040:E32C9245 20000902 00000000 00000000  [.,.E ...........]
 00000050:00000000 00000000 00000000 00000000  [................]
 00000060:00000000 51524550 4C592020 20202020  [....QREPLY      ]
 00000070:20202020 20202020 20202020 20202020  [                ]
 00000080-0000008F same as above
 00000090:20202020 00000000 00000000 00000000  [    ............]
 000000A0:00000000 00000000 00000000 00000000  [................]
 000000B0-000000BF same as above
 000000C0:00000000 6F696E6F 75652020 20202020  [....oinoue      ]
 000000D0:16010515 000000BE 043E324B 2CBC1A82  [.........>2K,...]
```

```
000000E0:8BA628EB 03000000 00000000 0000000B  [..(.............]
000000F0:20202020 20202020 20202020 20202020  [                ]
00000100-0000010F same as above
00000110:0B000000 756E5C69 73656C5F 6D716C6F  [....un¥isel_mqlo]
00000120:6E677275 6E5F6D71 69633332 2E657865  [ngrun_mqic32.exe]
00000130:32303036 31323237 30383232 31353132  [2006122708221512]
00000140:20202020 00000000 00000000 00000000  [    ............]
00000150:00000000 00000000 00000000 01000000  [................]
00000160:00000000 00000000 FFFFFFFF           [............]
 |> Version                 : 2
 |> Report                  : (none)
 |> MsgType                 : DATAGRAM
 |> Expiry                  : -1
 |> Feedback                : 0
 |> Encoding                : 546 (Native encoding) INTEGER_REVERSED / DECIMAL_REVERSED / FLOAT_IEEE_REVERSED
 |> CodedCharSetId          : 943
 |> Format                  : MQHRF2
 |> Priority                : 0
 |> Persistence             : PERSISTENT
 |> MsgId                   : 414D5120 514D3120 20202020 20202020 E32C9245 20000902  [AMQ QM1         .,.E ...]
 |> CorrelId                : (null)
 |> BackoutCount            : 0
 |> ReplyToQ                : QREPLY
 |> ReplyToQMgr             :
 |> UserIdentifier          : oinoue
 |> AccountingToken         : 16010515 000000BE 043E324B 2CBC1A82 8BA628EB 03000000 00000000 0000000B
 |> ApplIdentityData        :
 |> PutApplType             : Windows
 |> PutApplName             : un¥isel_mqlongrun_mqic32.exe
 |> PutDate                 : 20061227
 |> PutTime                 : 08221512
 |> ApplOriginData          :
 |> --- MD version 2----------
 |> GroupId                 : (null)
 |> MsgSeqNumber            : 1
 |> Offset                  : 0
 |> MsgFlags                : (none)
 |> OriginalLength          : -1
AgentBuffer-PMO  offset 408-535(=0x198-0x217):(128=0x80 bytes, Address 00AB2F0C)
00000000:504D4F20 01000000 42800000 FFFFFFFF  [PMO ....B.......]
00000010:00000000 01000000 00000000 00000000  [................]
00000020:51312020 20202020 20202020 20202020  [Q1              ]
00000030:20202020 20202020 20202020 20202020  [                ]
00000040-0000004F same as above
00000050:514D3120 20202020 20202020 20202020  [QM1             ]
00000060:20202020 20202020 20202020 20202020  [                ]
00000070-0000007F same as above
 |> Version                 : 1
 |> Options                 : SYNCPOINT / NEW_MSG_ID / LOGICAL_ORDER
 |> Timeout                 : -1
 |> Context                 : 00000000
 |> KnownDestCount          : 1
 |> UnknownDestCount        : 0
 |> InvalidDestCount        : 0
 |> ResolvedQName           : Q1
 |> ResolvedQMgrName        : QM1
AgentBuffer- offset 536-539(=0x218-0x21B):(4=0x4 bytes, Address 00AB2F8C)
00000000:C6000000                             [....]
-
|--- Seq # : 8 --------------------------------------
[2006/12/27 17:22:15.296]0000000024-0106669451
<XMIT
(CHANNEL_RCV_EXIT
DataLength        : 48 (=0x30)
AgentBufferLength : 32766 (=0x7FFE)
API call type     : MQCLOSE request (=0x84)
Segmentation      : none (=0x30)
ObjectName        : Q1
AgentBuffer-TSH  offset 0-43(=0x0-0x2B):(44=0x2C bytes, Address 00AB2D74)
00000000:54534820 00000030 02843000 00000000  [TSH ...0..0.....]
00000010:00000000 22020000 A4030000 00000030  [....″..........0]
00000020:00000000 00000000 F0FB3E00           [..........>.]
AgentBuffer- offset 44-47(=0x2C-0x2F):(4=0x4 bytes, Address 00AB2DA0)
00000000:00000000                             [....]
-
|--- Seq # : 9 --------------------------------------
[2006/12/27 17:22:15.296]0000000024-0106678410
<XMIT
(CHANNEL_SEND_EXIT
DataLength        : 44 (=0x2C)
AgentBufferLength : 4000 (=0xFA0)
API call type     : MQCLOSE reply (=0x94)
Segmentation      : none (=0x30)
ObjectName        : Q1
CompCode          : 0
Reason            : 0
AgentBuffer-TSH  offset 0-43(=0x0-0x2B):(44=0x2C bytes, Address 00AB2D74)
00000000:54534820 2C000000 02943000 00000000  [TSH ,.....0.....]
00000010:00000000 22020000 A4030000 0000002C  [....″..........,]
00000020:00000000 00000000 FFFFFFFF           [............]
```

```
-
|--- Seq # : 10 --------------------------------------
[2006/12/27 17:22:15.296]0000000024-0106680050
<XMIT
(CHANNEL_RCV_EXIT
DataLength        : 44 (=0x2C)
AgentBufferLength : 32766 (=0x7FFE)
API call type     : MQDISC request (=0x82)
Segmentation      : none (=0x30)
AgentBuffer-TSH  offset 0-43(=0x0-0x2B):(44=0x2C bytes, Address 00AB2D74)
00000000:54534820 0000002C 02823000 00000000  [TSH ...,..0.....]
00000010:00000000 22020000 A4030000 0000002C  [....".........,]
00000020:00000000 00000000 00000000           [............]
-
|--- Seq # : 11 --------------------------------------
[2006/12/27 17:22:15.312]0000000024-0106726500
<XMIT
(CHANNEL_SEND_EXIT
DataLength        : 44 (=0x2C)
AgentBufferLength : 4000 (=0xFA0)
API call type     : MQDISC reply (=0x92)
Segmentation      : none (=0x30)
CompCode          : 0
Reason            : 0
AgentBuffer-TSH  offset 0-43(=0x0-0x2B):(44=0x2C bytes, Address 00AB2D74)
00000000:54534820 2C000000 02923000 00000000  [TSH ,.....0.....]
00000010:00000000 22020000 A4030000 0000002C  [....".........,]
00000020:00000000 00000000 00000000           [............]
-
|--- Seq # : 12 --------------------------------------
[2006/12/27 17:22:15.312]0000000024-0106728140
<XMIT
(CHANNEL_RCV_EXIT
DataLength        : 28 (=0x1C)
AgentBufferLength : 32766 (=0x7FFE)
API call type     : (N/A) (=0x05)
Segmentation      : (others) (=0x08)
AgentBuffer-TSH  offset 0-27(=0x0-0x1B):(28=0x1C bytes, Address 00AB2D74)
00000000:54534820 0000001C 02050800 00000000  [TSH ............]
00000010:00000000 22020000 A4030000           [....".......]
-
[2006/12/27 17:22:15.343]0000000024-0106858054
<TERM
(CHANNEL_SEND_EXIT
-
[2006/12/27 17:22:15.343]0000000024-0106858101
<TERM
(CHANNEL_RCV_EXIT
-
|---------------------------------------------------------------------------------------------
|Call summary - [2006/12/27 17:22:15.343]0000000024-0106858182
|                                                                ---- Data length (byte) -----
|  API              Success     Info  Warning     Fail    Total  Average  Minimum  Maximum
|  --------------- -------- -------- -------- -------- -------- -------- -------- --------
|  MQCONN  request                                           1      304      304      304
|  MQCONN  reply          1        0        0        0       1      304      304      304
|  MQDISC  request                                           1       44       44       44
|  MQDISC  reply          1        0        0        0       1       44       44       44
|  MQOPEN  request                                           1      216      216      216
|  MQOPEN  reply          1        0        0        0       1      216      216      216
|  MQCLOSE request                                           1       48       48       48
|  MQCLOSE reply          1        0        0        0       1       44       44       44
|  MQPUT   request                                           1      738      738      738
|  MQPUT   reply          1        0        0        0       1      540      540      540
|  MQPUT1  request                                           0
|  MQPUT1  reply          0        0        0        0       0
|  MQGET   request                                           0
|  MQGET   reply          0        0        0        0       0
|  MQINQ   request                                           0
|  MQINQ   reply          0        0        0        0       0
|  MQSET   request                                           0
|  MQSET   reply          0        0        0        0       0
|  MQCMIT  request                                           0
|  MQCMIT  reply          0        0        0        0       0
|  MQBACK  request                                           0
|  MQBACK  reply          0        0        0        0       0
|  XA      request                                           0
|  XA      reply          0        0        0        0       0
|  XMIT    send                                              0
|  XMIT    receive                                           2       94       28      160
|  MSG                                                       0
|  others                                                    0
|  --------------- -------- -------- -------- -------- -------- -------- -------- --------
|  Total                  5        0        0        0      12
|---------------------------------------------------------------------------------------------
|
```

(Channel Exit – non MQI) with ShowSummaryOnly=yes parameter

```
|----------------------------------------------------------
|Program information
|  Program name       : WEBSPHERE MQ Channel Exit (SupportPac MA0W)
|  Version            : 5.50
|  Provided by        : oinoue@jp.ibm.com

       ........

|Process parameters
|  Parameter file     : /home/apiexit/tracedata/qx.def
|  Show max MQPUT len : (all)
|  Show max MQGET len : (all)
|  Show max chl len   : (all)
|  Show data conv     : no
|  Show summary only  : yes
|  Show parsed data   : yes
|  Show parsed XML    : yes
|  Show hex char image: yes
|  Show hex char ccsid: 0
|  Show perf only     : no
|  Selected call only : no
|  Alert file (error) : yes
|  Alert file (warn)  : no
|  Perf intvl (time)  : 0
|  Perf intvl (count) : 0
|  Max file size      : 100,000,000
|  Max XML line length: 100
|  Compress command   : (none)
|  Compress threshold : 1,000,000
|  Create empty file  : yes
|  Write stream mode  : no
|  Write cache        : yes
|  Write after call   : yes
|  Write buffer size  : 50,000
|  Write interval     : 30
|  Check update req   : 30
|  Suppress alert     : 2, 2085, MQOPEN , SYSTEM.BROKER.INTER.BROKER.COMMUNICATIONS, DataFlowEngine
|  Suppress alert     : 2, 2038, MQINQ  , SYSTEM.BROKER.ADMIN.REPLY, DataFlowEngine
|   (Debug flag)      : off
|----------------------------------------------------------
-
[2006/05/11 17:17:45.935663]
<INIT
(CHANNEL_MSG_EXIT
-
[2006/05/11 17:17:45.935720]
<INIT
(CHANNEL_SEND_EXIT
-
[2006/05/11 17:17:45.935764]
<INIT
(CHANNEL_RCV_EXIT
-
        Time                          API                 Segmentation    DataLen CC Reason MsgId        Queue  Queue manager
      1 [2006/05/11 17:17:49.280867]  CHANNEL_MSG_EXIT       none        50000428  -    -   414D512051..  Q1          QM2
      2 [2006/05/11 17:17:49.280989]  CHANNEL_SEND_EXIT      first          32750  -    -   414D512051..  Q1          QM2
      3 [2006/05/11 17:17:49.281179]  CHANNEL_SEND_EXIT      middle         32750  -    -
      4 [2006/05/11 17:17:49.431942]  CHANNEL_SEND_EXIT      middle         32750  -    -
      5 [2006/05/11 17:17:49.631971]  CHANNEL_SEND_EXIT      middle         32750  -    -
      6 [2006/05/11 17:17:49.832010]  CHANNEL_SEND_EXIT      middle         32750  -    -
      7 [2006/05/11 17:17:49.832097]  CHANNEL_SEND_EXIT      middle         32750  -    -
      8 [2006/05/11 17:17:49.832204]  CHANNEL_SEND_EXIT      middle         32750  -    -
      9 [2006/05/11 17:17:49.832289]  CHANNEL_SEND_EXIT      middle         32750  -    -
     10 [2006/05/11 17:17:49.832370]  CHANNEL_SEND_EXIT      middle         32750  -    -
     11 [2006/05/11 17:17:49.832453]  CHANNEL_SEND_EXIT      middle         32750  -    -

            ........

   1525 [2006/05/11 17:17:50.041443]  CHANNEL_SEND_EXIT      middle         32750  -    -
   1526 [2006/05/11 17:17:50.041581]  CHANNEL_SEND_EXIT      middle         32750  -    -
   1527 [2006/05/11 17:17:50.041718]  CHANNEL_SEND_EXIT      middle         32750  -    -
   1528 [2006/05/11 17:17:50.041853]  CHANNEL_SEND_EXIT      middle         32750  -    -
   1529 [2006/05/11 17:17:50.041988]  CHANNEL_SEND_EXIT      middle         32750  -    -
   1530 [2006/05/11 17:17:50.048600]  CHANNEL_SEND_EXIT      last           31820  -    -
   1531 [2006/05/11 17:17:52.957220]  CHANNEL_RCV_EXIT       none              28  -    -
   1532 [2006/05/11 17:22:52.967873]  CHANNEL_SEND_EXIT      (others)          28  -    -
   1533 [2006/05/11 17:22:52.968431]  CHANNEL_RCV_EXIT       none              28  -    -
   1534 [2006/05/11 17:27:52.977817]  CHANNEL_SEND_EXIT      (others)          28  -    -
   1535 [2006/05/11 17:27:52.978183]  CHANNEL_RCV_EXIT       none              28  -    -
   1536 [2006/05/11 17:32:52.987858]  CHANNEL_SEND_EXIT      (others)          28  -    -
   1537 [2006/05/11 17:32:52.988242]  CHANNEL_RCV_EXIT       none              28  -    -
   1538 [2006/05/11 17:37:52.997841]  CHANNEL_SEND_EXIT      (others)          28  -    -
   1539 [2006/05/11 17:37:52.998241]  CHANNEL_RCV_EXIT       none              28  -    -
   1540 [2006/05/11 17:42:53.007829]  CHANNEL_SEND_EXIT      (others)          28  -    -
```

```
    1541 [2006/05/11 17:42:53.008227]  CHANNEL_RCV_EXIT       none              28  -    -
    1542 [2006/05/11 17:47:53.017820]  CHANNEL_SEND_EXIT      (others)          28  -    -
    1543 [2006/05/11 17:47:53.018209]  CHANNEL_RCV_EXIT       none              28  -    -
    1544 [2006/05/11 17:52:53.027825]  CHANNEL_SEND_EXIT      (others)          28  -    -
    1545 [2006/05/11 17:52:53.028221]  CHANNEL_RCV_EXIT       none              28  -    -
    1546 [2006/05/11 17:57:53.037825]  CHANNEL_SEND_EXIT      (others)          28  -    -
    1547 [2006/05/11 17:57:53.038238]  CHANNEL_RCV_EXIT       none              28  -    -
    1548 [2006/05/11 18:02:53.048769]  CHANNEL_SEND_EXIT      (others)          28  -    -
    1549 [2006/05/11 18:02:53.049157]  CHANNEL_RCV_EXIT       none              28  -    -
    1550 [2006/05/11 18:06:13.767122]  CHANNEL_SEND_EXIT      none              28  -    -
-
[2006/05/11 18:06:13.767301]
<TERM
(CHANNEL_MSG_EXIT
-
[2006/05/11 18:06:13.767315]
<TERM
(CHANNEL_SEND_EXIT
-
[2006/05/11 18:06:13.767332]
<TERM
(CHANNEL_RCV_EXIT
-
|-------------------------------------------------------------------------------------------
|Call summary - [2006/05/11 18:06:13.767347]
|                                                                 ---- Data length (byte) -----
|  API                Success     Info  Warning     Fail    Total   Average   Minimum   Maximum
|  --------------- -------- -------- -------- -------- -------- --------- --------- ---------
|  MQCONN  request                                             0
|  MQCONN  reply           0        0        0        0        0
|  MQDISC  request                                             0
|  MQDISC  reply           0        0        0        0        0
|  MQOPEN  request                                             0
|  MQOPEN  reply           0        0        0        0        0
|  MQCLOSE request                                             0
|  MQCLOSE reply           0        0        0        0        0
|  MQPUT   request                                             0
|  MQPUT   reply           0        0        0        0        0
|  MQPUT1  request                                             0
|  MQPUT1  reply           0        0        0        0        0
|  MQGET   request                                             0
|  MQGET   reply           0        0        0        0        0
|  MQINQ   request                                             0
|  MQINQ   reply           0        0        0        0        0
|  MQSET   request                                             0
|  MQSET   reply           0        0        0        0        0
|  MQCMIT  request                                             0
|  MQCMIT  reply           0        0        0        0        0
|  MQBACK  request                                             0
|  MQBACK  reply           0        0        0        0        0
|  XA      request                                             0
|  XA      reply           0        0        0        0        0
|  XMIT    send                                             1539     32537        28     32750
|  XMIT    receive                                            10        28        28        28
|  MSG                                                         1  50000428  50000428  50000428
|  others                                                      0
|  --------------- -------- -------- -------- -------- -------- --------- --------- ---------
|  Total                   0        0        0        0     1550
|-------------------------------------------------------------------------------------------
|
```

(Channel Exit – non MQI) with default parameter values

```
|-----------------------------------------------------------
|Program information
|  Program name       : WEBSPHERE MQ Channel Exit (SupportPac MA0W)
|  Version            : 5.80
|  Provided by        : oinoue@jp.ibm.com
|Module attributes
|  Addressing mode    : 32-bit
|  Threading          : Common to single thread and multi thread
|Operating environment
|  Operating system   : Windows XP
|  Major version      : 5
|  Minor version      : 1
|  Service level      : Service Pack 2
|  Build number       : 2600
|  Computer name      : WIGAN
|API Exit context
|  Exit name          : qxEntry
|  API caller type    :
|  Queue manager      : QM1
|  Environment        :
|  User id            :
|  Connection name    : localhost(1415)
|  Appl/Channel name  : TO.QM2
|  Appl/Channel type  : SENDER
|  Program name       : channel_SENDER_TO.QM2
|  Program description:
|  Process id         : 2500
|  Thread id          : 3136
|  Process id (real)  : 2500
|  Thread id  (real)  : 3136
|Process parameters
|  Parameter file     : E:¥ApiExit¥tracedata¥qx.def
|  Check update req   : 30
|  Show summary only  : no
|  Show perf only     : no
|  Perf intvl (time)  : 0
|  Perf intvl (count) : 0
|  Show parsed data   : yes
|  Show parsed content: yes
|  Max out line length: 100
|  Show hex char image: yes
|  Show hex char ccsid: 0
|  Show data conv     : no
|  Show max len MQPUT : (all)
|  Show max len MQGET : (all)
|  Show max len chl   : (all)
|  Show max len parsed: (all)
|  Count only selected: no
|  Alert file (error) : yes
|  Alert file (warn)  : no
|  Suppress alert     : 2, 2085, MQOPEN , SYSTEM.BROKER.INTER.BROKER.COMMUNICATIONS, DataFlowEngine
|  Suppress alert     : 2, 2038, MQINQ  , SYSTEM.BROKER.ADMIN.REPLY, DataFlowEngine
|  Max file size      : 100,000,000
|  Compress command   : (none)
|  Compress threshold : 1,000,000
|  Create empty file  : yes
|  Write stream mode  : no
|  Write cache        : yes
|  Write after call   : yes
|  Write buffer size  : 10,000
|  Write interval     : 30
|   (Debug flag)      : off
|-----------------------------------------------------------
-
[2006/09/12 15:28:35.718]0000017000-3340792093
<INIT
(CHANNEL_MSG_EXIT
-
[2006/09/12 15:28:35.718]0000017000-3345479014
<INIT
(CHANNEL_SEND_EXIT
-
[2006/09/12 15:28:35.718]0000017000-3346689910
<INIT
(CHANNEL_RCV_EXIT
-
|--- Seq # : 1 ----------------------------------------
[2006/09/12 15:28:37.375]0000017002-0689626031
<MSG
(CHANNEL_MSG_EXIT
DataLength        : 626 (=0x272)
AgentBufferLength : 32718 (=0x7FCE)
AgentBuffer-XQH  offset 0-103(=0x0-0x67):(104=0x68 bytes, Address 01135A44)
```

```
00000000:58514820 01000000 51322020 20202020  [XQH ....Q2      ]
00000010:20202020 20202020 20202020 20202020  [                ]
00000020-0000002F same as above
00000030:20202020 20202020 514D3220 20202020  [        QM2     ]
00000040:20202020 20202020 20202020 20202020  [                ]
00000050-0000005F same as above
00000060:20202020 20202020                    [        ]
 |> Version                 : 1
 |> RemoteQName             : Q2
 |> RemoteQMgrName          : QM2
AgentBuffer-MD   offset 104-427(=0x68-0x1AB):(324=0x144 bytes, Address 01135AAC)
00000000:4D442020 01000000 00000000 08000000  [MD  ............]
00000010:FFFFFFFF 00000000 22020000 AF030000  [........".......]
00000020:4D514852 46322020 00000000 01000000  [MQHRF2  ........]
00000030:414D5120 514D3120 20202020 20202020  [AMQ QM1         ]
00000040:F5520645 20000802 00000000 00000000  [.R.E ...........]
00000050:00000000 00000000 00000000 00000000  [................]
00000060:00000000 51524550 4C592020 20202020  [....QREPLY      ]
00000070:20202020 20202020 20202020 20202020  [                ]
00000080-0000008F same as above
00000090:20202020 514D3120 20202020 20202020  [    QM1         ]
000000A0:20202020 20202020 20202020 20202020  [                ]
000000B0-000000BF same as above
000000C0:20202020 6973656C 6D712020 20202020  [    iselmq      ]
000000D0:16010515 0000009D BEDA5285 E77E2F43  [..........R..~/C]
000000E0:170A32EB 03000000 00000000 0000000B  [..2.............]
000000F0:20202020 20202020 20202020 20202020  [                ]
00000100-0000010F same as above
00000110:0B000000 5F6D716C 6F6E6772 756E5F6D  [...._mqlongrun_m]
00000120:6F646966 795F6D71 69633332 2E657865  [odify_mqic32.exe]
00000130:32303036 30393132 30363237 30353238  [2006091206270528]
00000140:20202020                             [    ]
 |> Version                 : 1
 |> Report                  : (none)
 |> MsgType                 : DATAGRAM
 |> Expiry                  : -1
 |> Feedback                : 0
 |> Encoding                : 546 (Native encoding) INTEGER_REVERSED / DECIMAL_REVERSED / FLOAT_IEEE_REVERSED
 |> CodedCharSetId          : 943
 |> Format                  : MQHRF2
 |> Priority                : 0
 |> Persistence             : PERSISTENT
 |> MsgId                   : 414D5120 514D3120 20202020 20202020 F5520645 20000802
 |> CorrelId                : (null)
 |> BackoutCount            : 0
 |> ReplyToQ                : QREPLY
 |> ReplyToQMgr             : QM1
 |> UserIdentifier          : iselmq
 |> AccountingToken         : 16010515 0000009D BEDA5285 E77E2F43 170A32EB 03000000 00000000 0000000B
 |> ApplIdentityData        :
 |> PutApplType             : Windows
 |> PutApplName             : _mqlongrun_modify_mqic32.exe
 |> PutDate                 : 20060912
 |> PutTime                 : 06270528
 |> ApplOriginData          :
AgentBuffer-RFH  offset 428-559(=0x1AC-0x22F):(132=0x84 bytes, Address 01135BF0)
00000000:52464820 02000000 84000000 22020000  [RFH ........"...]
00000010:BA130000 4D515354 52202020 00000000  [....MQSTR   ....]
00000020:B8040000 5C000000 3C6D6364 3E3C4D73  [....¥...<mcd><Ms]
00000030:643E4D52 4D3C2F4D 73643E3C 5365743E  [d>MRM</Msd><Set>]
00000040:444D5051 4F434330 37393939 393C2F53  [DMPQOCC079999</S]
00000050:65743E3C 54797065 3E4D5347 5F4D4958  [et><Type>MSG_MIX]
00000060:45444341 53455F49 443C2F54 7970653E  [EDCASE_ID</Type>]
00000070:3C466D74 3E435746 3C2F466D 743E3C2F  [<Fmt>CWF</Fmt></]
00000080:6D63643E                             [mcd>]
 |> offset 0 (=0x0) - MQHRF2 (Rules and formatting header 2)
 |> Version                 : 2
 |> StrucLength             : 132
 |> Encoding                : 546 (Native encoding) INTEGER_REVERSED / DECIMAL_REVERSED / FLOAT_IEEE_REVERSED
 |> CodedCharSetId          : 5050
 |> Format                  : MQSTR
 |> Flags                   : 0
 |> NameValueCCSID          : 1208
 |> <mcd>
 |>   <Msd>
 |>     MRM
 |>   </Msd>
 |>   <Set>
 |>     DMPQOCC079999
 |>   </Set>
 |>   <Type>
 |>     MSG_MIXEDCASE_ID
 |>   </Type>
 |>   <Fmt>
 |>     CWF
 |>   </Fmt>
 |> </mcd>
AgentBuffer- offset 560-625(=0x230-0x271):(66=0x42 bytes, Address 01135C74)
00000000:4F73616D 7520496E 6F756520 536F7461  [Osamu Inoue Sota]
00000010:20496E6F 75658140 814082A0 82A282A4  [ Inoue.@.@......]
```

```
00000020:20205361 746F6520 82A682A6 82A682A6  [  Satoe ........]
00000030:8140B0B3 B2A9B3C2 B3C9F2CD F2F4E9DA  [.@..............]
00000040:D9F8                                 [..]
-
|--- Seq # : 2 ------------------------------------
[2006/09/12 15:28:37.375]0000017002-0692516156
<XMIT
(CHANNEL_SEND_EXIT
DataLength        : 674 (=0x2A2)
AgentBufferLength : 32766 (=0x7FFE)
API call type     : (N/A) (=0x04)
Segmentation      : none (=0x31)
ObjectName        : Q2
AgentBuffer-TSH  offset 0-27(=0x0-0x1B):(28=0x1C bytes, Address 01135A14)
00000000:54534820 A2020000 02043100 F4520645  [TSH .......1..R.E]
00000010:10000101 22020000 A4030000           [....″.......]
AgentBuffer-MSH  offset 28-47(=0x1C-0x2F):(20=0x14 bytes, Address 01135A30)
00000000:4D534820 70010000 C6000000 00000000  [MSH p...........]
00000010:72020000                             [r...]
AgentBuffer-XQH  offset 48-151(=0x30-0x97):(104=0x68 bytes, Address 01135A44)
00000000:58514820 01000000 51322020 20202020  [XQH ....Q2      ]
00000010:20202020 20202020 20202020 20202020  [                ]
00000020-0000002F same as above
00000030:20202020 20202020 514D3220 20202020  [        QM2     ]
00000040:20202020 20202020 20202020 20202020  [                ]
00000050-0000005F same as above
00000060:20202020 20202020                    [        ]
|> Version                    : 1
|> RemoteQName                : Q2
|> RemoteQMgrName             : QM2
AgentBuffer-MD   offset 152-475(=0x98-0x1DB):(324=0x144 bytes, Address 01135AAC)
00000000:4D442020 01000000 00000000 08000000  [MD  ............]
00000010:FFFFFFFF 00000000 22020000 AF030000  [........″.......]
00000020:4D514852 46322020 00000000 01000000  [MQHRF2  ........]
00000030:414D5120 514D3120 20202020 20202020  [AMQ QM1         ]
00000040:F5520645 20000802 00000000 00000000  [.R.E ...........]
00000050:00000000 00000000 00000000 00000000  [................]
00000060:00000000 51524550 4C592020 20202020  [....QREPLY      ]
00000070:20202020 20202020 20202020 20202020  [                ]
00000080-0000008F same as above
00000090:20202020 514D3120 20202020 20202020  [    QM1         ]
000000A0:20202020 20202020 20202020 20202020  [                ]
000000B0-000000BF same as above
000000C0:20202020 6973656C 6D712020 20202020  [    iselmq      ]
000000D0:16010515 0000009D BEDA5285 E77E2F43  [..........R..~/C]
000000E0:170A32EB 03000000 00000000 0000000B  [..2.............]
000000F0:20202020 20202020 20202020 20202020  [                ]
00000100-0000010F same as above
00000110:0B000000 5F6D716C 6F6E6772 756E5F6D  [...._mqlongrun_m]
00000120:6F646966 795F6D71 69633332 2E657865  [odify_mqic32.exe]
00000130:32303036 30393132 30363237 30353238  [2006091206270528]
00000140:20202020                             [    ]
|> Version                    : 1
|> Report                     : (none)
|> MsgType                    : DATAGRAM
|> Expiry                     : -1
|> Feedback                   : 0
|> Encoding                   : 546 (Native encoding) INTEGER_REVERSED / DECIMAL_REVERSED / FLOAT_IEEE_REVERSED
|> CodedCharSetId             : 943
|> Format                     : MQHRF2
|> Priority                   : 0
|> Persistence                : PERSISTENT
|> MsgId                      : 414D5120 514D3120 20202020 20202020 F5520645 20000802
|> CorrelId                   : (null)
|> BackoutCount               : 0
|> ReplyToQ                   : QREPLY
|> ReplyToQMgr                : QM1
|> UserIdentifier             : iselmq
|> AccountingToken            : 16010515 0000009D BEDA5285 E77E2F43 170A32EB 03000000 00000000 0000000B
|> ApplIdentityData           :
|> PutApplType                : Windows
|> PutApplName                : _mqlongrun_modify_mqic32.exe
|> PutDate                    : 20060912
|> PutTime                    : 06270528
|> ApplOriginData             :
AgentBuffer-RFH  offset 476-607(=0x1DC-0x25F):(132=0x84 bytes, Address 01135BF0)
00000000:52464820 02000000 84000000 22020000  [RFH ........″...]
00000010:BA130000 4D515354 52202020 00000000  [....MQSTR   ....]
00000020:B8040000 5C000000 3C6D6364 3E3C4D73  [....¥...<mcd><Ms]
00000030:643E4D52 4D3C2F4D 73643E3C 5365743E  [d>MRM</Msd><Set>]
00000040:444D5051 4F434330 37393939 393C2F53  [DMPQOCC079999</S]
00000050:65743E3C 54797065 3E4D5347 5F4D4958  [et><Type>MSG_MIX]
00000060:45444341 53455F49 443C2F54 7970653E  [EDCASE_ID</Type>]
00000070:3C466D74 3E435746 3C2F466D 743E3C2F  [<Fmt>CWF</Fmt></]
00000080:6D63643E                             [mcd>]
|> offset 0 (=0x0) - MQHRF2 (Rules and formatting header 2)
|> Version                    : 2
|> StrucLength                : 132
|> Encoding                   : 546 (Native encoding) INTEGER_REVERSED / DECIMAL_REVERSED / FLOAT_IEEE_REVERSED
|> CodedCharSetId             : 5050
|> Format                   : MQSTR
|> Flags                    : 0
|> NameValueCCSID           : 1208
|> <mcd>
|>   <Msd>
|>     MRM
|>   </Msd>
|>   <Set>
|>     DMPQOCC079999
|>   </Set>
|>   <Type>
|>     MSG_MIXEDCASE_ID
|>   </Type>
|>   <Fmt>
|>     CWF
|>   </Fmt>
|> </mcd>
AgentBuffer- offset 608-673(=0x260-0x2A1):(66=0x42 bytes, Address 01135C74)
00000000:4F73616D 7520496E 6F756520 536F7461  [Osamu Inoue Sota]
00000010:20496E6F 75658140 814082A0 82A282A4  [ Inoue.@.@......]
00000020:20205361 746F6520 82A682A6 82A682A6  [  Satoe ........]
00000030:8140B0B3 B2A9B3C2 B3C9F2CD F2F4E9DA  [.@..............]
00000040:D9F8                                 [..]
-
|--- Seq # : 3 ---------------------------------------
[2006/09/12 15:28:37.375]0000017002-0736095227
<XMIT
(CHANNEL_RCV_EXIT
DataLength        : 28 (=0x1C)
AgentBufferLength : 32766 (=0x7FFE)
API call type     : (N/A) (=0x05)
Segmentation      : none (=0x00)
AgentBuffer-TSH  offset 0-27(=0x0-0x1B):(28=0x1C bytes, Address 01135A14)
00000000:54534820 0000001C 02050008 6187E2C8  [TSH ........a...]
00000010:00000000 22020000 A4030000           [....″.......]
-
|--- Seq # : 4 ---------------------------------------
[2006/09/12 15:29:01.718]0000017022-2448193740
<XMIT
(CHANNEL_SEND_EXIT
DataLength        : 28 (=0x1C)
AgentBufferLength : 32766 (=0x7FFE)
API call type     : (N/A) (=0x05)
Segmentation      : none (=0x08)
AgentBuffer-TSH  offset 0-27(=0x0-0x1B):(28=0x1C bytes, Address 01135A14)
00000000:54534820 1C000000 02050800 00000000  [TSH ............]
00000010:00000000 22020000 A4030000           [....″.......]
-
[2006/09/12 15:29:01.718]0000017022-2448905235
<TERM
(CHANNEL_MSG_EXIT
-
[2006/09/12 15:29:01.718]0000017022-2448942540
<TERM
(CHANNEL_SEND_EXIT
-
[2006/09/12 15:29:01.718]0000017022-2448972195
<TERM
(CHANNEL_RCV_EXIT
-
|---------------------------------------------------------------------------------------------
|Call summary - [2006/09/12 15:29:01.718]0000017022-2449023540
|                                                                ---- Data length (byte) -----
|  API               Success     Info  Warning     Fail    Total   Average   Minimum   Maximum
|  ---------------- -------- -------- -------- -------- -------- --------- --------- ---------
|  MQCONN  request                                             0
|  MQCONN  reply           0        0        0        0        0
|  MQDISC  request                                             0
|  MQDISC  reply           0        0        0        0        0
|  MQOPEN  request                                             0
|  MQOPEN  reply           0        0        0        0        0
|  MQCLOSE request                                             0
|  MQCLOSE reply           0        0        0        0        0
|  MQPUT   request                                             0
|  MQPUT   reply           0        0        0        0        0
|  MQPUT1  request                                             0
|  MQPUT1  reply           0        0        0        0        0
|  MQGET   request                                             0
|  MQGET   reply           0        0        0        0        0
|  MQINQ   request                                             0
|  MQINQ   reply           0        0        0        0        0
|  MQSET   request                                             0
|  MQSET   reply           0        0        0        0        0
|  MQCMIT  request                                             0
|  MQCMIT  reply           0        0        0        0        0
|  MQBACK  request                                             0
|  MQBACK  reply           0        0        0        0        0
|  XA      request                                             0
|  XA      reply           0        0        0        0        0
|  XMIT    send                                                2       351        28       674
```

```
  XMIT    receive                                         1        28        28        28
  MSG                                                     1       626       626       626
  others                                                  0
  --------------- -------- -------- -------- -------- -------- --------- --------- ---------
  Total                  0        0        0        0        4
---------------------------------------------------------------------------------------------
```

End of Document