

MQSeries®



Programming Interfaces Reference Summary

Note!

Before using this information and the product it supports, be sure to read the general information under "Appendix. Notices" on page 309.

Eighth Edition (November 2000)

This edition applies to the following products:

- MQSeries[®] for AIX[®] V5.1
- MQSeries for AS/400[®] Version 5 Release 1
- MQSeries for AT&T GIS UNIX V2.2
- MQSeries for Compaq (DIGITAL) OpenVMS, V2.2.1.1.1
- MQSeries for Compaq Tru64 Unix Version 5 Release 1
- MQSeries for HP-UX, V5.1
- MQSeries for OS/2[®] Warp V5.1
- MQSeries for OS/390[®] V5.2
- MQSeries for SINIX and DC/OSx, V2.2
- MQSeries for Sun Solaris, V5.1
- MQSeries for Sun Solaris, Intel Platform Edition, V5.1
- MQSeries for Tandem NonStop Kernel V2R2.0.1
- MQSeries for VSE/ESA[™] V2.1
- MQSeries for Windows NT[®] V5.1
- MQSeries for Windows[®] V2.0
- MQSeries for Windows V2.1

and to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright International Business Machines Corporation 1993, 2000. All rights reserved.**

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

Contents

About this book	x	Channel Stopped events	16
Related MQSeries publications	xi	Default Transmission Queue Type Error.	17
Terms used in this book.	xiii	Default Transmission Queue Usage Error	17
Windows products	xiii	Get Inhibited.	17
Appearance of values in this book	xiii	Not Authorized (type 1)	18
Summary of changes	xv	Not Authorized (type 2)	18
Changes for this edition (SX33-6095-07)	xv	Not Authorized (type 3)	18
Changes for the first edition (SX33-6095-06)	xv	Not Authorized (type 4)	19
Chapter 1. MQI calls	1	Put Inhibited.	19
MQBACK	1	Queue Depth High.	19
MQBEGIN	1	Queue Depth Low	20
MQCLOSE	2	Queue Full	20
MQCMIT	2	Queue Manager Active	21
MQCONN	2	Queue Manager Not Active.	21
MQCONNX	3	Queue Service Interval High	21
MQDISC	3	Queue Service Interval OK	21
MQGET	4	Queue Type Error	22
MQINQ	4	Remote Queue Name Error	22
MQOPEN	5	Transmission Queue Type Error	23
MQPUT	5	Transmission Queue Usage Error	23
MQPUT1	6	Unknown Alias Base Queue.	23
MQSET	6	Unknown Default Transmission Queue	24
MQSYNC	7	Unknown Object Name	24
Chapter 2. Exit calls	9	Unknown Remote Queue Manager	25
MQ_CHANNEL_AUTO_DEF_EXIT	9	Unknown Transmission Queue.	25
MQ_CHANNEL_EXIT	9	Chapter 4. Programmable Command	
MQ_CLUSTER_WORKLOAD_EXIT	10	Format (PCF) structures	27
MQ_DATA_CONV_EXIT.	10	Change, Copy, and Create Channel	27
MQ_TRANSPORT_EXIT	10	Required parameters (Change Channel).	27
MQXCNVC	11	Required parameters (Copy Channel and	
MQXWAIT	11	Create Channel).	27
Chapter 3. Event message summary	13	Optional parameters (sender or server	
Alias Base Queue Type Error	13	channel)	28
Bridge Started	13	Optional parameters (receiver channel)	29
Bridge Stopped	14	Optional parameters (requester channel)	30
Channel Activated	14	Optional parameters (server-connection	
Channel Auto-definition Error	14	channel)	31
Channel Auto-definition OK	15	Optional parameters (client-connection	
Channel Conversion Error	15	channel)	31
Channel Not Activated	15	Optional parameters (cluster-receiver	
Channel Started.	16	channel)	32
		Optional parameters (cluster-sender	
		channel)	33

Change, Copy, and Create Namelist commands	34	Ping Queue Manager	57
Required parameters (Change and Create Namelist)	35	Refresh Cluster	57
Required parameters (Copy Namelist)	35	Reset Channel	58
Optional parameters	35	Reset Cluster.	58
Change, Copy, and Create Process commands	35	Reset Queue Statistics.	58
Required parameters (Change and Create Process)	36	Reset Queue Statistics (Response)	59
Required parameters (Copy Process)	36	Resume Queue Manager Cluster	59
Optional parameters	36	Start Channel	59
Change, Copy, and Create Queue commands	36	Start Channel Initiator	60
Required parameters (Change Queue)	37	Start Channel Listener	60
Required parameters (Copy Queue and Create Queue)	37	Stop Channel	60
Optional parameters (alias queue).	37	Suspend Queue Manager Cluster	60
Optional parameters (local queue).	38		
Optional parameters (model queue)	39	Chapter 5. MQAI calls	63
Optional parameters (remote queue)	40	mqAddInquiry	63
Change Queue Manager	41	mqAddInteger	63
Clear Queue	41	mqAddString	64
Delete Channel	41	mqBagToBuffer	64
Delete Namelist	42	mqBufferToBag	65
Delete Process	42	mqClearBag	65
Delete Queue	42	mqCountItems	65
Escape	43	mqCreateBag.	66
Escape (Response)	43	mqDeleteBag.	66
Inquire Channel.	43	mqDeleteItem	66
Inquire Channel (Response)	44	mqExecute	67
Inquire Channel Names	46	mqGetBag.	67
Inquire Channel Names (Response)	46	mqInquireBag	68
Inquire Channel Status	46	mqInquireInteger	68
Inquire Channel Status (Response)	47	mqInquireItemInfo	69
Inquire Cluster Queue Manager	48	mqInquireString.	69
Inquire Cluster Queue Manager (Response)	48	mqPad	70
Inquire Namelist	50	mqPutBag.	70
Inquire Namelist (Response)	51	mqSetInteger.	71
Inquire Namelist Names	51	mqSetString	71
Inquire Namelist Names (Response)	51	mqTrim	72
Inquire Process	52	mqTruncateBag	72
Inquire Process (Response)	52		
Inquire Process Names	52	Chapter 6. Installable services interface	73
Inquire Process Names (Response)	53	MQZ_CHECK_AUTHORITY and	
Inquire Queue	53	MQZ_CHECK_AUTHORITY_2	73
Inquire Queue (Response)	53	MQZ_COPY_ALL_AUTHORITY	74
Inquire Queue Manager	55	MQZ_DELETE_AUTHORITY	74
Inquire Queue Manager (Response)	55	MQZ_DELETE_NAME	75
Inquire Queue Names.	56	MQZ_FIND_USERID	75
Inquire Queue Names (Response)	57	MQZ_GET_AUTHORITY and	
Ping Channel	57	MQZ_GET_AUTHORITY_2	76
		MQZ_GET_EXPLICIT_AUTHORITY and	
		MQZ_GET_EXPLICIT_AUTHORITY_2	76
		MQZ_INIT_AUTHORITY	78
		MQZ_INIT_NAME.	78

MQZ_INIT_USERID	79	amMsgGetCorrelId	102
MQZ_INSERT_NAME	79	amMsgGetDataLength	102
MQZ_LOOKUP_NAME	80	amMsgGetDataOffset	103
MQZ_SET_AUTHORITY and		amMsgGetElement	103
MQZ_SET_AUTHORITY_2	80	amMsgGetElementCCSID	103
MQZ_TERM_AUTHORITY	81	amMsgGetElementCount	104
MQZ_TERM_NAME	82	amMsgGetEncoding	104
MQZ_TERM_USERID	82	amMsgGetFilter	104
MQZEP	82	amMsgGetFilterCount	105
		amMsgGetFormat	105
Chapter 7. AMI high-level interface calls	85	amMsgGetGroupStatus	106
amBackout	85	amMsgGetLastError	106
amBegin	85	amMsgGetMsgId	107
amBrowseMsg	86	amMsgGetName	107
amCommit	86	amMsgGetNamedElement	107
amInitialize	87	amMsgGetNamedElementCount	108
amPublish	87	amMsgGetReportCode	108
amReceiveFile	88	amMsgGetTopic	109
amReceiveMsg	88	amMsgGetTopicCount	109
amReceivePublication	89	amMsgGetType	110
amReceiveRequest	90	amMsgReadBytes	110
amSendFile	90	amMsgReset	111
amSendMsg	91	amMsgSetCCSID	111
amSendRequest	91	amMsgSetCorrelId	111
amSendResponse	92	amMsgSetDataOffset	112
amSubscribe	93	amMsgSetElementCCSID	112
amTerminate	93	amMsgSetEncoding	112
amUnsubscribe	94	amMsgSetFormat	113
		amMsgSetGroupStatus	113
		amMsgWriteBytes	113
Chapter 8. AMI object interface calls	95	Policy interface functions	114
Distribution List interface functions	95	amPolClearErrorCodes	114
amDstClearErrorCodes	95	amPolGetLastError	114
amDstClose	95	amPolGetName	114
amDstGetLastError	95	amPolGetWaitTime	115
amDstGetName	96	amPolSetWaitTime	115
amDstGetSenderCount	96	Publisher interface functions	116
amDstGetSenderHandle	97	amPubClearErrorCodes	116
amDstOpen	97	amPubClose	116
amDstSend	98	amPubGetCCSID	116
amDstSendFile	98	amPubGetEncoding	117
Message Interface functions	99	amPubGetLastError	117
amMsgAddElement	99	amPubGetName	118
amMsgAddFilter	99	amPubOpen	118
amMsgAddTopic	100	amPubPublish	119
amMsgClearErrorCodes	100	Receiver interface functions	120
amMsgDeleteElement	100	amRcvBrowse	120
amMsgDeleteFilter	100	amRcvBrowseSelect	120
amMsgDeleteNamedElement	101	amRcvClearErrorCodes	121
amMsgDeleteTopic	101	amRcvClose	122
amMsgGetCCSID	102		

amRcvGetDefnType	122	amSubClearErrorCodes	142
amRcvGetLastError	122	amSubClose	142
amRcvGetName	123	amSubGetCCSID	142
amRcvGetQueueName	124	amSubGetDefnType	143
amRcvOpen	124	amSubGetEncoding	143
amRcvReceive	125	amSubGetLastError	143
amRcvReceiveFile	125	amSubGetName	144
amRcvSetQueueName	126	amSubGetQueueName	145
Session interface functions	127	amSubOpen	145
amSesBegin	127	amSubReceive	145
amSesClearErrorCodes	127	amSubSetQueueName	146
amSesClose	127	amSubSubscribe	146
amSesCommit	128	amSubUnsubscribe	147
amSesCreate	128		
amSesCreateDistList	128	Chapter 9. Elementary data types	149
amSesCreateMessage	129	Elementary data types - C	150
amSesCreatePolicy	129	Elementary data types - COBOL	151
amSesCreatePublisher	129	Elementary data types - PL/I	152
amSesCreateReceiver	130	Elementary data types - System/390 [®]	
amSesCreateSender	130	assembler	153
amSesCreateSubscriber	130	Elementary data types - TAL	154
amSesDelete	131	Elementary data types - Visual Basic	
amSesDeleteDistList	131	(Windows 95, Windows 98, and Windows	
amSesDeleteMessage	131	NT)	155
amSesDeletePolicy	132		
amSesDeletePublisher	132	Chapter 10. Structure data types	157
amSesDeleteReceiver	132	MQBO (begin options)	158
amSesDeleteSender	133	MQCD (Channel data structure)	158
amSesDeleteSubscriber	133	MQCFH (PCF header)	160
amSesGetDistListHandle	133	MQCFIL (PCF integer list parameter)	161
amSesGetLastError	134	MQCFIN (PCF integer parameter)	161
amSesGetMessageHandle	134	MQCFSL (PCF string list parameter)	162
amSesGetPolicyHandle	134	MQCFST (PCF string parameter)	162
amSesGetPublisherHandle	135	MQCIH (CICS bridge header)	162
amSesGetReceiverHandle	135	MQCNO (connect options)	164
amSesGetSenderHandle	135	MQCXP (Channel exit parameter structure)	164
amSesGetSubscriberHandle	136	MQDH (distribution header)	166
amSesOpen	136	MQDLH (dead-letter header)	166
amSesRollback	137	MQDXP (data-conversion exit parameter)	167
Sender interface functions	138	MQGMO (get-message options)	168
amSndClearErrorCodes	138	MQIIIH (IMS information header)	169
amSndClose	138	MQMD (message descriptor)	169
amSndGetCCSID	138	MQMDE (message descriptor extension)	171
amSndGetEncoding	139	MQOD (object descriptor)	172
amSndGetLastError	139	MQOR (object record)	172
amSndGetName	140	MQPMO (put-message options)	173
amSndOpen	140	MQPMR (put-message record)	173
amSndSend	141	MQRFH & MQRFH2 (rules and formatting	
amSndSendFile	141	header)	174
Subscriber interface functions	142	MQRMH (reference-message header)	175

MQRR (response record)	176	MQCCT_* (CICS header conversational task)	202
MQTM (trigger message)	176	MQCD_* (Channel definition structure length)	202
MQTMC (trigger message in character format)	177	MQCD_* (Channel definition structure version)	203
MQTMC2 (trigger message—character format 2).	177	MQCDC_* (Channel data conversion)	203
MQTXP (Transport exit parameter structure)	178	MQCF_* (Channel capability flags)	203
MQWCR (Cluster workload cluster-record structure)	178	MQCFAC_* (CICS header facility)	204
MQWDR (Cluster workload destination-record structure)	178	MQCFH_* (Command format header structure length)	204
MQWIH (Work Information Header)	179	MQCFH_* (Command format header version)	204
MQWQR (Cluster workload queue-record structure)	179	MQCFIL_* (Command format integer-list parameter structure length)	205
MQWXP (Cluster workload exit parameter structure)	180	MQCFIN_* (Command format integer parameter structure)	205
MQXP (API-crossing exit parameter block)	181	MQCFSL_* (Command format string-list parameter structure)	205
MQXQH (transmission-queue header)	181	MQCFST_* (Command format string parameter structure length)	206
MQXWD (Exit wait descriptor structure)	182	MQCFT_* (Command structure type)	206
MQZED (Entity Data structure)	182	MQCFUNC_* (CICS header function name)	206
Chapter 11. Attributes of MQSeries objects	183	MQCGWI_* (CICS header get-wait interval)	207
Local and model queue attributes	183	MQCHAD_* (Channel auto-definition event reporting)	207
Local definition of remote queue attributes	185	MQCHIDS_* (Channel indoubt status)	208
Alias queue attributes	186	MQCHS_* (Channel status)	208
Namelist attributes	186	MQCHSR_* (Channel stop requested)	209
Process definition attributes	187	MQCHT_* (Channel type)	209
Queue manager attributes	187	MQCHTAB_* (Channel table)	210
Chapter 12. MQI constants	189	MQCI_* (Correlation identifier)	210
MQ_* (Lengths of character string and byte fields).	189	MQCIH_* (CICS header flags)	210
MQACT_* (Accounting token)	192	MQCIH_* (CICS header length)	211
MQACT_* (Action option)	192	MQCIH_* (CICS header structure identifier)	211
MQACTT_* (Accounting token type)	193	MQCIH_* (CICS header version)	211
MQAT_* (Application type)	193	MQCLT_* (CICS header link type)	212
MQBND_* (Binding options)	195	MQCMD_* (Command identifier)	212
MQBO_* (Begin options)	195	MQCMDL_* (Command level)	214
MQBO_* (Begin options structure identifier)	195	MQCNO_* (Connect options)	215
MQBO_* (Begin options version).	196	MQCNO_* (Connect options structure identifier)	216
MQCA_* (Character attribute selector)	196	MQCNO_* (Connect options version)	216
MQCACF_* (Character attribute command format parameter)	198	MQCO_* (Close options)	216
MQCACH_* (Channel character attribute command format parameter)	200	MQCODL_* (CICS header output data length)	217
MQCADSD_* (CICS header ADS description).	201	MQCQT_* (Cluster queue type)	217
MQCC_* (Completion code)	201	MQCRC_* (CICS header return code)	218
MQCCSI_* (Coded character set identifier)	202	MQCSC_* (CICS header transaction start code)	218
		MQCT_* (Connection tag)	219

MQCTES_* (CICS header task end status)	219	MQIAV_* (Integer attribute value)	239
MQCUOW_* (CICS header unit-of-work control)	219	MQICM_* (IMS commit mode)	240
MQCXP_* (Channel-exit parameter structure version)	220	MQIDO_* (Indoubt resolution)	240
MQDCC_* (Convert-characters masks and factors)	220	MQIGQ_* (Intra-group queuing)	241
MQDCC_* (Convert-characters option)	221	MQIIH_* (IMS header flags)	241
MQDELO_* (Delete options)	222	MQIIH_* (IMS header length)	241
MQDH_* (Distribution header structure identifier)	222	MQIIH_* (IMS header structure identifier)	242
MQDH_* (Distribution header version)	222	MQIIH_* (IMS header version)	242
MQDHF_* (Distribution header flags)	223	MQISS_* (IMS security scope)	242
MQDL_* (Distribution list support)	223	MQIT_* (Index type)	243
MQDLH_* (Dead-letter header structure identifier)	223	MQITII_* (IMS transaction instance identifier)	243
MQDLH_* (Dead-letter header structure version)	224	MQITS_* (IMS transaction state)	243
MQDT_* (Destination type)	224	MQMCAS_* (MCA status)	244
MQDXP_* (Data-conversion exit identifier)	224	MQMCAT_* (MCA type)	244
MQDXP_* (Data-conversion exit version)	224	MQMD_* (Message descriptor structure identifier)	244
MQEC_* (Signal event control block completion code)	225	MQMD_* (Message descriptor version)	245
MQEI_* (Expiry interval)	225	MQMDE_* (Message descriptor extension length)	245
MQENC_* (Encoding)	225	MQMDE_* (Message descriptor extension structure identifier)	245
MQENC_* (Encoding mask)	226	MQMDE_* (Message descriptor extension version)	246
MQENC_* (Encoding for packed-decimal integers)	226	MQMDEF_* (Message descriptor extension flags)	246
MQENC_* (Encoding for floating-point numbers)	226	MQMDS_* (Message delivery sequence)	246
MQENC_* (Encoding for binary integers)	226	MQMF_* (Message flags)	247
MQET_* (Escape type)	227	MQMF_* (Message flags masks)	247
MQEVR_* (Event reporting)	227	MQMI_* (Message identifier)	247
MQFB_* (Feedback)	227	MQMO_* (Match options)	248
MQFC_* (Force option)	229	MQMT_* (Message type)	248
MQFMT_* (Format)	229	MQMTOK_* (Message token)	249
MQGI_* (Group identifier)	230	MQNC_* (Name count)	249
MQGMO_* (Get-message options)	231	MQNPMS_* (Nonpersistent message speed)	249
MQGMO_* (Get-message options structure identifier)	232	MQOD_* (Object descriptor length)	250
MQGMO_* (Get-message options version)	232	MQOD_* (Object descriptor structure identifier)	250
MQGS_* (Group status)	232	MQOD_* (Object descriptor version)	250
MQHC_* (Connect handle)	233	MQOII_* (Object instance identifier)	251
MQHO_* (Object handle)	233	MQOL_* (Original length)	251
MQIA_* (Integer attribute selector)	233	MQOO_* (Open options)	251
MQIACF_* (Integer attribute command format parameter)	236	MQOT_* (Object type)	252
MQIACH_* (Channel integer attribute command format parameter)	237	MQPA_* (Put authority)	253
MQIAUT_* (IMS authenticator)	239	MQPER_* (Persistence)	254
		MQPL_* (Platform)	254
		MQPMO_* (Put-message options)	255
		MQPMO_* (Put-message options structure length)	255

MQPMO_* (Put-message options structure identifier)	256	MQRO_* (Report-options masks).	286
MQPMO_* (Put-message options structure version)	256	MQRP_* (Replace option)	286
MQPMRF_* (Put-message record field flags)	256	MQRQ_* (Reason qualifier)	287
MQPO_* (Purge option).	257	MQSCO_* (Queue scope)	287
MQPRI_* (Priority)	257	MQSEG_* (Segmentation)	288
MQPS_* (Publish/subscribe tags)	257	MQSID_* (Security identifier)	288
MQPS_* (Publish/subscribe tag commands)	259	MQSIDT_* (Security identifier type).	288
MQPS_* (Publish/subscribe tag options)	260	MQSP_* (Syncpoint)	289
MQPUBO_* (Publication options)	261	MQSS_* (Segment status)	289
MQPXP_* (Publish/subscribe routing exit structure identifier)	261	MQSUS_* (Suspend status)	289
MQPXP_* (Publish/subscribe routing exit version)	261	MQTC_* (Trigger control)	290
MQQA_* (Inhibit get)	262	MQTM_* (Trigger message structure identifier)	290
MQQA_* (Inhibit put)	262	MQTM_* (Trigger message structure version)	290
MQQA_* (Backout hardening)	262	MQTMC_* (Trigger message character format identifier)	291
MQQA_* (Queue shareability)	262	MQTMC_* (Trigger message character format structure)	291
MQQDT_* (Queue definition type)	263	MQTT_* (Trigger type)	291
MQQF_* (Queue flags)	263	MQTXP_* (Transport retry exit structure identifier)	292
MQQMDT_* (Queue-manager definition type).	263	MQTXP_* (Transport retry exit version)	292
MQQMF_* (Queue-manager flags)	264	MQUA_* (User-attribute selectors)	292
MQQMT_* (Queue—manager type).	264	MQUS_* (Usage)	292
MQQO_* (Quiesce option).	264	MQWDR_* (Cluster workload exit destination-record length)	293
MQQSG_* (Queue sharing group disposition).	265	MQWDR_* (Cluster workload exit destination-record structure identifier)	293
MQQSIE_* (Service interval events).	265	MQWDR_* (Cluster workload exit destination-record structure identifier)	293
MQQT_* (Queue type)	266	MQWDR_* (Cluster workload exit destination-record version).	293
MQRC_* (Reason code).	266	MQWI_* (Wait interval).	294
MQRCCF_* (Reason code for command format)	277	MQWIH_* (Workload information header flags)	294
MQREGO_* (Registration options)	282	MQWIH_* (Workload information header structure length)	294
MQRFH_* (Rules and formatting header flags)	282	MQWIH_* (Workload information header structure identifier)	295
MQRFH_* (Rules and formatting header length)	283	MQWIH_* (Workload information header version)	295
MQRFH_* (Rules and formatting header structure identifier)	283	MQWQR_* (Cluster workload exit queue-record length).	295
MQRFH_* (Rules and formatting header version)	283	MQWQR_* (Cluster workload exit queue-record structure identifier)	296
MQRL_* (Returned length).	284	MQWQR_* (Cluster workload exit queue-record version)	296
MQRMH_* (Reference message header structure identifier)	284	MQWXP_* (Cluster workload exit structure identifier)	296
MQRMH_* (Reference message header version)	284	MQWXP_* (Cluster workload exit version)	296
MQRMHF_* (Reference message header flags)	285		
MQRO_* (Report options)	285		

MQXC_* (Exit command identifier)	297	MQZCI_* (Continuation indicator)	304
MQXCC_* (Exit response)	297	MQZED_* (Entity descriptor structure identifier)	305
MQXDR_* (Data-conversion exit response)	298	MQZED_* (Entity descriptor version)	305
MQXP_* (Exit parameter block)	298	MQZID_* (Function identifier, all services)	305
MQXP_* (Exit parameter block version)	298	MQZID_* (Function identifier, authority service)	306
MQXPT_* (Transmission protocol type)	298	MQZID_* (Function identifier, name service)	306
MQXQH_* (Transmission queue header structure identifier)	299	MQZID_* (Function identifier, userid service)	307
MQXQH_* (Transmission queue header structure version)	299	MQZIO_* (Initialization options).	307
MQXR_* (Exit reason)	299	MQZNS_* (Name service version)	307
MQXR2_* (Secondary exit response)	300	MQZTO_* (Termination options).	308
MQXT_* (Exit identifier)	301	MQZUS_* (Userid service version)	308
MQXUA_* (Exit user area).	302		
MQXWD_* (Exit wait descriptor structure identifier)	302	Appendix. Notices	309
MQXWD_* (Exit wait descriptor version)	302	Trademarks	311
MQZAET_* (Authority service entity type)	303		
MQZAO_* (Authority service authorization type)	303	Sending your comments to IBM	313
MQZAS_* (Authority service version)	304		

About this book

This book summarizes the programming interface information that is described fully in the:

- *MQSeries Administration Interface Programming Guide and Reference* manual
- *MQSeries Application Programming Reference* manual
- *MQSeries Application Messaging Interface* manual
- *MQSeries Intercommunication* manual
- *MQSeries Event Monitoring* manual
- *MQSeries Programmable System Management* manual
- *MQSeries Queue Manager Clusters* manual.

It contains summaries of:

- Message queue interface (MQI) calls
- Exit calls
- Event messages
- Programmable Command Format structures
- MQAI calls
- Installable services interface
- AMI calls
- Elementary and structure data types
- Attributes of MQSeries objects
- MQI constants

The information in this book is applicable to all platforms, unless otherwise stated.

Related MQSeries publications

This book specifies the values of all the named constants used in the programming interfaces.

For detailed information about the MQI, see these MQSeries publications:

- *MQSeries Application Programming Guide*, SC33-0807
- *MQSeries Application Programming Reference*, SC33-1673

Each of the following books contains a list of constants used in that book:

- *MQSeries Administration Interface Programming Guide and Reference*, SC34-5390
- *MQSeries Application Programming Reference*, SC33-1673
- *MQSeries Queue Manager Clusters*, SC34-5349
- *MQSeries Intercommunication*, SC33-1872
- *MQSeries Programmable System Management*, SC33-1482

About this book

- *MQSeries Event Monitoring*, SC34-5760

The information in this book does not apply to the MQSeries C++ language or the MQSeries for AS/400 product using the RPG programming language. For RPG information, refer to

- *MQSeries for AS/400 Application Programming Reference (ILE RPG)*, SC34-5558

Terms used in this book

In this book, the term “UNIX[®] systems” refers to the following MQSeries products:

- MQSeries for AIX, V5.1
- MQSeries for AT&T GIS UNIX V2.2
- MQSeries for HP-UX, V5.1
- MQSeries for Compaq Tru64 UNIX, V5.1
- MQSeries for SINIX and DC/OSx, V2.2
- MQSeries for Sun Solaris (SPARC and Intel Platform Editions).

The term “MQSeries Version 5.1 products” applies to the following MQSeries products:

- MQSeries for AIX, V5.1
- MQSeries for AS/400 V5.1
- MQSeries for Compaq Tru64 UNIX, V5.1
- MQSeries for HP-UX, V5.1
- MQSeries for OS/2 Warp, V5.1
- MQSeries for Sun Solaris, V5.1
- MQSeries for Windows NT, V5.1

Windows products

The following table lists the MQSeries products available for Windows, and shows the Windows platforms on which each runs.

Product	Windows 3.1	Windows 95	Windows 98	Windows NT	Windows 2000
MQSeries for Windows NT Client	Yes	Yes	Yes	Yes	Yes
MQSeries for Windows NT	No	No	No	Yes	Yes
MQSeries for Windows V2.0	Yes	Yes	No	No	No
MQSeries for Windows V2.1	No	Yes	Yes	Yes	No

MQSeries for Windows Versions 2.0 and 2.1 support most of the features of the MQI described in this book. For information on these products, see the *MQSeries for Windows User's Guide*.

Appearance of values in this book

In this book:

- The symbol ‘b’ represents a single blank character.
- The value ‘blanks’ denotes the null string in C and blank characters in other programming languages.

About this book

- The notation $X'hhhh'$ represents a hexadecimal value. Each 'h' denotes a single hexadecimal digit.

Summary of changes

This section describes changes in this edition of *Programming Interfaces Reference Summary*. Changes since the previous edition of the book are marked by vertical lines to the left of the changes.

Changes for this edition (SX33-6095-07)

The following changes have been made for this edition of the book.

This is a new book, and replaces the *Application Programming Reference Summary* book. In addition to the contents of that book, the following information is provided in this book:

- Event messages
- Programmable Command Format structures
- MQAI calls
- Installable services interface
- AMI calls

Changes for the first edition (SX33-6095-06)

This edition was not published.

Chapter 1. MQI calls

Full details of these calls can be found in the *MQSeries Application Programming Reference* manual.

MQBACK

Purpose: Indicates to the queue manager that all messages put or retrieved as part of a unit of work since the last syncpoint are to be backed out.

Table 1. MQBACK call

Parameter	Data type	Usage
<i>Hconn</i>	MQHCONN	input
<i>CompCode</i>	MQLONG	output
<i>Reason</i>	MQLONG	output

Note: This call is not supported by VSE/ESA and can be used with OS/390 in the batch environment only. On Tandem NSK, this call always returns a *CompCode* of MQCC_FAILED and a *Reason* of MQRC_ENVIRONMENT_ERROR.

MQBEGIN

Purpose: Begins a unit of work that is coordinated by the queue manager.

Table 2. MQBEGIN call

Parameter	Data type	Usage
<i>Hconn</i>	MQHCONN	input
<i>BeginOptions</i>	MQBO	input/output
<i>CompCode</i>	MQLONG	output
<i>Reason</i>	MQLONG	output

Note: This call is supported only on Version 5 products or later, except AS/400.

MQCLOSE

MQCLOSE

Purpose: Relinquishes access to an object (inverse of MQOPEN).

Table 3. MQCLOSE call

Parameter	Data type	Usage
<i>Hconn</i>	MQHCONN	input
<i>Hobj</i>	MQHOBJ	input/output
<i>Options</i>	MQLONG	input
<i>CompCode</i>	MQLONG	output
<i>Reason</i>	MQLONG	output

MQCMIT

Purpose: Indicates to the queue manager that the application has reached a syncpoint, and that all of the messages put or retrieved as part of a unit of work since the last syncpoint are to be made permanent.

Table 4. MQCMIT call

Parameter	Data type	Usage
<i>Hconn</i>	MQHCONN	input
<i>CompCode</i>	MQLONG	output
<i>Reason</i>	MQLONG	output

Note: This call is not supported by VSE/ESA and can be used with OS/390 in the batch environment only. On Tandem NSK, this call always returns a *CompCode* of MQCC_FAILED and a *Reason* of MQRC_ENVIRONMENT_ERROR.

MQCONN

Purpose: Connects an application program to a queue manager.

Table 5. MQCONN call

Parameter	Data type	Usage
<i>Name</i>	MQCHAR48	input
<i>Hconn</i>	MQHCONN	output
<i>CompCode</i>	MQLONG	output
<i>Reason</i>	MQLONG	output

MQCONN

Purpose: Provides options when connecting an application program to a queue manager.

Table 6. MQCONN call

Parameter	Data type	Usage
<i>Name</i>	MQCHAR48	input
<i>ConnectOpts</i>	MQCNO	input/output
<i>Hconn</i>	MQHCONN	output
<i>CompCode</i>	MQLONG	output
<i>Reason</i>	MQLONG	output
Note: This call is supported on Version 5 products, or later, only.		

MQDISC

Purpose: Disconnects an application program from a queue manager (inverse of MQCONN).

Table 7. MQDISC call

Parameter	Data type	Usage
<i>Hconn</i>	MQHCONN	input/output
<i>CompCode</i>	MQLONG	output
<i>Reason</i>	MQLONG	output

MQGET

MQGET

Purpose: Retrieves a message from a queue owned by the local queue manager.

Table 8. MQGET call

Parameter	Data type	Usage
<i>Hconn</i>	MQHCONN	input
<i>Hobj</i>	MQHOBJ	input
<i>MsgDesc</i>	MQMD	input/output
<i>GetMsgOpts</i>	MQGMO	input/output
<i>BufferLength</i>	MLONG	input
<i>Buffer</i>	MQBYTE x BufferLength	output
<i>DataLength</i>	MLONG	output
<i>CompCode</i>	MLONG	output
<i>Reason</i>	MLONG	output

MQINQ

Purpose: Returns the attributes of an object.

Table 9. MQINQ call

Parameter	Data type	Usage
<i>Hconn</i>	MQHCONN	input
<i>Hobj</i>	MQHOBJ	input
<i>SelectorCount</i>	MLONG	input
<i>Selectors</i>	MLONG x SelectorCount	input
<i>IntAttrCount</i>	MLONG	input
<i>IntAttrs</i>	MLONG x IntAttrCount	output
<i>CharAttrLength</i>	MLONG	input
<i>CharAttrs</i>	MQCHAR x CharAttrLength	output
<i>CompCode</i>	MLONG	output
<i>Reason</i>	MLONG	output

MQOPEN

Purpose: Establishes access to an object.

Table 10. MQOPEN call

Parameter	Data type	Usage
<i>Hconn</i>	MQHCONN	input
<i>ObjDesc</i>	MQOD	input/output
<i>Options</i>	MQLONG	input
<i>Hobj</i>	MQHOBJ	output
<i>CompCode</i>	MQLONG	output
<i>Reason</i>	MQLONG	output

MQPUT

Purpose: Puts a message on an open queue.

Table 11. MQPUT call

Parameter	Data type	Usage
<i>Hconn</i>	MQHCONN	input
<i>Hobj</i>	MQHOBJ	input
<i>MsgDesc</i>	MQMD	input/output
<i>PutMsgOpts</i>	MQPMO	input/output
<i>BufferLength</i>	MQLONG	input
<i>Buffer</i>	MQBYTE x BufferLength	input
<i>CompCode</i>	MQLONG	output
<i>Reason</i>	MQLONG	output

MQPUT1

MQPUT1

Purpose: Puts one message on a queue. This is equivalent to using the sequence of calls: MQOPEN, MQPUT, MQCLOSE.

Table 12. MQPUT1 call

Parameter	Data type	Usage
<i>Hconn</i>	MQHCONN	input
<i>Objdesc</i>	MQOD	input/output
<i>Msgdesc</i>	MQMD	input/output
<i>PutMsgOpts</i>	MQPMO	input/output
<i>BufferLength</i>	MLONG	input
<i>Buffer</i>	MQBYTE x BufferLength	input
<i>CompCode</i>	MLONG	output
<i>Reason</i>	MLONG	output

MQSET

Purpose: Changes the attributes of a queue.

Table 13. MQSET call

Parameter	Data type	Usage
<i>Hconn</i>	MQHCONN	input
<i>Hobj</i>	MQHOBJ	input
<i>SelectorCount</i>	MLONG	input
<i>Selectors</i>	MLONG x SelectorCount	input
<i>IntAttrCount</i>	MLONG	input
<i>IntAttrs</i>	MLONG x IntAttrCount	input
<i>CharAttrLength</i>	MLONG	input
<i>CharAttrs</i>	MQCHAR x CharAttrLength	input
<i>CompCode</i>	MLONG	output
<i>Reason</i>	MLONG	output

MQSYNC

Purpose: Synchronizes statistics updates. This call is included for backwards compatibility but performs no function.

Table 14. MQSYNC call

Parameter	Data type	Usage
<i>TransID</i>	MQLONG x TransID	input
<i>CommitAbort</i>	MQBYTE	input
<i>CompCode</i>	MQLONG	output
<i>Reason</i>	MQLONG	output
Note: This call is supported on Tandem NonStop Kernel only.		

MQSYNC

Chapter 2. Exit calls

A full description of the data-conversion exit calls can be found in the *MQSeries Application Programming Reference* manual.

Note: Data conversion is not supported on VSE/ESA.

A full description of the channel exit calls can be found in the *MQSeries Intercommunication* manual.

A full description of the cluster exit calls can be found in the *MQSeries Queue Manager Clusters* manual.

MQ_CHANNEL_AUTO_DEF_EXIT

Purpose: Provided to describe the parameters that are passed to the channel auto-definition exit called by the Message Channel Agent.

Table 15. MQ_CHANNEL_AUTO_DEF_EXIT call

Parameter	Data type	Usage
<i>ChannelExitParms</i>	MQCXP	input/output
<i>ChannelDefinition</i>	MQCD	input/output

MQ_CHANNEL_EXIT

Purpose: Provided to describe the parameters that are passed to each of the channel exits called by the Message Channel Agent.

Table 16. MQ_CHANNEL_EXIT call

Parameter	Data type	Usage
<i>ChannelExitParms</i>	MQCXP	input/output
<i>ChannelDefinition</i>	MQCD	input/output
<i>DataLength</i>	MQLONG	input/output
<i>AgentBufferLength</i>	MQLONG	input
<i>AgentBuffer</i>	MQBYTE x AgentBufferLength	input/output
<i>ExitBufferLength</i>	MQLONG	input/output
<i>ExitBufferAddr</i>	MQPTR	output

MQ_CLUSTER_WORKLOAD_EXIT

MQ_CLUSTER_WORKLOAD_EXIT

Purpose: Provided to describe the parameters that are passed to the cluster workload exit called by the queue manager.

Table 17. MQ_CLUSTER_WORKLOAD_EXIT call

Parameter	Data type	Usage
<i>ExitParms</i>	MQWXP	input/output

MQ_DATA_CONV_EXIT

Purpose: Describes the parameters that are passed to the data-conversion exit.

Table 18. MQ_DATA_CONV_EXIT call

Parameter	Data type	Usage
<i>DataConvExitParms</i>	MQDXP	input/output
<i>MsgDesc</i>	MQMD	input/output
<i>InBufferLength</i>	MQLONG	input
<i>InBuffer</i>	MQBYTE x <i>InBufferLength</i>	input
<i>OutBufferLength</i>	MQLONG	input
<i>OutBuffer</i>	MQBYTE x <i>OutBufferLength</i>	output

MQ_TRANSPORT_EXIT

Purpose: Provided to describe the parameters that are passed to the transport retry exit called by the Message Channel Agent.

Note: This exit is supported only on AIX and 16-bit Windows.

Table 19. MQ_TRANSPORT_EXIT call

Parameter	Data type	Usage
<i>ExitParms</i>	MQTXP	input/output
<i>DestAddressLength</i>	MQLONG	input
<i>DestAddress</i>	MQCHAR x <i>DestAddressLength</i>	input

MQXCNV

Purpose: Converts characters from one character set to another. This call can be used only from a data-conversion exit.

Table 20. MQXCNV call

Parameter	Data type	Usage
<i>HConn</i>	MQHCONN	input
<i>Options</i>	MQLONG	input
<i>SourceCCSID</i>	MQLONG	input
<i>SourceLength</i>	MQLONG	input
<i>SourceBuffer</i>	MQCHAR x SourceLength	input
<i>TargetCCSID</i>	MQLONG	input
<i>TargetLength</i>	MQLONG	input
<i>TargetBuffer</i>	MQCHAR x TargetLength	output
<i>DataLength</i>	MQLONG	output
<i>CompCode</i>	MQLONG	output
<i>Reason</i>	MQLONG	output

MQXWAIT

Purpose: Provided only for use from a channel exit on OS/390 when not using CICS®. The call waits for an event to occur.

Table 21. MQXWAIT call

Parameter	Data type	Usage
<i>HConn</i>	MQHCONN	input
<i>WaitDesc</i>	MQXWD	input/output
<i>CompCode</i>	MQLONG	output
<i>Reason</i>	MQLONG	output

MQXWAIT

Chapter 3. Event message summary

Full details of these calls can be found in the *MQSeries Event Monitoring* manual.

Alias Base Queue Type Error

The event message contains a reason code parameter with a value of MQRC_ALIAS_BASE_Q_TYPE_ERROR and the event data.

Table 22. Event data summary (Alias Base Queue Type Error)

Parameter	Data type	Returned
<i>QMgrName</i>	MQCFST	Always
<i>QName</i>	MQCFST	Always
<i>BaseQName</i>	MQCFST	Always
<i>QType</i>	MQCFIN	Always
<i>AppLType</i>	MQCFIN	Always
<i>AppLName</i>	MQCFST	Always
<i>ObjectQMgrName</i>	MQCFST	Optionally

Bridge Started

The event message contains a reason code parameter with a value of MQRC_BRIDGE_STARTED and the event data.

Table 23. Event data summary (Bridge Started)

Parameter	Data type	Returned
<i>QMgrName</i>	MQCFST	Always
<i>BridgeType</i>	MQCFIN	Always
<i>BridgeName</i>	MQCFST	Always

Bridge Stopped

Bridge Stopped

The event message contains a reason code parameter with a value of MQRC_BRIDGE_STOPPED and the event data.

Table 24. Event data summary (Bridge Stopped)

Parameter	Data type	Returned
<i>QMgrName</i>	MQCFST	Always
<i>ReasonQualifier</i>	MQCFIN	Always
<i>BridgeType</i>	MQCFIN	Always
<i>BridgeName</i>	MQCFST	Always
<i>ErrorIdentifier</i>	MQCFIN	Optionally

Channel Activated

The event message contains a reason code parameter with a value of MQRC_CHANNEL_ACTIVATED and the event data.

Table 25. Event data summary (Channel Activated)

Parameter	Data type	Returned
<i>QMgrName</i>	MQCFST	Always
<i>ChannelName</i>	MQCFST	Always
<i>XmitQName</i>	MQCFST	Optionally
<i>ConnectionName</i>	MQCFST	Optionally

Channel Auto-definition Error

The event message contains a reason code parameter with a value of MQRC_CHANNEL_AUTO_DEF_ERROR and the event data.

Table 26. Event data summary (Channel Auto-definition Error)

Parameter	Data type	Returned
<i>QMgrName</i>	MQCFST	Always
<i>ChannelName</i>	MQCFST	Always
<i>ChannelType</i>	MQCFIN	Always
<i>ErrorIdentifier</i>	MQCFIN	Always
<i>ConnectionName</i>	MQCFST	Always
<i>AuxErrorDataInt1</i>	MQCFIN	Optionally

Channel Auto-definition OK

The event message contains a reason code parameter with a value of MQRC_CHANNEL_AUTO_DEF_OK and the event data.

Table 27. Event data summary (Channel Auto-definition OK)

Parameter	Data type	Returned
<i>QMgrName</i>	MQCFST	Always
<i>ChannelName</i>	MQCFST	Always
<i>ChannelType</i>	MQCFIN	Always
<i>ConnectionName</i>	MQCFST	Always

Channel Conversion Error

The event message contains a reason code parameter with a value of MQRC_CHANNEL_CONV_ERROR and the event data.

Table 28. Event data summary (Channel Conversion Error)

Parameter	Data type	Returned
<i>QMgrName</i>	MQCFST	Always
<i>ConversionReasonCode</i>	MQCFIN	Always
<i>ChannelName</i>	MQCFST	Always
<i>Format</i>	MQCFST	Always
<i>XmitQName</i>	MQCFST	Optionally
<i>ConnectionName</i>	MQCFST	Optionally

Channel Not Activated

The event message contains a reason code parameter with a value of MQRC_CHANNEL_NOT_ACTIVATED and the event data.

Table 29. Event data summary (Channel Not Activated)

Parameter	Data type	Returned
<i>QMgrName</i>	MQCFST	Always
<i>ChannelName</i>	MQCFST	Always
<i>XmitQName</i>	MQCFST	Optionally
<i>ConnectionName</i>	MQCFST	Optionally

Channel Started

Channel Started

The event message contains a reason code parameter with a value of MQRC_CHANNEL_STARTED and the event data.

Table 30. Event data summary (Channel Started)

Parameter	Data type	Returned
<i>QMgrName</i>	MQCFST	Always
<i>ChannelName</i>	MQCFST	Always
<i>XmitQName</i>	MQCFST	Optionally
<i>ConnectionName</i>	MQCFST	Optionally

Channel Stopped events

For the Channel Stopped event, the event message contains a reason code parameter with a value of MQRC_CHANNEL_STOPPED and the event data.

For the Channel Stopped By User event, the event message contains a reason code parameter with a value of MQRC_CHANNEL_STOPPED_BY_USER and the event data.

Table 31. Event data summary (Channel Stopped)

Parameter	Data type	Returned
<i>QMgrName</i>	MQCFST	Always
<i>ReasonQualifier</i>	MQCFIN	Always
<i>ChannelName</i>	MQCFST	Always
<i>ErrorIdentifier</i>	MQCFIN	Always
<i>AuxErrorDataInt1</i>	MQCFIN	Always
<i>AuxErrorDataInt2</i>	MQCFIN	Always
<i>AuxErrorDataStr1</i>	MQCFST	Always
<i>AuxErrorDataStr2</i>	MQCFST	Always
<i>AuxErrorDataStr3</i>	MQCFST	Always
<i>XmitQName</i>	MQCFST	Optionally
<i>ConnectionName</i>	MQCFST	Optionally

Default Transmission Queue Type Error

The event message contains a reason code parameter with a value of MQRC_DEF_XMIT_Q_TYPE_ERROR and the event data.

Table 32. Event data summary (Default Transmission Queue Type Error)

Parameter	Data type	Returned
<i>QMgrName</i>	MQCFST	Always
<i>QName</i>	MQCFST	Always
<i>XmitQName</i>	MQCFST	Always
<i>QType</i>	MQCFIN	Always
<i>AppLType</i>	MQCFIN	Always
<i>AppLName</i>	MQCFST	Always
<i>ObjectQMgrName</i>	MQCFST	Optionally

Default Transmission Queue Usage Error

The event message contains a reason code parameter with a value of MQRC_DEF_XMIT_Q_USAGE_ERROR and the event data.

Table 33. Event data summary (Default Transmission Queue Usage Error)

Parameter	Data type	Returned
<i>QMgrName</i>	MQCFST	Always
<i>QName</i>	MQCFST	Always
<i>XmitQName</i>	MQCFST	Always
<i>AppLType</i>	MQCFIN	Always
<i>AppLName</i>	MQCFST	Always
<i>ObjectQMgrName</i>	MQCFST	Optionally

Get Inhibited

The event message contains a reason code parameter with a value of MQRC_GET_INHIBITED and the event data.

Table 34. Event data summary (Get Inhibited)

Parameter	Data type	Returned
<i>QMgrName</i>	MQCFST	Always
<i>QName</i>	MQCFST	Always
<i>AppLType</i>	MQCFIN	Always

Get Inhibited

Table 34. Event data summary (Get Inhibited) (continued)

Parameter	Data type	Returned
<i>ApplName</i>	MQCFST	Always

Not Authorized (type 1)

The event message contains a reason code parameter with a value of MQRC_NOT_AUTHORIZED and the event data.

Table 35. Event data summary (Not Authorized (type 1))

Parameter	Data type	Returned
<i>QMgrName</i>	MQCFST	Always
<i>ReasonQualifier</i>	MQCFIN	Always
<i>UserIdentifier</i>	MQCFST	Always
<i>ApplType</i>	MQCFIN	Always
<i>ApplName</i>	MQCFST	Always

Not Authorized (type 2)

The event message contains a reason code parameter with a value of MQRC_NOT_AUTHORIZED and the event data.

Table 36. Event data summary (Not Authorized (type 2))

Parameter	Data type	Returned
<i>QMgrName</i>	MQCFST	Always
<i>ReasonQualifier</i>	MQCFIN	Always
<i>Options</i>	MQCFIN	Always
<i>UserIdentifier</i>	MQCFST	Always
<i>ApplType</i>	MQCFIN	Always
<i>ApplName</i>	MQCFST	Always
<i>ObjectQMgrName</i>	MQCFST	Optionally
<i>QName</i>	MQCFST	Optionally
<i>ProcessName</i>	MQCFST	Optionally

Not Authorized (type 3)

The event message contains a reason code parameter with a value of MQRC_NOT_AUTHORIZED and the event data.

Table 37. Event data summary (Not Authorized (type 3))

Parameter	Data type	Returned
<i>QMgrName</i>	MQCFST	Always
<i>ReasonQualifier</i>	MQCFIN	Always
<i>QName</i>	MQCFST	Always
<i>UserIdentifier</i>	MQCFST	Always
<i>AppIType</i>	MQCFIN	Always
<i>AppIName</i>	MQCFST	Always

Not Authorized (type 4)

The event message contains a reason code parameter with a value of MQRC_NOT_AUTHORIZED and the event data.

Table 38. Event data summary (Not Authorized (type 4))

Parameter	Data type	Returned
<i>QMgrName</i>	MQCFST	Always
<i>ReasonQualifier</i>	MQCFIN	Always
<i>Command</i>	MQCFIN	Always
<i>UserIdentifier</i>	MQCFST	Always

Put Inhibited

The event message contains a reason code parameter with a value of MQRC_PUT_INHIBITED and the event data.

Table 39. Event data summary (Put Inhibited)

Parameter	Data type	Returned
<i>QMgrName</i>	MQCFST	Always
<i>QName</i>	MQCFST	Always
<i>AppIType</i>	MQCFIN	Always
<i>AppIName</i>	MQCFST	Always
<i>ObjectQMgrName</i>	MQCFST	Optionally

Queue Depth High

The event message contains a reason code parameter with a value of MQRC_Q_DEPTH_HIGH and the event data.

Queue Depth High

Table 40. Event data summary (Queue Depth High)

Parameter	Data type	Returned
<i>QMgrName</i>	MQCFST	Always
<i>QName</i>	MQCFST	Always
<i>TimeSinceReset</i>	MQCFIN	Always
<i>HighQDepth</i>	MQCFIN	Always
<i>MsgEnqCount</i>	MQCFIN	Always
<i>MsgDeqCount</i>	MQCFIN	Always

Queue Depth Low

The event message contains a reason code parameter with a value of MQRC_Q_DEPTH_LOW and the event data.

Table 41. Event data summary (Queue Depth Low)

Parameter	Data type	Returned
<i>QMgrName</i>	MQCFST	Always
<i>QName</i>	MQCFST	Always
<i>TimeSinceReset</i>	MQCFIN	Always
<i>HighQDepth</i>	MQCFIN	Always
<i>MsgEnqCount</i>	MQCFIN	Always
<i>MsgDeqCount</i>	MQCFIN	Always

Queue Full

The event message contains a reason code parameter with a value of MQRC_Q_FULL and the event data.

Table 42. Event data summary (Queue Full)

Parameter	Data type	Returned
<i>QMgrName</i>	MQCFST	Always
<i>QName</i>	MQCFST	Always
<i>TimeSinceReset</i>	MQCFIN	Always
<i>HighQDepth</i>	MQCFIN	Always
<i>MsgEnqCount</i>	MQCFIN	Always
<i>MsgDeqCount</i>	MQCFIN	Always

Queue Manager Active

The event message contains a reason code parameter with a value of MQRC_Q_MGR_ACTIVE and the event data.

Table 43. Event data summary (Queue Manager Active)

Parameter	Data type	Returned
<i>QMgrName</i>	MQCFST	Always

Queue Manager Not Active

The event message contains a reason code parameter with a value of MQRC_Q_MGR_NOT_ACTIVE and the event data.

Table 44. Event data summary (Queue Manager Not Active)

Parameter	Data type	Returned
<i>QMgrName</i>	MQCFST	Always
<i>ReasonQualifier</i>	MQCFIN	Always

Queue Service Interval High

The event message contains a reason code parameter with a value of MQRC_Q_SERVICE_INTERVAL_HIGH and the event data.

Table 45. Event data summary (Queue Service Interval High)

Parameter	Data type	Returned
<i>QMgrName</i>	MQCFST	Always
<i>QName</i>	MQCFST	Always
<i>TimeSinceReset</i>	MQCFIN	Always
<i>HighQDepth</i>	MQCFIN	Always
<i>MsgEnqCount</i>	MQCFIN	Always
<i>MsgDeqCount</i>	MQCFIN	Always

Queue Service Interval OK

The event message contains a reason code parameter with a value of MQRC_Q_SERVICE_INTERVAL_OK and the event data.

Queue Service Interval OK

Table 46. Event data summary (Queue Service Interval OK)

Parameter	Data type	Returned
<i>QMgrName</i>	MQCFST	Always
<i>QName</i>	MQCFST	Always
<i>TimeSinceReset</i>	MQCFIN	Always
<i>HighQDepth</i>	MQCFIN	Always
<i>MsgEnqCount</i>	MQCFIN	Always
<i>MsgDeqCount</i>	MQCFIN	Always

Queue Type Error

The event message contains a reason code parameter with a value of MQRC_Q_TYPE_ERROR and the event data.

Table 47. Event data summary (Queue Type Error)

Parameter	Data type	Returned
<i>QMgrName</i>	MQCFST	Always
<i>QName</i>	MQCFST	Always
<i>ApplType</i>	MQCFIN	Always
<i>ApplName</i>	MQCFST	Always
<i>ObjectQMgrName</i>	MQCFST	Optionally

Remote Queue Name Error

The event message contains a reason code parameter with a value of MQRC_REMOTE_Q_NAME_ERROR and the event data.

Table 48. Event data summary (Remote Queue Name Error)

Parameter	Data type	Returned
<i>QMgrName</i>	MQCFST	Always
<i>QName</i>	MQCFST	Always
<i>ApplType</i>	MQCFIN	Always
<i>ApplName</i>	MQCFST	Always
<i>ObjectQMgrName</i>	MQCFST	Optionally

Transmission Queue Type Error

The event message contains a reason code parameter with a value of MQRC_XMIT_Q_TYPE_ERROR and the event data.

Table 49. Event data summary (Transmission Queue Type Error)

Parameter	Data type	Returned
<i>QMgrName</i>	MQCFST	Always
<i>QName</i>	MQCFST	Always
<i>XmitQName</i>	MQCFST	Always
<i>QType</i>	MQCFIN	Always
<i>AppLType</i>	MQCFIN	Always
<i>AppLName</i>	MQCFST	Always
<i>ObjectQMgrName</i>	MQCFST	Optionally

Transmission Queue Usage Error

The event message contains a reason code parameter with a value of MQRC_XMIT_Q_USAGE_ERROR and the event data.

Table 50. Event data summary (Transmission Queue Usage Error)

Parameter	Data type	Returned
<i>QMgrName</i>	MQCFST	Always
<i>QName</i>	MQCFST	Always
<i>XmitQName</i>	MQCFST	Always
<i>AppLType</i>	MQCFIN	Always
<i>AppLName</i>	MQCFST	Always
<i>ObjectQMgrName</i>	MQCFST	Optionally

Unknown Alias Base Queue

The event message contains a reason code parameter with a value of MQRC_UNKNOWN_ALIAS_BASE_Q and the event data.

Table 51. Event data summary (Unknown Alias Base Queue)

Parameter	Data type	Returned
<i>QMgrName</i>	MQCFST	Always
<i>QName</i>	MQCFST	Always
<i>BaseQName</i>	MQCFST	Always

Unknown Alias Base Queue

Table 51. Event data summary (Unknown Alias Base Queue) (continued)

Parameter	Data type	Returned
<i>ApplType</i>	MQCFIN	Always
<i>ApplName</i>	MQCFST	Always
<i>ObjectQMgrName</i>	MQCFST	Optionally

Unknown Default Transmission Queue

The event message contains a reason code parameter with a value of MQRC_UNKNOWN_DEF_XMIT_Q and the event data.

Table 52. Event data summary (Unknown Default Transmission Queue)

Parameter	Data type	Returned
<i>QMgrName</i>	MQCFST	Always
<i>QName</i>	MQCFST	Always
<i>XmitQName</i>	MQCFST	Always
<i>ApplType</i>	MQCFIN	Always
<i>ApplName</i>	MQCFST	Always
<i>ObjectQMgrName</i>	MQCFST	Optionally

Unknown Object Name

The event message contains a reason code parameter with a value of MQRC_UNKNOWN_OBJECT_NAME and the event data.

Table 53. Event data summary (Unknown Object Name)

Parameter	Data type	Returned
<i>QMgrName</i>	MQCFST	Always
<i>ApplType</i>	MQCFIN	Always
<i>ApplName</i>	MQCFST	Always
<i>QName</i> (1)	MQCFST	Always
<i>ProcessName</i> (1)	MQCFST	Always
<i>ObjectQMgrName</i>	MQCFST	Optionally

Note:

1. Only one of these parameters is used.

Unknown Remote Queue Manager

The event message contains a reason code parameter with a value of MQRC_UNKNOWN_REMOTE_Q_MGR and the event data.

Table 54. Event data summary (Unknown Remote Queue Manager)

Parameter	Data type	Returned
<i>QMgrName</i>	MQCFST	Always
<i>QName</i>	MQCFST	Always
<i>AppLType</i>	MQCFIN	Always
<i>AppLName</i>	MQCFST	Always
<i>ObjectQMgrName</i>	MQCFST	Optionally

Unknown Transmission Queue

The event message contains a reason code parameter with a value of MQRC_UNKNOWN_XMIT_Q and the event data.

Table 55. Event data summary (Unknown Transmission Queue)

Parameter	Data type	Returned
<i>QMgrName</i>	MQCFST	Always
<i>QName</i>	MQCFST	Always
<i>XmitQName</i>	MQCFST	Always
<i>AppLType</i>	MQCFIN	Always
<i>AppLName</i>	MQCFST	Always
<i>ObjectQMgrName</i>	MQCFST	Optionally

Unknown Transmission Queue

Chapter 4. Programmable Command Format (PCF) structures

Full details of these calls can be found in the *MQSeries Programmable System Management* manual.

Platform restrictions:

Unless specified these PCFs are supported on all platforms.

Change, Copy, and Create Channel

Purpose:

The Change Channel (MQCMD_CHANGE_CHANNEL) command changes the specified attributes in a channel definition.

The Copy Channel (MQCMD_COPY_CHANNEL) command creates a new channel definition using, for attributes not specified in the command, the attribute values of an existing channel definition.

The Create Channel (MQCMD_CREATE_CHANNEL) command creates an MQSeries channel definition. Any attributes that are not defined explicitly are set to the default values on the destination queue manager.

Required parameters (Change Channel)

Table 56. Required parameters (Change Channel)

Parameter	Data type	Parameter identifier
<i>ChannelName</i>	MQCFST	MQCACH_CHANNEL_NAME
<i>ChannelType</i>	MQCFIN	MQCACH_CHANNEL_TYPE

Required parameters (Copy Channel and Create Channel)

Table 57. Required parameters (Copy and Create Channel)

Parameter	Data type	Parameter identifier
<i>FromChannelName</i>	MQCFST	MQCACH_FROM_CHANNEL_NAME
<i>ToChannelName</i>	MQCFST	MQCACH_TO_CHANNEL_NAME
<i>ChannelType</i>	MQCFIN	MQCACH_CHANNEL_TYPE

Channel commands

Optional parameters (sender or server channel)

Table 58. Optional parameters (sender or server channel)

Parameter	Data type	Parameter identifier
<i>Replace</i> (1)	MQCFIN	MQIACF_REPLACE
<i>TransportType</i>	MQCFIN	MQIACH_XMIT_PROTOCOL_TYPE
<i>ChannelDesc</i>	MQCFST	MQCACH_DESC
<i>SecurityExit</i>	MQCFST	MQCACH_SEC_EXIT_NAME
<i>MsgExit</i>	MQCFST	MQCACH_MSG_EXIT_NAME
<i>SendExit</i>	MQCFST	MQCACH_SEND_EXIT_NAME
<i>ReceiveExit</i>	MQCFST	MQCACH_RCV_EXIT_NAME
<i>MaxMsgLength</i>	MQCFIN	MQIACH_MAX_MSG_LENGTH
<i>SecurityUserData</i>	MQCFST	MQCACH_SEC_EXIT_USER_DATA
<i>MsgUserData</i>	MQCFST	MQCACH_MSG_EXIT_USER_DATA
<i>SendUserData</i>	MQCFST	MQCACH_SEND_EXIT_USER_DATA
<i>ReceiveUserData</i>	MQCFST	MQCACH_RCV_EXIT_USER_DATA
<i>ModeName</i>	MQCFST	MQCACH_MODE_NAME
<i>TpName</i>	MQCFST	MQCACH_TP_NAME
<i>ConnectionName</i>	MQCFST	MQCACH_CONNECTION_NAME
<i>XmitQName</i>	MQCFST	MQCACH_XMIT_Q_NAME
<i>MCAName</i>	MQCFST	MQCACH_MCA_NAME
<i>BatchSize</i>	MQCFIN	MQIACH_BATCH_SIZE
<i>DscInterval</i>	MQCFIN	MQIACH_DISC_INTERVAL
<i>ShortRetryCount</i>	MQCFIN	MQIACH_SHORT_RETRY
<i>ShortRetryInterval</i>	MQCFIN	MQIACH_SHORT_TIMER
<i>LongRetryCount</i>	MQCFIN	MQIACH_LONG_RETRY
<i>LongRetryInterval</i>	MQCFIN	MQIACH_LONG_TIMER
<i>DataConversion</i>	MQCFIN	MQIACH_DATA_CONVERSION
<i>SeqNumberWrap</i>	MQCFIN	MQIACH_SEQUENCE_NUMBER_WRAP
<i>MCAType</i>	MQCFIN	MQIACH_MCA_TYPE
<i>MCAUserIdentifier</i>	MQCFIN	MQIACH_MCA_USER_ID
<i>UserIdentifier</i>	MQCFST	MQIACH_USER_ID
<i>Password</i>	MQCFST	MQCACH_PASSWORD
<i>HeartbeatInterval</i>	MQCFIN	MQIACH_HB_INTERVAL
<i>NonPersistentMsgSpeed</i>	MQCFIN	MQIACH_NPM_SPEED

Table 58. Optional parameters (sender or server channel) (continued)

Parameter	Data type	Parameter identifier
<i>BatchInterval</i>	MQCFIN	MQIACH_BATCH_INTERVAL
Note::		
1. This parameter applies to Copy Channel and Create Channel only.		

Optional parameters (receiver channel)

Table 59. Optional parameters (receiver channel)

Parameter	Data type	Parameter identifier
<i>Replace</i> (1)	MQCFIN	MQIACF_REPLACE
<i>TransportType</i>	MQCFIN	MQIACH_XMIT_PROTOCOL_TYPE
<i>ChannelDesc</i>	MQCFST	MQCACH_DESC
<i>SecurityExit</i>	MQCFST	MQCACH_SEC_EXIT_NAME
<i>MsgExit</i>	MQCFST	MQCACH_MSG_EXIT_NAME
<i>SendExit</i>	MQCFST	MQCACH_SEND_EXIT_NAME
<i>ReceiveExit</i>	MQCFST	MQCACH_RCV_EXIT_NAME
<i>MaxMsgLength</i>	MQCFIN	MQCACH_MAX_MSG_LENGTH
<i>SecurityUserData</i>	MQCFST	MQCACH_SEC_EXIT_USER_DATA
<i>MsgUserData</i>	MQCFST	MQCACH_MSG_EXIT_USER_DATA
<i>SendUserData</i>	MQCFST	MQCACH_SEND_EXIT_USER_DATA
<i>ReceiveUserData</i>	MQCFST	MQCACH_RCV_EXIT_USER_DATA
<i>BatchSize</i>	MQCFIN	MQIACH_BATCH_SIZE
<i>PutAuthority</i>	MQCFIN	MQIACH_PUT_AUTHORITY
<i>SeqNumberWrap</i>	MQCFIN	MQIACH_SEQUENCE_NUMBER_WRAP
<i>MCAUserIdentifier</i>	MQCFIN	MQCACH_MCA_USER_ID
<i>MsgRetryExit</i>	MQCFST	MQCACH_MR_EXIT_NAME
<i>MsgRetryUserData</i>	MQCFST	MQCACH_MR_EXIT_USER_DATA
<i>MsgRetryCount</i>	MQCFIN	MQIACH_MR_COUNT
<i>MsgRetryInterval</i>	MQCFIN	MQIACH_MR_INTERVAL
<i>HeartbeatInterval</i>	MQCFIN	MQIACH_HB_INTERVAL
<i>NonPersistentMsgSpeed</i>	MQCFIN	MQIACH_NPM_SPEED
Note::		
1. This parameter applies to Copy Channel and Create Channel only.		

Channel commands

Optional parameters (requester channel)

Table 60. Optional parameters (requester channel)

Parameter	Data type	Parameter identifier
<i>Replace</i> (1)	MQCFIN	MQIACF_REPLACE
<i>TransportType</i>	MQCFIN	MQIACH_XMIT_PROTOCOL_TYPE
<i>ChannelDesc</i>	MQCFST	MQCACH_DESC
<i>SecurityExit</i>	MQCFST	MQCACH_SEC_EXIT_NAME
<i>MsgExit</i>	MQCFST	MQCACH_MSG_EXIT_NAME
<i>SendExit</i>	MQCFST	MQCACH_SEND_EXIT_NAME
<i>ReceiveExit</i>	MQCFST	MQCACH_RCV_EXIT_NAME
<i>MaxMsgLength</i>	MQCFIN	MQIACH_MAX_MSG_LENGTH
<i>SecurityUserData</i>	MQCFST	MQCACH_SEC_EXIT_USER_DATA
<i>MsgUserData</i>	MQCFST	MQCACH_MSG_EXIT_USER_DATA
<i>SendUserData</i>	MQCFST	MQCACH_SEND_EXIT_USER_DATA
<i>ReceiveUserData</i>	MQCFST	MQCACH_RCV_EXIT_USER_DATA
<i>ModeName</i>	MQCFST	MQCACH_MODE_NAME
<i>TpName</i>	MQCFST	MQCACH_TP_NAME
<i>ConnectionName</i>	MQCFST	MQCACH_CONNECTION_NAME
<i>MCAName</i>	MQCFST	MQCACH_MCA_NAME
<i>BatchSize</i>	MQCFIN	MQIACH_BATCH_SIZE
<i>PutAuthority</i>	MQCFIN	MQIACH_PUT_AUTHORITY
<i>SeqNumberWrap</i>	MQCFIN	MQIACH_SEQUENCE_NUMBER_WRAP
<i>MCAType</i>	MQCFIN	MQIACH_MCA_TYPE
<i>MCAUserIdentifier</i>	MQCFIN	MQCACH_MCA_USED_ID
<i>UserIdentifier</i>	MQCFST	MQCACH_USER_ID
<i>Password</i>	MQCFST	MQCACH_PASSWORD
<i>MsgRetryExit</i>	MQCFST	MQCACH_MR_EXIT_NAME
<i>MsgRetryUserData</i>	MQCFST	MQCACH_MR_EXIT_USER_DATA
<i>MsgRetryCount</i>	MQCFIN	MQIACH_MR_COUNT
<i>MsgRetryInterval</i>	MQCFIN	MQIACH_MR_INTERVAL
<i>HeartbeatInterval</i>	MQCFIN	MQIACH_HB_INTERVAL
<i>NonPersistentMsgSpeed</i>	MQCFIN	MQIACH_NPM_SPEED

Note::

1. This parameter applies to Copy Channel and Create Channel only.

Optional parameters (server-connection channel)

Table 61. Optional parameters (server-connection channel)

Parameter	Data type	Parameter identifier
<i>Replace (1)</i>	MQCFIN	MQIACF_REPLACE
<i>TransportType</i>	MQCFIN	MQIACH_XMIT_PROTOCOL_TYPE
<i>ChannelDesc</i>	MQCFST	MQCACH_DESC
<i>SecurityExit</i>	MQCFST	MQCACH_SEC_EXIT_NAME
<i>MsgExit</i>	MQCFST	MQCACH_MSG_EXIT_NAME
<i>SendExit</i>	MQCFST	MQCACH_SEND_EXIT_NAME
<i>ReceiveExit</i>	MQCFST	MQCACH_RCV_EXIT_NAME
<i>MaxMsgLength</i>	MQCFIN	MQIACH_MAX_MSG_LENGTH
<i>SecurityUserData</i>	MQCFST	MQCACH_SEC_EXIT_USER_DATA
<i>MsgUserData</i>	MQCFST	MQCACH_MSG_EXIT_USER_DATA
<i>SendUserData</i>	MQCFST	MQCACH_SEND_EXIT_USER_DATA
<i>ReceiveUserData</i>	MQCFST	MQCACH_RCV_EXIT_USER_DATA
<i>MCAUserIdentifier</i>	MQCFIN	MQCACH_MCA_USER_ID
Note::		
1. This parameter applies to Copy Channel and Create Channel only.		

Optional parameters (client-connection channel)

Table 62. Optional parameters (client-connection channel)

Parameter	Data type	Parameter identifier
<i>Replace (1)</i>	MQCFIN	MQIACF_REPLACE
<i>TransportType</i>	MQCFIN	MQIACH_XMIT_PROTOCOL_TYPE
<i>ChannelDesc</i>	MQCFST	MQCACH_DESC
<i>SecurityExit</i>	MQCFST	MQCACH_SEC_EXIT_NAME
<i>MsgExit</i>	MQCFST	MQCACH_MSG_EXIT_NAME
<i>SendExit</i>	MQCFST	MQCACH_SEND_EXIT_NAME
<i>ReceiveExit</i>	MQCFST	MQCACH_RCV_EXIT_NAME
<i>MaxMsgLength</i>	MQCFIN	MQIACH_MAX_MSG_LENGTH
<i>SecurityUserData</i>	MQCFST	MQCACH_SEC_EXIT_USER_DATA
<i>MsgUserData</i>	MQCFST	MQCACH_MSG_EXIT_USER_DATA
<i>SendUserData</i>	MQCFST	MQCACH_SEND_EXIT_USER_DATA
<i>ReceiveUserData</i>	MQCFST	MQCACH_RCV_EXIT_USER_DATA

Channel commands

Table 62. Optional parameters (client-connection channel) (continued)

Parameter	Data type	Parameter identifier
<i>ModeName</i>	MQCFST	MQCACH_MODE_NAME
<i>TpName</i>	MQCFST	MQCACH_TP_NAME
<i>QMgrName</i>	MQCFST	MQCACH_Q_MGR_NAME
<i>ConnectionName</i>	MQCFST	MQCACH_Q_MGR_NAME
<i>UserIdentifier</i>	MQCFST	MQCACH_USER_ID
<i>Password</i>	MQCFST	MQCACH_PASSWORD
Note::		
1. This parameter applies to Copy Channel and Create Channel only.		

Optional parameters (cluster-receiver channel)

Table 63. Optional parameters (cluster-receiver channel)

Parameter	Data type	Parameter identifier
<i>Replace</i> (1)	MQCFIN	MQIACF_REPLACE
<i>TransportType</i>	MQCFIN	MQIACH_XMIT_PROTOCOL_TYPE
<i>ChannelDesc</i>	MQCFST	MQCACH_DESC
<i>SecurityExit</i>	MQCFST	MQCACH_SEC_EXIT_NAME
<i>MsgExit</i>	MQCFST	MQCACH_MSG_EXIT_NAME
<i>SendExit</i>	MQCFST	MQCACH_SEND_EXIT_NAME
<i>ReceiveExit</i>	MQCFST	MQCACH_RCV_EXIT_NAME
<i>MaxMsgLength</i>	MQCFIN	MQIACH_MAX_MSG_LENGTH
<i>SecurityUserData</i>	MQCFST	MQCACH_SEC_EXIT_USER_DATA
<i>MsgUserData</i>	MQCFST	MQCACH_MSG_EXIT_USER_DATA
<i>SendUserData</i>	MQCFST	MQCACH_SEND_EXIT_USER_DATA
<i>ReceiveUserData</i>	MQCFST	MQCACH_RCV_EXIT_USER_DATA
<i>ModeName</i>	MQCFST	MQCACH_MODE_NAME
<i>TpName</i>	MQCFST	MQCACH_TP_NAME
<i>ConnectionName</i>	MQCFST	MQCACH_CONNECTION_NAME
<i>DscInterval</i>	MQCFIN	MQIACH_DISC_INTERVAL
<i>ShortRetryCount</i>	MQCFIN	MQIACH_SHORT_RETRY
<i>ShortRetryInterval</i>	MQCFIN	MQIACH_SHORT_TIMER
<i>LongRetryCount</i>	MQCFIN	MQIACH_LONG_RETRY
<i>LongRetryInterval</i>	MQCFIN	MQIACH_LONG_TIMER
<i>DataConversion</i>	MQCFIN	MQIACH_DATA_CONVERSION

Table 63. Optional parameters (cluster-receiver channel) (continued)

Parameter	Data type	Parameter identifier
<i>BatchSize</i>	MQCFIN	MQIACH_BATCH_SIZE
<i>PutAuthority</i>	MQCFIN	MQIACH_PUT_AUTHORITY
<i>SeqNumberWrap</i>	MQCFIN	MQIACH_SEQUENCE_NUMBER_WRAP
<i>MCAUserIdentifier</i>	MQCFIN	MQCACH_MCA_USER_ID
<i>MsgRetryExit</i>	MQCFST	MQCACH_MR_EXIT_NAME
<i>MsgRetryUserData</i>	MQCFST	MQCACH_MR_EXIT_USER_DATA
<i>MsgRetryCount</i>	MQCFIN	MQIACH_MR_COUNT
<i>MsgRetryInterval</i>	MQCFIN	MQIACH_MR_INTERVAL
<i>HeartbeatInterval</i>	MQCFIN	MQIACH_HB_INTERVAL
<i>NonPersistentMsgSpeed</i>	MQCFIN	MQIACH_NPM_SPEED
<i>BatchInterval</i>	MQCFIN	MQIACH_BATCH_INTERVAL
<i>ClusterName</i>	MQCFST	MQCA_CLUSTER_NAME
<i>ClusterNameList</i>	MQCFST	MQCA_CLUSTER_NAMELIST
<i>NetworkPriority</i>	MQCFIN	MQIACH_NETWORK_PRIORITY
Note::		
1. This parameter applies to Copy Channel and Create Channel only.		

Optional parameters (cluster-sender channel)

Table 64. Optional parameters (cluster-sender channel)

Parameter	Data type	Parameter identifier
<i>Replace (1)</i>	MQCFIN	MQIACF_REPLACE
<i>TransportType</i>	MQCFIN	MQIACH_XMIT_PROTOCOL_TYPE
<i>ChannelDesc</i>	MQCFST	MQCACH_DESC
<i>SecurityExit</i>	MQCFST	MQCACH_SEC_EXIT_NAME
<i>MsgExit</i>	MQCFST	MQCACH_MSG_EXIT_NAME
<i>SendExit</i>	MQCFST	MQCACH_SEND_EXIT_NAME
<i>ReceiveExit</i>	MQCFST	MQCACH_RCV_EXIT_NAME
<i>MaxMsgLength</i>	MQCFIN	MQIACH_MAX_MSG_LENGTH
<i>SecurityUserData</i>	MQCFST	MQCACH_SEC_EXIT_USER_DATA
<i>MsgUserData</i>	MQCFST	MQCACH_MSG_EXIT_USER_DATA
<i>SendUserData</i>	MQCFST	MQCACH_SEND_EXIT_USER_DATA
<i>ReceiveUserData</i>	MQCFST	MQCACH_RCV_EXIT_USER_DATA
<i>ModeName</i>	MQCFST	MQCACH_MODE_NAME

Channel commands

Table 64. Optional parameters (cluster-sender channel) (continued)

Parameter	Data type	Parameter identifier
<i>TpName</i>	MQCFST	MQCACH_TP_NAME
<i>ConnectionName</i>	MQCFST	MQCACH_CONNECTION_NAME
<i>MCAName</i>	MQCFST	MQCACH_MCA_NAME
<i>BatchSize</i>	MQCFIN	MQIACH_BATCH_SIZE
<i>DscInterval</i>	MQCFIN	MQIACH_DISC_INTERVAL
<i>ShortRetryCount</i>	MQCFIN	MQIACH_SHORT_RETRY
<i>ShortRetryInterval</i>	MQCFIN	MQIACH_SHORT_TIMER
<i>LongRetryCount</i>	MQCFIN	MQIACH_LONG_RETRY
<i>LongRetryInterval</i>	MQCFIN	MQIACH_LONG_TIMER
<i>SeqNumberWrap</i>	MQCFIN	MQIACH_SEQUENCE_NUMBER_WRAP
<i>DataConversion</i>	MQCFIN	MQIACH_DATA_CONVERSION
<i>MCAType</i>	MQCFIN	MQIACH_MCA_TYPE
<i>MCAUserIdentifier</i>	MQCFIN	MQCACH_MCA_USER_ID
<i>UserIdentifier</i>	MQCFST	MQCACH_USER_ID
<i>Password</i>	MQCFST	MQCACH_PASSWORD
<i>HeartbeatInterval</i>	MQCFIN	MQIACH_HB_INTERVAL
<i>NonPersistentMsgSpeed</i>	MQCFIN	MQIACH_NPM_SPEED
<i>BatchInterval</i>	MQCFIN	MQIACH_BATCH_INTERVAL
<i>ClusterName</i>	MQCFST	MQCA_CLUSTER_NAME
<i>ClusterNamelist</i>	MQCFST	MQCA_CLUSTER_NAMELIST
Note:: 1. This parameter applies to Copy Channel and Create Channel only.		

Change, Copy, and Create Namelist commands

Purpose:

The Change Namelist (MQCMD_CHANGE_NAMELIST) command changes the specified attributes of an existing MQSeries namelist definition.

The Copy Namelist (MQCMD_COPY_NAMELIST) command creates a new MQSeries Namelist definition using, for attributes not specified in the command, the attribute values of an existing namelist definition.

The Create Namelist (MQCMD_CREATE_NAMELIST) command creates an MQSeries namelist definition. Any attributes that are not defined explicitly are set to the default values on the destination queue manager.

Platforms:

These PCFs are supported if you are using AIX, HP-UX, OS/2, OS/400, Sun Solaris, or Windows NT.

Required parameters (Change and Create Namelist)

Table 65. Required parameters (Change Namelist)

Parameter	Data type	Parameter identifier
<i>NamelistName</i>	MQCFST	MQCA_NAMELIST_NAME

Required parameters (Copy Namelist)

Table 66. Required parameters (Copy Namelist)

Parameter	Data type	Parameter identifier
<i>FromNameListName</i>	MQCFST	MQCACF_FROM_NAMELIST_NAME
<i>ToNameListName</i>	MQCFST	MQCACF_TO_NAMELIST_NAME

Optional parameters

Table 67. Optional parameters (Change, Copy and Create Namelist)

Parameter	Data type	Parameter identifier
<i>Replace</i> (1)	MQCFIN	MQIACF_REPLACE
<i>NamelistDesc</i>	MQCFST	MQCA_NAMELIST_DESC
<i>Names</i>	MQCFSL	MQCA_NAMES
Note::		
1. This parameter applies to Copy and Create Namelist only.		

Change, Copy, and Create Process commands

Purpose:

The Change Process (MQCMD_CHANGE_PROCESS) command changes the specified attributes of an existing MQSeries process definition.

The Copy Process (MQCMD_COPY_PROCESS) command creates a new MQSeries Process definition using, for attributes not specified in the command, the attribute values of an existing process definition.

Change, Copy, Create Process

The Create Process (MQCMD_CREATE_PROCESS) command creates an MQSeries process definition. Any attributes that are not defined explicitly are set to the default values on the destination queue manager.

Platforms:

These PCFs are not supported if you are using MQSeries for Windows Version 2.1.

Required parameters (Change and Create Process)

Table 68. Required parameters (Change Channel)

Parameter	Data type	Parameter identifier
<i>ProcessName</i>	MQCFST	MQCA_PROCESS_NAME

Required parameters (Copy Process)

Table 69. Required parameters (Change Channel)

Parameter	Data type	Parameter identifier
<i>FromProcessName</i>	MQCFST	MQCA_FROM_PROCESS_NAME
<i>ToProcessName</i>	MQCFST	MQCA_TO_PROCESS_NAME

Optional parameters

Table 70. Required parameters (Change Channel)

Parameter	Data type	Parameter identifier
<i>Replace</i> (1)	MQCFIN	MQIACF_REPLACE
<i>ProcessDesc</i>	MQCFST	MQCA_PROCESS_DESC
<i>ApplType</i>	MQCFIN	MQIA_APPL_TYPE
<i>ApplId</i>	MQCFST	MQCA_APPL_ID
<i>EnvData</i>	MQCFST	MQCA_ENV_DATA
<i>UserData</i>	MQCFST	MQCA_USER_DATA

Note:
1. This parameter applies to Copy Process only.

Change, Copy, and Create Queue commands

Purpose:

The Change Queue (MQCMD_CHANGE_Q) command changes the specified attributes of an existing MQSeries queue.

The Copy Queue (MQCMD_COPY_Q) command creates a new queue definition using, for attributes not specified in the command, the attribute values of an existing queue definition.

The Create Queue (MQCMD_CREATE_Q) command creates an MQSeries queue definition. Any attributes that are not defined explicitly are set to the default values for the queue type being created.

Required parameters (Change Queue)

Table 71. Required parameters (Change Channel)

Parameter	Data type	Parameter identifier
<i>QName</i>	MQCFST	MQCA_Q_NAME
<i>QType</i>	MQCFIN	MQIA_Q_TYPE

Required parameters (Copy Queue and Create Queue)

Table 72. Required parameters (Change Channel)

Parameter	Data type	Parameter identifier
<i>FromQName</i>	MQCFST	MQCA_FROM_Q_NAME
<i>ToQName</i>	MQCFST	MQCA_TO_Q_NAME
<i>QType</i>	MQCFIN	MQIA_Q_TYPE

Optional parameters (alias queue)

Table 73. Required parameters (Change Channel)

Parameter	Data type	Parameter identifier
<i>QDesc</i>	MQCFST	MQCA_Q_DESC
<i>InhibitPut</i>	MQCFIN	MQIA_INHIBIT_PUT
<i>DefPriority</i>	MQCFIN	MQIA_DEF_PRIORITY
<i>DefPersistence</i>	MQCFIN	MQIA_DEF_PERSISTENCE
<i>Force</i> (1)	MQCFIN	MQIACF_FORCE
<i>InhibitGet</i>	MQCFIN	MQIA_INHIBIT_GET
<i>BaseQName</i>	MQCFST	MQCA_BASE_Q_NAME
<i>Scope</i>	MQCFIN	MQIA_SCOPE
<i>ClusterName</i>	MQCFST	MQCA_CLUSTER_NAME
<i>ClusterNameList</i>	MQCFST	MQCA_CLUSTER_NAMELIST
<i>DefBind</i>	MQCFIN	

Note::

1. This parameter applies to Change Queue only.

Queue commands

Optional parameters (local queue)

Table 74. Required parameters (Change Channel)

Parameter	Data type	Parameter identifier
<i>QDesc</i>	MQCFST	MQCA_Q_DESC
<i>InhibitPut</i>	MQCFIN	MQIA_INHIBIT_PUT
<i>DefPriority</i>	MQCFIN	MQIA_DEF_PRIORITY
<i>DefPersistence</i>	MQCFIN	MQIA_DEF_PERSISTENCE
<i>Force (1)</i>	MQCFIN	MQIACF_FORCE
<i>InhibitGet</i>	MQCFIN	MQIA_INHIBIT_GET
<i>ProcessName</i>	MQCFST	MQCA_PROCESS_NAME
<i>MaxQDepth</i>	MQCFIN	MQIA_MAX_Q_DEPTH
<i>MaxMsgLength</i>	MQCFIN	MQIA_MAX_MSG_LENGTH
<i>BackoutThreshold</i>	MQCFIN	MQIA_BACKOUT_THRESHOLD
<i>BackoutRequeueName</i>	MQCFST	MQCA_BACKOUT_REQ_Q_NAME
<i>Shareability</i>	MQCFIN	MQIA_SHAREABILITY
<i>DefInputOpenOption</i>	MQCFIN	MQIA_DEF_INPUT_OPEN_OPTION
<i>HardenGetBackout</i>	MQCFIN	MQIA_HARDEN_GET_BACKOUT
<i>MsgDeliverySequence</i>	MQCFIN	MQIA_MSG_DELIVERY_SEQUENCE
<i>RetentionInterval</i>	MQCFIN	MQIA_RETENTION_INTERVAL
<i>DistLists</i>	MQCFIN	MQIA_DIST_LISTS
<i>Usage</i>	MQCFIN	MQIA_USAGE
<i>InitiationQName</i>	MQCFST	MQCA_INITIATION_Q_NAME
<i>TriggerControl</i>	MQCFIN	MQIA_TRIGGER_CONTROL
<i>TriggerType</i>	MQCFIN	MQIA_TRIGGER_TYPE
<i>TriggerMsgPriority</i>	MQCFIN	MQIA_TRIGGER_MSG_PRIORITY
<i>TriggerDepth</i>	MQCFIN	MQIA_TRIGGER_DEPTH
<i>TriggerData</i>	MQCFST	MQIA_TRIGGER_DATA
<i>Scope</i>	MQCFIN	MQIA_SCOPE
<i>QDepthHighLimit</i>	MQCFIN	MQIA_Q_DEPTH_HIGH_LIMIT
<i>QDepthLowLimit</i>	MQCFIN	MQIA_Q_DEPTH_LOW_LIMIT
<i>QDepthMaxEvent</i>	MQCFIN	MQIA_Q_DEPTH_MAX_EVENT
<i>QDepthHighEvent</i>	MQCFIN	MQIA_Q_DEPTH_HIGH_EVENT
<i>QDepthLowEvent</i>	MQCFIN	MQIA_Q_DEPTH_LOW_EVENT

Table 74. Required parameters (Change Channel) (continued)

Parameter	Data type	Parameter identifier
<i>QServiceInterval</i>	MQCFIN	MQIA_Q_SERVICE_INTERVAL
<i>QServiceIntervalEvent</i>	MQCFIN	MQIA_Q_SERVICE_INTERVAL_EVENT
<i>ClusterName</i>	MQCFST	MQCA_CLUSTER_NAME
<i>ClusterNameList</i>	MQCFST	MQCA_CLUSTER_NAMELIST
<i>DefBind</i>	MQCFIN	
Note::		
1. This parameter applies to Change Queue only.		

Optional parameters (model queue)

Table 75. Required parameters (Change Channel)

Parameter	Data type	Parameter identifier
<i>QDesc</i>	MQCFST	MQCA_Q_DESC
<i>InhibitPut</i>	MQCFIN	MQIA_INHIBIT_PUT
<i>DefPriority</i>	MQCFIN	MQIA_DEF_PRIORITY
<i>DefPersistence</i>	MQCFIN	MQIA_DEF_PERSISTENCE
<i>InhibitGet</i>	MQCFIN	MQIA_INHIBIT_GET
<i>ProcessName</i>	MQCFST	MQCA_PROCESS_NAME
<i>MaxQDepth</i>	MQCFIN	MQIA_MAX_Q_DEPTH
<i>MaxMsgLength</i>	MQCFIN	MQIA_MAX_MSG_LENGTH
<i>BackoutThreshold</i>	MQCFIN	MQIA_BACKOUT_THRESHOLD
<i>BackoutRequeueName</i>	MQCFST	MQCA_BACKOUT_REQ_Q_NAME
<i>Shareability</i>	MQCFIN	MQIA_SHAREABILITY
<i>DefInputOpenOption</i>	MQCFIN	MQIA_DEF_INPUT_OPEN_OPTION
<i>HardenGetBackout</i>	MQCFIN	MQIA_HARDEN_GET_BACKOUT
<i>MsgDeliverySequence</i>	MQCFIN	MQIA_MSG_DELIVERY_SEQUENCE
<i>RetentionInterval</i>	MQCFIN	MQIA_RETENTION_INTERVAL
<i>DistLists</i>	MQCFIN	MQIA_DIST_LISTS
<i>Usage</i>	MQCFIN	MQIA_USAGE
<i>InitiationQName</i>	MQCFST	MQCA_INITIATION_Q_NAME
<i>TriggerControl</i>	MQCFIN	MQIA_TRIGGER_CONTROL
<i>TriggerType</i>	MQCFIN	MQIA_TRIGGER_TYPE
<i>TriggerMsgPriority</i>	MQCFIN	MQIA_TRIGGER_MSG_PRIORITY
<i>TriggerDepth</i>	MQCFIN	MQIA_TRIGGER_DEPTH

Queue commands

Table 75. Required parameters (Change Channel) (continued)

Parameter	Data type	Parameter identifier
<i>TriggerData</i>	MQCFST	MQIA_TRIGGER_DATA
<i>DefinitionType</i>	MQCFIN	MQIA_DEFINITION_TYPE
<i>QDepthHighLimit</i>	MQCFIN	MQIA_Q_DEPTH_HIGH_LIMIT
<i>QDepthLowLimit</i>	MQCFIN	MQIA_Q_DEPTH_LOW_LIMIT
<i>QDepthMaxEvent</i>	MQCFIN	MQIA_Q_DEPTH_MAX_EVENT
<i>QDepthHighEvent</i>	MQCFIN	MQIA_Q_DEPTH_HIGH_EVENT
<i>QDepthLowEvent</i>	MQCFIN	MQIA_Q_DEPTH_LOW_EVENT
<i>QServiceInterval</i>	MQCFIN	MQIA_Q_SERVICE_INTERVAL
<i>QServiceIntervalEvent</i>	MQCFIN	MQIA_Q_SERVICE_INTERVAL_EVENT

Optional parameters (remote queue)

Table 76. Required parameters (Change Channel)

Parameter	Data type	Parameter identifier
<i>QDesc</i>	MQCFST	MQCA_Q_DESC
<i>InhibitPut</i>	MQCFIN	MQIA_INHIBIT_PUT
<i>DefPriority</i>	MQCFIN	MQIA_DEF_PRIORITY
<i>DefPersistence</i>	MQCFIN	MQIA_DEF_PERSISTENCE
<i>Force (1)</i>	MQCFIN	MQIACF_FORCE
<i>RemoteQName</i>	MQCFST	MQCA_REMOTE_Q_NAME
<i>RemoteQMgrName</i>	MQCFST	MQCA_REMOTE_QMGR_NAME
<i>XmitQName</i>	MQCFST	MQCA_XMIT_Q_NAME
<i>Scope</i>	MQCFIN	MQIA_SCOPE
<i>ClusterName</i>	MQCFST	MQCA_CLUSTER_NAME
<i>ClusterNameList</i>	MQCFST	MQCA_CLUSTER_NAMELIST
<i>DefBind</i>	MQCFIN	
Note::		
1. This parameter applies to Change Queue only.		

Change Queue Manager

Purpose: The Change Queue Manager (MQCMD_CHANGE_Q_MGR) command changes the specified attributes of a queue manager.

Table 77. Inquire channel parameters

Parameter	Data type	Parameter identifier	
<i>Force</i>	MQCFIN	MQIACF_FORCE	Optional
<i>QMgrDesc</i>	MQCFST	MQCA_Q_MGR_DESC	Optional
<i>TriggerInterval</i>	MQCFIN	MQIA_TRIGGER_INTERVAL	Optional
<i>DeadLetterQName</i>	MQCFST	MQCA_DEAD_LETTER_Q_NAME	Optional
<i>MaxHandles</i>	MQCFIN	MQIA_MAX_HANDLES	Optional
<i>MaxUncommittedMsgs</i>	MQCFIN	MQIA_MAX_UNCOMMITTED_MSGS	Optional
<i>DefXmitQName</i>	MQCFST	MQCA_DEF_XMIT_Q_NAME	Optional
<i>AuthorityEvent</i>	MQCFIN	MQIA_AUTHORITY_EVENT	Optional
<i>InhibitEvent</i>	MQCFIN	MQIA_INHIBIT_EVENT	Optional
<i>LocalEvent</i>	MQCFIN	MQIA_LOCAL_EVENT	Optional
<i>RemoteEvent</i>	MQCFIN	MQIA_REMOTE_EVENT	Optional
<i>StartStopEvent</i>	MQCFIN	MQIA_START_STOP_EVENT	Optional
<i>PerformanceEvent</i>	MQCFIN	MQIA_PERFRORMANCE_EVENT	Optional
<i>MaxMsgLength</i>	MQCFIN	MQIA_MAX_MSG_LENGTH	Optional
<i>ChannelAutoDef</i>	MQCFIN	MQIA_CHANNEL_AUTO_DEF	Optional
<i>ChannelAutoDefEvent</i>	MQCFIN	MQIA_CHANNEL_AUTO_DEF_EVENT	Optional
<i>ChannelAutoDefExit</i>	MQCFST	MQIA_CHANNEL_AUTO_DEF_EXIT	Optional

Clear Queue

Purpose: The Clear Queue (MQCMD_CLEAR_Q) command deletes all of the messages from a local queue.

Table 78. Clear Queue parameters

Parameter	Data type	Parameter identifier	
<i>QName</i>	MQCFST	MQCA_Q_NAME	Required

Delete Channel

Purpose: The Delete Channel (MQCMD_DELETE_CHANNEL) command deletes the specified channel definition.

Delete Channel

Table 79. Delete Channel parameters

Parameter	Data type	Parameter identifier	
<i>ChannelName</i>	MQCFST	MQCACH_CHANNEL_NAME	Required
<i>ChannelTable</i>	MQCFIN	MQCACH_CHANNEL_NAME	Optional

Delete Namelist

Purpose: The Delete Namelist (MQCMD_DELETE_NAMELIST) command deletes an existing MQSeries namelist definition.

Platforms:

This PCF is supported if you are using AIX, HP-UX, OS/2, OS/400, Sun Solaris, or Windows NT.

Table 80. Delete Namelist parameters

Parameter	Data type	Parameter identifier	
<i>NamelistName</i>	MQCFST	MQCA_NAMELIST_NAME	Required

Delete Process

Purpose: The Delete Process (MQCMD_DELETE_PROCESS) command deletes an existing MQSeries process definition.

Platforms: This PCF is not supported if you are MQSeries for Windows Version 2.1.

Table 81. Delete Process parameters

Parameter	Data type	Parameter identifier	
<i>ProcessName</i>	MQCFST	MQCA_PROCESS_NAME	Required

Delete Queue

Purpose: The Delete Queue (MQCMD_DELETE_Q) command deletes an existing MQSeries queue.

Table 82. Delete Queue parameters

Parameter	Data type	Parameter identifier	
<i>QName</i>	MQCFST	MQCA_Q_NAME	Required
<i>QType</i>	MQCFIN	MQIA_Q_TYPE	Optional
<i>Purge (1)</i>	MQCFIN	MQIACF_PURGE	Optional
Note::			
1. This parameter applies to local queues only.			

Escape

Purpose: The Escape (MQCMD_ESCAPE) command conveys any MQSC command to a remote queue manager.

Platforms: This PCF is not supported if you are using MQSeries for Windows Version 2.1.

Table 83. Escape parameters

Parameter	Data type	Parameter identifier	
<i>EscapeType</i>	MQCFIN	MQIACF_ESCAPE_TYPE	Required
<i>EscapeText</i>	MQCFST	MQIACF_ESCAPE_TEXT	Required

Escape (Response)

Purpose: The response to the Escape command consists of the response header followed by two parameter structures, one containing the type and the other the text response.

Platforms: This response is not supported on 32-bit Windows.

Table 84. Response parameters (Escape)

Parameter	Data type	Returned
<i>EscapeType</i>	MQCFIN	Always
<i>EscapeText</i>	MQCFST	Always

Inquire Channel

Purpose: The Inquire Channel (MQCMD_INQUIRE_CHANNEL) command inquires about the attributes of MQSeries channel definitions.

Inquire Channel

Table 85. Inquire channel parameters

Parameter	Data type	Parameter identifier	
<i>ChannelName</i>	MQCFST	MQCACH_CHANNEL_NAME	Required
<i>ChannelType</i>	MQCFIN	MQIACH_CHANNEL_TYPE	Optional
<i>ChannelAttrs</i>	MQCFIL	MQIACF_CHANNEL_ATTRS	Optional

Inquire Channel (Response)

Purpose: The response to the Inquire Channel command consists of the response header followed by the *ChannelName* structure and the requested combination of attribute parameter structures.

Table 86. Response parameters (Inquire Channel)

Parameter	Data type	Returned
<i>ChannelName</i>	MQCFST	Always
<i>ChannelType</i>	MQCFIN	If requested
<i>TransportType</i>	MQCFIN	If requested
<i>ModeName</i>	MQCFST	If requested
<i>TpName</i>	MQCFST	If requested
<i>QMgrName</i>	MQCFST	If requested
<i>ConnectionName</i>	MQCFST	If requested
<i>XmitQName</i>	MQCFST	If requested
<i>MCAName</i>	MQCFST	If requested
<i>ChannelDesc</i>	MQCFST	If requested
<i>BatchSize</i>	MQCFIN	If requested
<i>DscInterval</i>	MQCFIN	If requested
<i>ShortRetryCount</i>	MQCFIN	If requested
<i>ShortRetryInterval</i>	MQCFIN	If requested
<i>LongRetryCount</i>	MQCFIN	If requested
<i>LongRetryInterval</i>	MQCFIN	If requested
<i>DataConversion</i>	MQCFIN	If requested
<i>SecurityExit</i>	MQCFST	If requested
<i>MsgExit</i>	MQCFST	If requested
<i>SendExit</i>	MQCFST	If requested
<i>ReceiveExit</i>	MQCFST	If requested
<i>PutAuthority</i>	MQCFIN	If requested

Table 86. Response parameters (Inquire Channel) (continued)

Parameter	Data type	Returned
<i>SeqNumberWrap</i>	MQCFIN	If requested
<i>MaxMsgLength</i>	MQCFIN	If requested
<i>SecurityUserData</i>	MQCFST	If requested
<i>MsgUserData</i>	MQCFST	If requested
<i>SendUserData</i>	MQCFST	If requested
<i>ReceiveUserData</i>	MQCFST	If requested
<i>MCAType</i>	MQCFIN	If requested
<i>MCAUserIdentifier</i>	MQCFIN	If requested
<i>UserIdentifier</i>	MQCFST	If requested
<i>Password</i>	MQCFST	If requested
<i>MsgRetryExit</i>	MQCFST	If requested
<i>MsgRetryUserData</i>	MQCFST	If requested
<i>MsgRetryCount</i>	MQCFIN	If requested
<i>MsgRetryInterval</i>	MQCFIN	If requested
<i>HeartbeatInterval</i>	MQCFIN	If requested
<i>NonPersistentMsgSpeed</i>	MQCFIN	If requested
<i>BatchInterval</i>	MQCFIN	If requested
<i>ClusterName</i>	MQCFST	If requested
<i>ClusterNameList</i>	MQCFST	If requested
<i>NetworkPriority</i>	MQCFIN	If requested

Inquire Channel Names

Inquire Channel Names

Purpose: The Inquire Channel (MQCMD_INQUIRE_CHANNEL_NAMES) command inquires a list of MQSeries channel names that match the generic name, and the optional channel type specified.

Table 87. Inquire Channel Names parameters

Parameter	Data type	Parameter identifier	
<i>ChannelName</i>	MQCFST	MQCACH_CHANNEL_NAME	Required
<i>ChannelType</i>	MQCFIN	MQCACH_CHANNEL_TYPE	Optional

Inquire Channel Names (Response)

The response to the Inquire Channel Names command consists of the response header followed by a single parameter structure giving zero or more names that match the specified channel name.

Table 88. Response parameters (Inquire Channel Names)

Parameter	Data type	Returned
<i>ChannelName</i>	MQCFST	Always

Inquire Channel Status

Purpose: The Inquire Channel Status (MQCMD_INQUIRE_CHANNEL_STATUS) command inquires a list of MQSeries channel names that match the generic name, and the optional channel type specified.

Table 89. Inquire Channel Status parameters

Parameter	Data type	Parameter identifier	
<i>ChannelName</i>	MQCFST	MQCACH_CHANNEL_NAME	Required
<i>XmitQName</i>	MQCFST	MQCACH_XMIT_Q_NAME	Optional
<i>ConnectionName</i>	MQCFST	MQCACH_CONNECTION_NAME	Optional
<i>ChannelInstanceType</i>	MQCFIN	MQIACH_CHANNEL_INSTANCE_TYPE	Optional
<i>ChannelInstanceAttrs</i>	MQCFIL	MQIACH_CHANNEL_INSTANCE_ATTRS	Optional

Inquire Channel Status (Response)

Purpose: The response to the Inquire Channel Status command consists of the response header followed by the:

- *ChannelName* structure
- *XmitQName* structure
- *ConnectionName* structure
- *ChannelInstanceType* structure
- *ChannelType* structure
- *ChannelStatus* structure

which are followed by the requested combination of the status attribute parameter structures.

Table 90. Response Parameters (Inquire Channel Status)

Parameter	Data type	Returned
<i>ChannelName</i>	MQCFST	Always
<i>XmitQName</i>	MQCFST	Always
<i>ConnectionName</i>	MQCFST	Always
<i>ChannelInstanceType</i>	MQCFIN	Always
<i>ChannelType</i>	MQCFIN	Always
<i>ChannelStatus</i>	MQCFIN	Always
<i>InDoubtStatus</i>	MQCFIN	If requested
<i>LastSequenceNumber</i>	MQCFIN	If requested
<i>LastLUWID</i>	MQCFST	If requested
<i>CurrentMsgs</i>	MQCFIN	If requested
<i>CurrentSequenceNumber</i>	MQCFIN	If requested
<i>CurrentLUWID</i>	MQCFST	If requested
<i>LastMsgTime</i>	MQCFST	If requested
<i>LastMsgTime</i>	MQCFST	If requested
<i>LastMsgDate</i>	MQCFST	If requested
<i>Msgs</i>	MQCFIN	If requested
<i>BytesSent</i>	MQCFIN	If requested
<i>BytesReceived</i>	MQCFIN	If requested
<i>Batches</i>	MQCFIN	If requested
<i>ChannelStartTime</i>	MQCFST	If requested
<i>ChannelStartDate</i>	MQCFST	If requested
<i>BuffersSent</i>	MQCFIN	If requested

Inquire Channel Status (Response)

Table 90. Response Parameters (Inquire Channel Status) (continued)

Parameter	Data type	Returned
<i>BuffersReceived</i>	MQCFIN	If requested
<i>LongRetriesLeft</i>	MQCFIN	If requested
<i>ShortRetriesLeft</i>	MQCFIN	If requested
<i>MCAJobName</i>	MQCFST	If requested
<i>MCAStatus</i>	MQCFIN	If requested
<i>StopRequested</i>	MQCFIN	If requested
<i>BatchSize</i>	MQCFIN	If requested
<i>HeartbeatInterval</i>	MQCFIN	If requested
<i>NonPersistentMsgSpeed</i>	MQCFIN	If requested
<i>ClusterName</i>	MQCFST	If requested
<i>ClusterNameList</i>	MQCFST	If requested
<i>NetworkPriority</i>	MQCFIN	If requested

Inquire Cluster Queue Manager

Purpose: The Inquire Cluster Queue Manager (MQCMD_INQUIRE_CLUSTER_Q_MGR) command inquires about the attributes of existing MQSeries queue managers in a cluster.

Platforms: This PCF is supported if you are using AIX, HP-UX, OS/2, OS/400, Sun Solaris, or Windows NT only.

Table 91. Inquire Cluster Queue Manager parameters

Parameter	Data type	Parameter identifier	
<i>ClusterQMgrName</i>	MQCFST	MQCA_CLUSTER_Q_MGR_NAME	Required
<i>Channel</i>	MQCFST	MQCACH_CHANNEL_NAME	Optional
<i>ClusterName</i>	MQCFST	MQCA_CLUSTER_NAME	Optional
<i>ClusterQMgrAttrs</i>	MQCFIL	MQIACF_CLUSTER_Q_MGR_ATTRS	Optional

Inquire Cluster Queue Manager (Response)

The response to the Inquire Cluster Queue Manager command consists of the response header followed by the *QMgrName* structure and the requested combination of attribute parameter structures.

Platforms: This response is supported if you are using AIX, HP-UX, OS/2, OS/400, Sun Solaris, or Windows NT only.

Inquire Cluster Queue Manager (Response)

Table 92. Response parameters (Inquire Cluster Queue Manager)

Parameter	Data type	Returned
<i>QMgrName</i>	MQCFST	Always
<i>ChannelName</i>	MQCFST	Always
<i>ClusterName</i>	MQCFST	Always
<i>TransportType</i>	MQCFIN	If requested
<i>ModeName</i>	MQCFST	If requested
<i>TpName</i>	MQCFST	If requested
<i>ConnectionName</i>	MQCFST	If requested
<i>MCAName</i>	MQCFST	If requested
<i>ChannelDesc</i>	MQCFST	If requested
<i>BatchSize</i>	MQCFIN	If requested
<i>DiscInterval</i>	MQCFIN	If requested
<i>ShortRetryCount</i>	MQCFIN	If requested
<i>ShortRetryInterval</i>	MQCFIN	If requested
<i>LongRetryCount</i>	MQCFIN	If requested
<i>LongRetryInterval</i>	MQCFIN	If requested
<i>DataConversion</i>	MQCFIN	If requested
<i>SecurityExit</i>	MQCFSL	If requested
<i>MsgExit</i>	MQCFSL	If requested
<i>SendExit</i>	MQCFSL	If requested
<i>ReceiveExit</i>	MQCFSL	If requested
<i>PutAuthority</i>	MQCFIN	If requested
<i>SeqNumberWrap</i>	MQCFIN	If requested
<i>MaxMsgLength</i>	MQCFIN	If requested
<i>SecurityUserData</i>	MQCFST	If requested
<i>MsgUserData</i>	MQCFSL	If requested
<i>SendUserData</i>	MQCFSL	If requested
<i>ReceiveUserData</i>	MQCFSL	If requested
<i>MCAType</i>	MQCFIN	If requested
<i>MCAUserIdentifier</i>	MQCFST	If requested
<i>UserIdentifier</i>	MQCFST	If requested
<i>Password</i>	MQCFST	If requested
<i>MsgRetryExit</i>	MQCFST	If requested

Inquire Cluster Queue Manager (Response)

Table 92. Response parameters (Inquire Cluster Queue Manager) (continued)

Parameter	Data type	Returned
<i>MsgRetryUserData</i>	MQCFST	If requested
<i>MsgRetryCount</i>	MQCFIN	If requested
<i>MsgRetryInterval</i>	MQCFIN	If requested
<i>HeartbeatInterval</i>	MQCFIN	If requested
<i>NonPersistentMsgSpeed</i>	MQCFIN	If requested
<i>BatchInterval</i>	MQCFIN	If requested
<i>AlterationDate</i>	MQCFST	If requested
<i>AlterationTime</i>	MQCFST	If requested
<i>ClusterInfo</i>	MQCFST	If requested
<i>ClusterName</i>	MQCFIN	If requested
<i>QMgrDefinitionType</i>	MQCFIN	If requested
<i>QMgrType</i>	MQCFIN	If requested
<i>QMgrIdentifier</i>	MQCFST	If requested
<i>ClusterDate</i>	MQCFST	If requested
<i>ClusterTime</i>	MQCFST	If requested
<i>ChannelStatus</i>	MQCFIN	If requested
<i>Suspend</i>	MQCFIN	If requested
<i>NetworkPriority</i>	MQCFIN	If requested

Inquire Namelist

Purpose: The Inquire Namelist (MQCMD_INQUIRE_NAMELIST) command inquires about the specified attributes of existing MQSeries namelists.

Platforms: This PCF is supported if you are using AIX, HP-UX, OS/2, OS/400, Sun Solaris, or Windows NT only.

Table 93. Inquire Namelist parameters

Parameter	Data type	Parameter identifier	
<i>NamelistName</i>	MQCFST	MQCA_NAMELIST_NAME	Required
<i>NamelistAttrs</i>	MQCFIL	MQIACF_NAMELIST_ATTRS	Optional

Inquire Namelist (Response)

Purpose: The response to the Inquire Namelist (MQCMD_INQUIRE_NAMELIST) command consists of the response header followed by the *NamelistName* structure and the requested combination of attribute parameter structures.

Platforms: This response is supported if you are using AIX, HP-UX, OS/2, OS/400, Sun Solaris, or Windows NT only.

Table 94. Response parameters (Inquire Namelist)

Parameter	Data type	Returned
<i>NamelistName</i>	MQCFST	Always
<i>NamelistDesc</i>	MQCFST	If requested
<i>Names</i>	MQCFSL	If requested
<i>AlterationDate</i>	MQCFST	If requested
<i>AlterationTime</i>	MQCFST	If requested

Inquire Namelist Names

Purpose: The Inquire Namelist Names (MQCMD_INQUIRE_NAMELIST_NAMES) command inquires for a list of namelist names that match the generic namelist name specified.

Platforms: This PCF is supported if you are using AIX, HP-UX, OS/2, OS/400, Sun Solaris, or Windows NT.

Table 95. Inquire Namelist Names parameters

Parameter	Data type	Parameter identifier	
<i>NamelistNames</i>	MQCFSL	MQCA_NAMELIST_NAMES	Required

Inquire Namelist Names (Response)

Purpose: The response to the Inquire Namelist Names (MQCMD_INQUIRE_NAMELIST_NAMES) command consists of the response header followed by a single parameter structure giving zero or more names that match the specified namelist name.

Platforms: This response is supported if you are using AIX, HP-UX, OS/2, OS/400, Sun Solaris, or Windows NT.

Inquire Namelist Names (Response)

Table 96. Response parameters (Inquire Namelist Names)

Parameter	Data type	Returned
<i>NamelistNames</i>	MQCFSL	Always

Inquire Process

Purpose: The Inquire Process (MQCMD_INQUIRE_PROCESS) command inquires about the attributes of existing MQSeries process definitions.

Platforms: This PCF is not supported if you are using MQSeries for Windows version 2.1.

Table 97. Inquire Process parameters

Parameter	Data type	Parameter identifier	
<i>ProcessName</i>	MQCFST	MQCA_PROCESS_NAME	Required
<i>ProcessAttrs</i>	MQCFIL	MQIACF_PROCESS_ATTRS	Optional

Inquire Process (Response)

Purpose: The response to the Inquire Process (MQCMD_INQUIRE_PROCESS) command the specified attributes of an existing MQSeries process definition.

Platforms: This response is not supported if you are using 32-bit Windows.

Table 98. Response parameters (Inquire Process)

Parameter	Data type	Returned
<i>ProcessName</i>	MQCFST	Always
<i>ProcessDesc</i>	MQCFST	If requested
<i>ApplType</i>	MQCFIN	If requested
<i>ApplId</i>	MQCFST	If requested
<i>EnvData</i>	MQCFST	If requested
<i>UserData</i>	MQCFST	If requested

Inquire Process Names

Purpose: The Inquire Process Names (MQCMD_INQUIRE_PROCESS_NAMES) command inquires for a list of process names that match the generic process name specified.

Platforms: This PCF is not supported if you are using MQSeries for Windows version 2.1.

Table 99. Inquire Process Names parameters

Parameter	Data type	Parameter identifier	
<i>ProcessName</i>	MQCFST	MQCA_PROCESS_NAME	Required

Inquire Process Names (Response)

Purpose: The response to the Inquire Process Names (MQCMD_INQUIRE_PROCESS_NAMES) command consists of the response header followed by a single parameter structure giving zero or more names that match the specified process name.

Platforms: This PCF is not supported on 32-bit Windows.

Table 100. Response parameters (Inquire Process Names)

Parameter	Data type	Returned
<i>ProcessName</i>	MQCFST	Always

Inquire Queue

Purpose: The Inquire Queue (MQCMD_INQUIRE_Q) command inquires about the the attributes of existing MQSeries queues.

Table 101. Inquire Queue parameters

Parameter	Data type	Parameter identifier	
<i>QName</i>	MQCFST	MQCA_Q_NAME	Required
<i>QType</i>	MQCFIN	MQIA_Q_TYPE	Optional
<i>QAttrs</i>	MQCFIL	MQIACF_Q_ATTRS	Optional

Inquire Queue (Response)

The response to the Inquire Queue command consists of the response header followed by the *QName* structure and the requested combination of attribute parameter structures.

Table 102. Response parameters (Inquire Queue)

Parameter	Data type	Returned
<i>QName</i>	MQCFST	Always
<i>QType</i>	MQCFIN	If requested

Inquire Queue (Response)

Table 102. Response parameters (Inquire Queue) (continued)

Parameter	Data type	Returned
<i>QDesc</i>	MQCFST	If requested
<i>InhibitPut</i>	MQCFIN	If requested
<i>DefPriority</i>	MQCFIN	If requested
<i>DefPersistence</i>	MQCFIN	If requested
<i>InhibitGet</i>	MQCFIN	If requested
<i>ProcessName</i>	MQCFST	If requested
<i>MaxQDepth</i>	MQCFIN	If requested
<i>MaxMsgLength</i>	MQCFIN	If requested
<i>BackoutThreshold</i>	MQCFIN	If requested
<i>BackoutRequeueName</i>	MQCFST	If requested
<i>Shareability</i>	MQCFIN	If requested
<i>DefInputOpenOption</i>	MQCFIN	If requested
<i>HardenGetBackout</i>	MQCFIN	If requested
<i>MsgDeliverySequence</i>	MQCFIN	If requested
<i>RetentionInterval</i>	MQCFIN	If requested
<i>DefinitionType</i>	MQCFIN	If requested
<i>DistLists</i>	MQCFIN	If requested
<i>Usage</i>	MQCFIN	If requested
<i>OpenInputCount</i>	MQCFIN	If requested
<i>OpenOutputCount</i>	MQCFIN	If requested
<i>CurrentQDepth</i>	MQCFIN	If requested
<i>CreationDate</i>	MQCFST	If requested
<i>CreationTime</i>	MQCFST	If requested
<i>InitiationQName</i>	MQCFST	If requested
<i>TriggerControl</i>	MQCFIN	If requested
<i>TriggerType</i>	MQCFIN	If requested
<i>TriggerMsgPriority</i>	MQCFIN	If requested
<i>TriggerDepth</i>	MQCFIN	If requested
<i>TriggerData</i>	MQCFST	If requested
<i>BaseQName</i>	MQCFST	If requested
<i>RemoteQName</i>	MQCFST	If requested
<i>RemoteQMgrName</i>	MQCFST	If requested

Table 102. Response parameters (Inquire Queue) (continued)

Parameter	Data type	Returned
<i>XmitQName</i>	MQCFST	If requested
<i>Scope</i>	MQCFIN	If requested
<i>QDepthHighLimit</i>	MQCFIN	If requested
<i>QDepthLowLimit</i>	MQCFIN	If requested
<i>QDepthMaxEvent</i>	MQCFIN	If requested
<i>QDepthHighEvent</i>	MQCFIN	If requested
<i>QDepthLowEvent</i>	MQCFIN	If requested
<i>QServiceInterval</i>	MQCFIN	If requested
<i>QServiceIntervalEvent</i>	MQCFIN	If requested
<i>ClusterName</i>	MQCFST	If requested
<i>ClusterNameList</i>	MQCFST	If requested
<i>DefBind</i>	MQCFIN	If requested

Inquire Queue Manager

Purpose: The Inquire Queue Manager (MQCMD_INQUIRE_Q_MGR) command inquires about the specified attributes of an existing MQSeries queue manager.

Table 103. Inquire Queue Manager parameters

Parameter	Data type	Parameter identifier	
<i>QMgrAttrs</i>	MQCFIL	MQIACF_Q_MGR_ATTRS	Optional

Inquire Queue Manager (Response)

The response to the Inquire Queue Manager (MQCMD_INQUIRE_Q_MGR) command consists of the response header followed by the *QmgrName* structure and the requested combination of attribute parameter structures.

Table 104. Response parameters (Inquire Queue Manager)

Parameter	Data type	Returned
<i>QMgrName</i>	MQCFST	Always
<i>QMgrDesc</i>	MQCFST	If requested
<i>Platform</i>	MQCFIN	If requested
<i>CommandLevel</i>	MQCFIN	If requested
<i>TriggerInterval</i>	MQCFIN	If requested

Inquire Queue Manager (Response)

Table 104. Response parameters (Inquire Queue Manager) (continued)

Parameter	Data type	Returned
<i>DeadLetterQName</i>	MQCFST	If requested
<i>MaxPriority</i>	MQCFIN	If requested
<i>CommandInputQName</i>	MQCFST	If requested
<i>DefXmitQName</i>	MQCFST	If requested
<i>CodedCharSetId</i>	MQCFST	If requested
<i>MaxHandles</i>	MQCFIN	If requested
<i>MaxUncommittedMsgs</i>	MQCFIN	If requested
<i>MaxMsgLength</i>	MQCFIN	If requested
<i>DistLists</i>	MQCFIN	If requested
<i>SyncPoint</i>	MQCFIN	If requested
<i>AuthorityEvent</i>	MQCFIN	MQIA_AUTHORITY_EVENT
<i>InhibitEvent</i>	MQCFIN	MQIA_INHIBIT_EVENT
<i>LocalEvent</i>	MQCFIN	MQIA_LOCAL_EVENT
<i>RemoteEvent</i>	MQCFIN	MQIA_REMOTE_EVENT
<i>StartStopEvent</i>	MQCFIN	MQIA_START_STOP_EVENT
<i>PerformanceEvent</i>	MQCFIN	MQIA_PERFORMANCE_EVENT
<i>ChannelAutoDef</i>	MQCFIN	MQIA_CHANNEL_AUTO_DEF
<i>ChannelAutoDefEvent</i>	MQCFIN	MQIA_CHANNEL_AUTO_DEF_EVENT
<i>ChannelAutoDefExit</i>	MQCFST	MQIA_CHANNEL_AUTO_DEF_EXIT

Inquire Queue Names

Purpose: The Inquire Queue Names (MQCMD_INQUIRE_Q_NAMES) command inquires a list of queue names that match the generic queue name, and the optional queue type specified.

Table 105. Inquire Queue Names parameters

Parameter	Data type	Parameter identifier	
<i>QName</i>	MQCFST	MQCA_Q_NAME	Required
<i>QType</i>	MQCFIN	MQIA_Q_TYPE	Optional

Inquire Queue Names (Response)

Purpose: The response to the Inquire Queue Names (MQCMD_INQUIRE_Q_NAMES) command consists of the response header followed by a single parameter structure giving zero or more names that match the specified queue name.

Table 106. Response parameters (Inquire Queue Names)

Parameter	Data type	Returned
<i>QNames</i>	MQCFSL	Always

Ping Channel

Purpose: The Ping Channel (MQCMD_PING_CHANNEL) command tests a channel by sending data as a special message to the remote message queue manager and checking that the data is returned.

Platforms: This PCF is not supported if you are using MQSeries for Windows version 2.1.

Table 107. Ping Channel parameters

Parameter	Data type	Parameter identifier	
<i>ChannelName</i>	MQCFST	MQCACH_CHANNEL_NAME	Required
<i>DataCount</i>	MQCFIN	MQIACH_DATA_COUNT	Optional

Ping Queue Manager

Purpose: The Ping Queue Manager (MQCMD_PING_Q_MGR) command tests whether the queue manager and its command server is responsive to commands.

This command has no parameters.

Refresh Cluster

Purpose: The Refresh Cluster (MQCMD_REFRESH_CLUSTER) command discards all locally held cluster information, including any auto-defined channels that are not in doubt, and forces the repository to be rebuilt.

Platforms: This PCF is supported if you are using AIX, HP-UX, OS/2, OS/400, Sun Solaris, or Windows NT.

Refresh Cluster

Table 108. Reset Cluster parameters

Parameter	Data type	Parameter identifier	
<i>ClusterName</i>	MQCFST	MQCA_CLUSTER_NAME	Required

Reset Channel

Purpose: The Reset Channel (MQCMD_RESET_CHANNEL) command resets the message sequence number for an MQSeries channel with, optionally, a specified sequence number to be used the next time that the channel is started.

Table 109. Reset Channel parameters

Parameter	Data type	Parameter identifier	
<i>ChannelName</i>	MQCFST	MQCACH_CHANNEL_NAME	Required
<i>MsgSeqNumber</i>	MQCFIN	MQIACH_MSG_SEQUENCE_NUMBER	Optional

Reset Cluster

Purpose: The Reset Cluster (MQCMD_RESET_CLUSTER) command forces a queue manager to leave a cluster.

Platforms: This PCF is supported if you are using AIX, HP-UX, OS/2, OS/400, Sun Solaris, or Windows NT.

Table 110. Reset Cluster parameters

Parameter	Data type	Parameter identifier	
<i>ClusterName</i>	MQCFST	MQCA_CLUSTER_NAME	Required
<i>QmgrName</i>	MQCFST	MQCA_Q_MGR_NAME	Required
<i>Action</i>	MQCFIN	MQIACF_ACTION	Required

Reset Queue Statistics

Purpose: The Reset Queue Statistics (MQCMD_RESET_Q_STATS) command reports the performance data for a queue and then resets the performance data.

Table 111. Reset Queue Statistics parameters

Parameter	Data type	Parameter identifier	
<i>QName</i>	MQCFST	MQCA_Q_NAME	Required

Reset Queue Statistics (Response)

Purpose: The response to the Reset Queue Statistics (MQCMD_RESET_Q_STATS) command consists of the response header followed by the *QName* structure and the attribute parameter structures listed.

Table 112. Response parameters (Reset Queue Statistics)

Parameter	Data type	Returned
<i>QName</i>	MQCFST	Always
<i>TimeSinceReset</i>	MQCFIN	Always
<i>HighQDepth</i>	MQCFIN	Always
<i>MsgEnqCount</i>	MQCFIN	Always
<i>MsgDeqCount</i>	MQCFIN	Always

Resume Queue Manager Cluster

Purpose: The Resume Queue Manager Cluster (MQCMD_RESUME_Q_MGR_CLUSTER) command informs other queue managers in a cluster that the local queue manager is again available for processing, and can be sent messages.

Platforms: This PCF is supported if you are using AIX, HP-UX, OS/2, OS/400, Sun Solaris, or Windows NT.

Table 113. Resume Queue Manager Cluster parameters

Parameter	Data type	Parameter identifier	
<i>ClusterName</i>	MQCFST	MQCA_CLUSTER_NAME	Required
<i>ClusterNameList</i>	MQCFST	MQCA_CLUSTER_NAMELIST	Required

Start Channel

Purpose: The Start Channel (MQCMD_START_CHANNEL) command starts an MQSeries channel.

Table 114. Start Channel parameters

Parameter	Data type	Parameter identifier	
<i>ChannelName</i>	MQCFST	MQCACH_CHANNEL_NAME	Required

Start Channel Initiator

Start Channel Initiator

Purpose: The Start Channel Initiator (MQCMD_START_CHANNEL_INIT) command starts an MQSeries channel initiator.

Platforms: This PCF is not supported if you are using MQSeries for Windows Version 2.1.

Table 115. Start Channel Initiator

Parameter	Data type	Parameter identifier	
<i>InitiationQName</i>	MQCFST	MQCACH_INITIATION_Q_NAME	Required

Start Channel Listener

Purpose: The Start Channel Listener (MQCMD_START_CHANNEL_LISTENER) command starts an MQSeries TCP/IP listener.

Platforms: This PCF is supported if you are using MQSeries for AS/400 V5.1, MQSeries for OS/2 Warp V5.1, or MQSeries for Windows NT V5.1.

This command has no parameters.

Stop Channel

Purpose: The Stop Channel (MQCMD_STOP_CHANNEL) command stops an MQSeries channel.

Table 116. Stop Channel parameters

Parameter	Data type	Parameter identifier	
<i>ChannelName</i>	MQCFST	MQCACH_CHANNEL_NAME	Required
<i>Quiesce</i>	MQCFIN	MQIACF QUIESCE	Optional

Suspend Queue Manager Cluster

Purpose: The Suspend Queue Manager Cluster (MQCMD_SUSPEND_Q_MGR_CLUSTER) command informs other queue managers in a cluster that the local queue manager is not available for processing, and cannot be sent messages.

Platforms: This PCF is supported if you are using AIX, HP-UX, OS/2, OS/400, Sun Solaris, or Windows NT.

Suspend Queue Manager Cluster

Table 117. Suspend Queue Manager Cluster parameters

Parameter	Data type	Parameter identifier	
<i>ClusterName</i>	MQCFST	MQCA_CLUSTER_NAME	Required
<i>ClusterNameList</i>	MQCFST	MQCA_CLUSTER_NAMELIST	Required
<i>Quiesce</i>	MQCFIN	MQIACF_QUIESCE	Optional

Suspend Queue Manager Cluster

Chapter 5. MQAI calls

Full details of these calls can be found in the *MQSeries Administration Interface Programming Guide and Reference* manual.

mqAddInquiry

Purpose: The mqAddInquiry call adds a selector to an administration bag.

Table 118. mqAddInquiry call

Parameter	Data type	Usage
<i>Bag</i>	MQHBAG	input
<i>Selector</i>	MQLONG	input
<i>CompCode</i>	MQLONG	output
<i>Reason</i>	MQLONG	output

mqAddInteger

Purpose: The mqAddInteger call adds an integer item identified by a user selector to the end of a specified bag.

Table 119. mqAddInteger call

Parameter	Data type	Usage
<i>Bag</i>	MQHBAG	input
<i>Selector</i>	MQLONG	input
<i>ItemValue</i>	MQLONG	input
<i>CompCode</i>	MQLONG	output
<i>Reason</i>	MQLONG	output

mqAddString

mqAddString

Purpose: The mqAddString call adds a character data item identified by a user selector to the end of a specified bag.

Table 120. mqAddString call

Parameter	Data type	Usage
<i>Bag</i>	MQHBAG	input
<i>Selector</i>	MQLONG	input
<i>BufferLength</i>	MQLONG	input
<i>Buffer</i>	MQCHAR x BufferLength	output
<i>CompCode</i>	MQLONG	output
<i>Reason</i>	MQLONG	output

mqBagToBuffer

Purpose: The mqBagToBuffer call converts the bag into a PCF message in the supplied buffer.

Table 121. mqBagToBuffer call

Parameter	Data type	Usage
<i>OptionsBag</i>	MQHBAG	input
<i>DataBag</i>	MQHBAG	input
<i>BufferLength</i>	MQLONG	input
<i>Buffer</i>	MQBYTE x BufferLength	output
<i>DataLength</i>	MQLONG	output
<i>CompCode</i>	MQLONG	output
<i>Reason</i>	MQLONG	output

mqBufferToBag

Purpose: The mqBufferToBag call converts the specified buffer into bag form.

Table 122. mqBufferToBag call

Parameter	Data type	Usage
<i>OptionsBag</i>	MQHBAG	input
<i>BufferLength</i>	MQLONG	input
<i>Buffer</i>	MQBYTE × BufferLength	output
<i>DataBag</i>	MQHBAG	input/output
<i>CompCode</i>	MQLONG	output
<i>Reason</i>	MQLONG	output

mqClearBag

Purpose: The mqClearBag call deletes all user items from the bag and resets system items to their initial values.

Table 123. mqClearBag call

Parameter	Data type	Usage
<i>Bag</i>	MQHBAG	input
<i>CompCode</i>	MQLONG	output
<i>Reason</i>	MQLONG	output

mqCountItems

Purpose: The mqCountItems call returns the number of occurrences of user items, system items, or both, that are stored in a bag with the same specific selector.

Table 124. mqCountItems call

Parameter	Data type	Usage
<i>Bag</i>	MQHBAG	input
<i>Selector</i>	MQLONG	input
<i>ItemCount</i>	MQLONG	output
<i>CompCode</i>	MQLONG	output
<i>Reason</i>	MQLONG	output

mqCreateBag

mqCreateBag

Purpose: The mqCreateBag call creates a new bag.

Table 125. mqCreateBag call

Parameter	Data type	Usage
<i>Options</i>	MQHBAG	input
<i>Bag</i>	MQHBAG	output
<i>CompCode</i>	MQLONG	output
<i>Reason</i>	MQLONG	output

mqDeleteBag

Purpose: The mqDeleteBag call deletes a bag.

Table 126. mqDeleteBag call

Parameter	Data type	Usage
<i>Bag</i>	MQHBAG	input/output
<i>CompCode</i>	MQLONG	output
<i>Reason</i>	MQLONG	output

mqDeleteItem

Purpose: The mqDeleteItem call removes one or more user items from a bag.

Table 127. mqDeleteItem call

Parameter	Data type	Usage
<i>Bag</i>	MQHBAG	input
<i>Selector</i>	MQLONG	input
<i>ItemIndex</i>	MQLONG	input
<i>CompCode</i>	MQLONG	output
<i>Reason</i>	MQLONG	output

mqExecute

Purpose: The mqExecute call sends an administration command message and waits for the reply call (if expected).

Table 128. mqExecute call

Parameter	Data type	Usage
<i>Hconn</i>	MQHCONN	input
<i>Command</i>	MQLONG	input
<i>OptionsBag</i>	MQHBAG	input
<i>AdminBag</i>	MQHBAG	input
<i>ResponseBag</i>	MQHBAG	input
<i>AdminQ</i>	MQHOBJ	input
<i>ResponseQ</i>	MQHOBJ	output
<i>CompCode</i>	MQLONG	output
<i>Reason</i>	MQLONG	output

mqGetBag

Purpose: The mqGetBag call removes a message from the specified queue and converts the message data into a data bag.

Table 129. mqGetBag call

Parameter	Data type	Usage
<i>Hconn</i>	MQHCONN	input
<i>Hobj</i>	MQHOBJ	input
<i>MsgDesc</i>	MQMD	input/output
<i>GetMsgOpts</i>	MQGMO	input/output
<i>Bag</i>	MQHBAG	input/output
<i>CompCode</i>	MQLONG	output
<i>Reason</i>	MQLONG	output

mqlInquireBag

mqlInquireBag

Purpose: The mqlInquireBag call inquires the value of a bag handle that is present in the bag. The data item can be a user item or system item.

Table 130. mqlInquireBag call

Parameter	Data type	Usage
<i>Bag</i>	MQHBAG	input
<i>Selector</i>	MQLONG	input
<i>ItemIndex</i>	MQLONG	input
<i>ItemValue</i>	MQHBAG	output
<i>CompCode</i>	MQLONG	output
<i>Reason</i>	MQLONG	output

mqlInquireInteger

Purpose: The mqlInquireInteger call requests the value of an integer data item that is present in the bag. The data item can be a user item or system item.

Table 131. mqlInquireInteger call

Parameter	Data type	Usage
<i>Bag</i>	MQHBAG	input
<i>Selector</i>	MQLONG	input
<i>ItemIndex</i>	MQLONG	input
<i>ItemValue</i>	MQLONG	output
<i>CompCode</i>	MQLONG	output
<i>Reason</i>	MQLONG	output

mqInquireItemInfo

Purpose: The mqInquireItemInfo call returns information about a specified item in a bag. The data item can be a user item or system item.

Table 132. mqInquireItemInfo call

Parameter	Data type	Usage
<i>Bag</i>	MQHBAG	input
<i>Selector</i>	MQLONG	input
<i>ItemIndex</i>	MQLONG	input
<i>ItemType</i>	MQLONG	output
<i>OutSelector</i>	MQLONG	output
<i>CompCode</i>	MQLONG	output
<i>Reason</i>	MQLONG	output

mqInquireString

Purpose: The mqInquireString call requests the value of a character data item that is present in the bag. The data item can be a user item or system item.

Table 133. mqInquireString call

Parameter	Data type	Usage
<i>Bag</i>	MQHBAG	input
<i>Selector</i>	MQLONG	input
<i>ItemIndex</i>	MQLONG	input
<i>BufferLength</i>	MQLONG	input
<i>Buffer</i>	MQCHAR x BufferLength	output
<i>StringLength</i>	MQLONG	output
<i>CodedCharSetId</i>	MQLONG	output
<i>OutSelector</i>	MQLONG	output
<i>CompCode</i>	MQLONG	output
<i>Reason</i>	MQLONG	output

mqPad

mqPad

Purpose: The mqPad call pads a null-terminated string with blanks.

Table 134. mqPad call

Parameter	Data type	Usage
<i>String</i>	PMQCHAR	input
<i>BufferLength</i>	MQLONG	input
<i>Buffer</i>	MQCHAR x BufferLength	output
<i>CompCode</i>	MQLONG	output
<i>Reason</i>	MQLONG	output

mqPutBag

Purpose: The mqPutBag call converts the contents of the specified bag into a PCF message and sends the message to the specified queue. The contents of the bag are unchanged after the call.

Table 135. mqPutBag call

Parameter	Data type	Usage
<i>Hconn</i>	MQHCONN	input
<i>Hobj</i>	MQHOBJ	input
<i>MsgDesc</i>	MQMD	input/output
<i>PutMsgOPts</i>	MQPMO	input/output
<i>Bag</i>	MQHBAG	input
<i>CompCode</i>	MQLONG	output
<i>Reason</i>	MQLONG	output

mqSetInteger

Purpose: The mqSetInteger call either modifies an integer item that is already present in the bag or deletes existing occurrences of the specified selector and adds a new occurrence at the end of the bag.

Table 136. mqSetInteger call

Parameter	Data type	Usage
<i>Bag</i>	MQHBAG	input
<i>Selector</i>	MQLONG	input
<i>ItemIndex</i>	MQLONG	input
<i>ItemValue</i>	MQLONG	input
<i>CompCode</i>	MQLONG	output
<i>Reason</i>	MQLONG	output

mqSetString

Purpose: The mqSetString call either modifies a character data item that is already present in the bag or deletes existing occurrences of the specified selector and adds a new occurrence at the end of the bag.

Table 137. mqSetString call

Parameter	Data type	Usage
<i>Bag</i>	MQHBAG	input
<i>Selector</i>	MQLONG	input
<i>ItemIndex</i>	MQLONG	input
<i>BufferLength</i>	MQLONG	input
<i>Buffer</i>	MQCHAR x BufferLength	input
<i>CompCode</i>	MQLONG	output
<i>Reason</i>	MQLONG	output

mqTrim

mqTrim

Purpose: The mqTrim call trims the blanks from a blank-padded string, then ends it with a null.

Table 138. mqTrim call

Parameter	Data type	Usage
<i>BufferLength</i>	MQLONG	input
<i>Buffer</i>	MQCHAR x BufferLength	input
<i>String</i>	MQCHAR x BufferLength+1	input
<i>CompCode</i>	MQLONG	output
<i>Reason</i>	MQLONG	output

mqTruncateBag

Purpose: The mqTruncateBag call reduces the number of user items in a user bag to the specified value.

Table 139. mqTruncate call

Parameter	Data type	Usage
<i>Bag</i>	MQHBAG	input
<i>ItemCount</i>	MQLONG	input
<i>CompCode</i>	MQLONG	output
<i>Reason</i>	MQLONG	output

Chapter 6. Installable services interface

Full details of these calls can be found in the *MQSeries Programmable System Management* manual.

MQZ_CHECK_AUTHORITY and MQZ_CHECK_AUTHORITY_2

Purpose: The MQZ_CHECK_AUTHORITY function is provided by an authorization service component, and is invoked by the queue manager to check whether an entity has authority to perform a particular action, or actions, on a specified object.

The MQZ_CHECK_AUTHORITY_2 function is provided by an MQZAS_VERSION_2 authorization service component, and is invoked by the queue manager to check whether an entity has authority to perform a particular action, or actions, on a specified object.

The function identifier, for MQZEP, is MQZID_CHECK_AUTHORITY.

Table 140. MQZID_CHECK_AUTHORITY function

Parameter	Data type	Usage
<i>QmgrName</i>	MQCHAR48	input
<i>EntityName</i> (1)	MQCHAR12	input
<i>EntityType</i>	MQLONG	input
<i>ObjectName</i>	MQCHAR48	input
<i>ObjectType</i>	MQLONG	input
<i>Authority</i>	MQLONG	input
<i>ComponentData</i>	MQBYTE x ComponentDataLength	input/output
<i>Continuation</i>	MQLONG	output
<i>CompCode</i>	MQLONG	output
<i>Reason</i>	MQLONG	output

Note:

1. This parameter is replaced by the *EntityData* parameter in the MQZ_CHECK_AUTHORITY_2 function, which uses the MQZED data type structure.

MQZ_COPY_ALL_AUTHORITY

MQZ_COPY_ALL_AUTHORITY

Purpose: This function is provided by an authorization service component, and is invoked by the queue manager to copy all of the authorizations that are currently in force for a reference object to another object.

The function identifier, for MQZEP, is MQZID_COPY_ALL_AUTHORITY.

Table 141. MQZID_COPY_ALL_AUTHORITY function

Parameter	Data type	Usage
<i>QmgrName</i>	MQCHAR48	input
<i>RefObjectName</i>	MQCHAR12	input
<i>ObjectName</i>	MQCHAR48	input
<i>ObjectType</i>	MQLONG	input
<i>ComponentData</i>	MQBYTE x ComponentDataLength	input/output
<i>Continuation</i>	MQLONG	output
<i>CompCode</i>	MQLONG	output
<i>Reason</i>	MQLONG	output

MQZ_DELETE_AUTHORITY

Purpose: This function is provided by an authorization service component, and is invoked by the queue manager to delete all of the authorizations associated with the specified object.

The function identifier, for MQZEP, is MQZID_DELETE_AUTHORITY.

Table 142. MQZID_DELETE_AUTHORITY function

Parameter	Data type	Usage
<i>QmgrName</i>	MQCHAR48	input
<i>ObjectName</i>	MQCHAR48	input
<i>ObjectType</i>	MQLONG	input
<i>ComponentData</i>	MQBYTE x ComponentDataLength	input/output
<i>Continuation</i>	MQLONG	output
<i>CompCode</i>	MQLONG	output
<i>Reason</i>	MQLONG	output

MQZ_DELETE_NAME

Purpose: This function is provided by a name service component, and is invoked by the queue manager to delete an entry for the specified queue.

The function identifier, for MQZEP, is MQZID_DELETE_NAME

Table 143. MQZID_DELETE_NAME

Parameter	Data type	Usage
<i>QmgrName</i>	MQCHAR48	input
<i>QName</i>	MQCHAR48	input
<i>ComponentData</i>	MQBYTE x ComponentDataLength	input/output
<i>Continuation</i>	MQLONG	output
<i>CompCode</i>	MQLONG	output
<i>Reason</i>	MQLONG	output

MQZ_FIND_USERID

Purpose: This function is provided by a user ID service component, and is invoked by the queue manager to find the user ID, and optionally the password, to be associated with an application, when the application issues an MQCONN call.

The function identifier, for MQZEP, is MQZID_FIND_USERID.

Platforms: The *user identifier service* is available on MQSeries for OS/2 Warp only.

Table 144. MQZID_FIND_USERID function

Parameter	Data type	Usage
<i>QmgrName</i>	MQCHAR48	input
<i>Userid</i>	MQCHAR12	output
<i>Password</i>	MQCHAR12	output
<i>ComponentData</i>	MQBYTE x ComponentDataLength	input/output
<i>Continuation</i>	MQLONG	output
<i>CompCode</i>	MQLONG	output
<i>Reason</i>	MQLONG	output

MQZ_GET_AUTHORITY

MQZ_GET_AUTHORITY and MQZ_GET_AUTHORITY_2

Purpose: The MQZ_GET_AUTHORITY function is provided by an authorization service component, and is invoked by the queue manager to retrieve the authority that an entity has to access the specified object.

The MQZ_GET_AUTHORITY_2 function is provided by an MQZAS_VERSION_2 authorization service component, and is invoked by the queue manager to retrieve the authority that an entity has to access the specified object.

The function identifier, for MQZEP, is MQZID_GET_AUTHORITY.

Table 145. MQZID_GET_AUTHORITY function

Parameter	Data type	Usage
<i>QmgrName</i>	MQCHAR48	input
<i>EntityName</i> (1)	MQCHAR12	input
<i>EntityType</i>	MQLONG	input
<i>ObjectName</i>	MQCHAR48	input
<i>ObjectType</i>	MQLONG	input
<i>Authority</i>	MQLONG	input
<i>ComponentData</i>	MQBYTE x ComponentDataLength	input/output
<i>Continuation</i>	MQLONG	output
<i>CompCode</i>	MQLONG	output
<i>Reason</i>	MQLONG	output
Note:		
1. This parameter is replaced by the <i>EntityData</i> parameter in the MQZ_GET_AUTHORITY_2 function, which uses the MQZED data type structure.		

MQZ_GET_EXPLICIT_AUTHORITY and MQZ_GET_EXPLICIT_AUTHORITY_2

Purpose: The MQZ_GET_EXPLICIT_AUTHORITY function is provided by an authorization service component, and is invoked by the queue manager to retrieve the authority that a named group has to access a specified object (but without the additional authority of the **nobody** group) or the authority that the primary group of the named principal has to access a specified object.

The MQZ_GET_EXPLICIT_AUTHORITY_2 function is provided by an MQZAS_VERSION_2 authorization service component, and is invoked by the queue manager to retrieve the authority that a named group has to access a specified object or the authority that the named group has to access a specified object.

MQZ_GET_EXPLICIT_AUTHORITY

The function identifier, for MQZEP, is MQZID_GET_EXPLICIT_AUTHORITY.

Table 146. MQZID_GET_EXPLICIT_AUTHORITY function

Parameter	Data type	Usage
<i>QmgrName</i>	MQCHAR48	input
<i>EntityName</i> (1)	MQCHAR12	input
<i>EntityType</i>	MQLONG	input
<i>ObjectName</i>	MQCHAR48	input
<i>ObjectType</i>	MQLONG	input
<i>Authority</i>	MQLONG	input
<i>AuthorityMask</i> (2)	MQLONG	input
<i>ComponentData</i>	MQBYTE x ComponentDataLength	input/output
<i>Continuation</i>	MQLONG	output
<i>CompCode</i>	MQLONG	output
<i>Reason</i>	MQLONG	output

Notes:

1. This parameter is replaced by the *EntityData* parameter in the MQZ_GET_EXPLICIT_AUTHORITY_2 function, which uses the MQZED data type structure.
2. This parameter applies only to the MQZ_GET_EXPLICIT_AUTHORITY_2 function.

MQZ_INIT_AUTHORITY

MQZ_INIT_AUTHORITY

Purpose: This function is provided by an authorization service component, and is invoked by the queue manager during configuration of the component. It is expected to call MQZEP in order to provide information to the queue manager.

The function identifier, for MQZEP, is MQZID_INIT_AUTHORITY.

Table 147. MQZID_INIT_AUTHORITY function

Parameter	Data type	Usage
<i>Hconfig</i>	MQHCONFIG	input
<i>Options</i>	MQLONG	input
<i>QmgrName</i>	MQCHAR48	input
<i>ComponentDataLength</i>	MQLONG	input
<i>ComponentData</i>	MQBYTE x ComponentDataLength	input/output
<i>Version</i>	MQLONG	input/output
<i>CompCode</i>	MQLONG	output
<i>Reason</i>	MQLONG	output

MQZ_INIT_NAME

Purpose: This function is provided by a name service component, and is invoked by the queue manager during configuration of the component. It is expected to call MQZEP in order to provide information to the queue manager.

The function identifier, for MQZEP, is MQZID_INIT_NAME.

Table 148. MQZID_INIT_NAME function

Parameter	Data type	Usage
<i>Hconfig</i>	MQHCONFIG	input
<i>Options</i>	MQLONG	input
<i>QmgrName</i>	MQCHAR48	input
<i>ComponentDataLength</i>	MQLONG	input
<i>ComponentData</i>	MQBYTE x ComponentDataLength	input/output
<i>Version</i>	MQLONG	input/output
<i>CompCode</i>	MQLONG	output
<i>Reason</i>	MQLONG	output

MQZ_INIT_USERID

Purpose: This function is provided by a user ID service component, and is invoked by the queue manager during configuration of the component. It is expected to call MQZEP in order to provide information to the queue manager.

The function identifier, for MQZEP, is MQZID_INIT_USERID

Platforms: The *user identifier service* is available on MQSeries for OS/2 Warp only.

Table 149. MQZID_INIT_USERID function

Parameter	Data type	Usage
<i>Hconfig</i>	MQHCONFIG	input
<i>Options</i>	MQLONG	input
<i>QmgrName</i>	MQCHAR48	input
<i>ComponentDataLength</i>	MQLONG	input
<i>ComponentData</i>	MQBYTE × ComponentDataLength	input/output
<i>Version</i>	MQLONG	input/output
<i>CompCode</i>	MQLONG	output
<i>Reason</i>	MQLONG	output

MQZ_INSERT_NAME

Purpose: This function is provided by a name service component, and is invoked by the queue manager to insert an entry for the specified queue, containing the name of the queue manager that owns the queue.

The function identifier, for MQZEP, is MQZID_INSERT_NAME.

Table 150. MQZID_INSERT_NAME function

Parameter	Data type	Usage
<i>QmgrName</i>	MQCHAR48	input
<i>QName</i>	MQCHAR48	input
<i>ResolvedQmgrName</i>	MQCHAR48	input
<i>ComponentData</i>	MQBYTE × ComponentDataLength	input/output
<i>Continuation</i>	MQLONG	output
<i>CompCode</i>	MQLONG	output
<i>Reason</i>	MQLONG	output

MQZ_LOOKUP_NAME

MQZ_LOOKUP_NAME

Purpose: This function is provided by a name service component, and is invoked by the queue manager to retrieve the name of the owning queue manager, for a specified queue.

The function identifier, for MQZEP, is MQZID_LOOKUP_NAME.

Table 151. MQZID_LOOKUP_NAME function

Parameter	Data type	Usage
<i>QmgrName</i>	MQCHAR48	input
<i>QName</i>	MQCHAR48	input
<i>ResolvedQmgrName</i>	MQCHAR48	input
<i>ComponentData</i>	MQBYTE x ComponentDataLength	input/output
<i>Continuation</i>	MQLONG	output
<i>CompCode</i>	MQLONG	output
<i>Reason</i>	MQLONG	output

MQZ_SET_AUTHORITY and MQZ_SET_AUTHORITY_2

Purpose: The MQZ_SET_AUTHORITY function is provided by an authorization service component, and is invoked by the queue manager to set the authority that an entity has to access the specified object.

The MQZ_SET_AUTHORITY_2 function is provided by an MQZAS_VERSION_2 authorization service component, and is invoked by the queue manager to set the authority that an entity has to access the specified object.

The function identifier, for MQZEP, is MQZID_SET_AUTHORITY.

Table 152. MQZID_SET_AUTHORITY function

Parameter	Data type	Usage
<i>QmgrName</i>	MQCHAR48	input
<i>EntityName</i> (1)	MQCHAR12	input
<i>EntityType</i>	MQLONG	input
<i>ObjectName</i>	MQCHAR48	input
<i>ObjectType</i>	MQLONG	input
<i>Authority</i>	MQLONG	input
<i>ComponentData</i>	MQBYTE x ComponentDataLength	input/output
<i>Continuation</i>	MQLONG	output
<i>CompCode</i>	MQLONG	output
<i>Reason</i>	MQLONG	output
Note:		
1. This parameter is replaced by the <i>EntityData</i> parameter in the MQZ_SET_AUTHORITY_2 function, which uses the MQZED data type structure.		

MQZ_TERM_AUTHORITY

Purpose: This function is provided by an authorization service component, and is invoked by the queue manager when it no longer requires the services of the component.

The function identifier, for MQZEP, is MQZID_TERM_AUTHORITY.

Table 153. MQZID_TERM_AUTHORITY function

Parameter	Data type	Usage
<i>Hconfig</i>	MQHCONFIG	input
<i>Options</i>	MQLONG	input
<i>QmgrName</i>	MQCHAR48	input
<i>ComponentData</i>	MQBYTE x ComponentDataLength	input/output
<i>CompCode</i>	MQLONG	output
<i>Reason</i>	MQLONG	output

MQZ_TERM_NAME

MQZ_TERM_NAME

Purpose: This function is provided by a name service component, and is invoked by the queue manager when it no longer requires the services of the component.

The function identifier, for MQZEP, is MQZID_TERM_NAME.

Table 154. MQZID_TERM_NAME function

Parameter	Data type	Usage
<i>Hconfig</i>	MQHCONFIG	input
<i>Options</i>	MQLONG	input
<i>QmgrName</i>	MQCHAR48	input
<i>ComponentData</i>	MQBYTE x ComponentDataLength	input/output
<i>CompCode</i>	MQLONG	output
<i>Reason</i>	MQLONG	output

MQZ_TERM_USERID

Purpose: This function is provided by a user identifier service component, and is invoked by the queue manager when it no longer requires the services of the component.

The function identifier, for MQZEP, is MQZID_TERM_USERID.

Platforms: The *user identifier service* is available on MQSeries for OS/2 Warp only.

Table 155. MQZID_TERM_USERID function

Parameter	Data type	Usage
<i>Hconfig</i>	MQHCONFIG	input
<i>Options</i>	MQLONG	input
<i>QmgrName</i>	MQCHAR48	input
<i>ComponentData</i>	MQBYTE x ComponentDataLength	input/output
<i>CompCode</i>	MQLONG	output
<i>Reason</i>	MQLONG	output

MQZEP

Purpose: Invoked by a service component, during initialization, to add an entry point to the entry point vector for that service component.

Table 156. MQZEP function

Parameter	Data type	Usage
<i>Hconfig</i>	MQHCONFIG	input
<i>Function</i>	MQLONG	input
<i>EntryPoint</i>	PMQFUNC	input
<i>CompCode</i>	MQLONG	output
<i>Reason</i>	MQLONG	output

Chapter 7. AMI high-level interface calls

Full details of these calls can be found in the *MQSeries Application Messaging Interface* manual.

Note: The C++ and Java™ interfaces are documented only in the AMI reference manual.

amBackout

Purpose: Backs out a unit of work.

Table 157. amBackout call

Parameter	Data type	Usage
<i>Session</i>	AMHSES	input
<i>PolicyName</i>	AMSTR	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note: 1. In the C programming language this parameter is a pointer.		

amBegin

Purpose: Begins a unit of work.

Table 158. amBegin call

Parameter	Data type	Usage
<i>Session</i>	AMHSES	input
<i>policyName</i>	AMSTR	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note: 1. In the C programming language this parameter is a pointer.		

amBrowseMsg

amBrowseMsg

Purpose: Browses a message.

Table 159. amBrowseMsg call

Parameter	Data type	Usage
<i>Session</i>	AMHSES	input
<i>receiverName</i>	AMSTR	input
<i>PolicyName</i>	AMSTR	input
<i>options</i>	AMLONG	input
<i>buffLen</i>	AMLONG	input
<i>DataLen</i> (1)	AMLONG	output
<i>Data</i> (1)	AMBYTE	output
<i>rcvMsgName</i>	AMSTR	output
<i>senderName</i>	AMSTR	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amCommit

Purpose: Commits a unit of work.

Table 160. amCommit call

Parameter	Data type	Usage
<i>Session</i>	AMHSES	input
<i>PolicyName</i>	AMSTR	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amInitialize

Purpose: Creates and opens an AMI session. In the C programming language the call returns a session handle.

Table 161. amInitialize call

Parameter	Data type	Usage
<i>Name</i>	AMSTR	input
<i>PolicyName</i>	AMSTR	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amPublish

Purpose: Publishes a message to a publish/subscribe broker.

Table 162. amPublish call

Parameter	Data type	Usage
<i>Session</i>	AMHSES	input
<i>PublisherName</i>	AMSTR	input
<i>PolicyName</i>	AMSTR	input
<i>ReceiverName</i>	AMSTR	input
<i>TopicLen</i>	AMLONG	input
<i>Topic</i>	AMSTR	input
<i>DataLen</i>	AMLONG	input
<i>Data</i> (1)	AMBYTE	output
<i>PubMsgName</i>	AMSTR	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amReceiveFile

amReceiveFile

Purpose: Receives a message.

Table 163. amReceiveFile call

Parameter	Data type	Usage
<i>Session</i>	AMHSES	input
<i>receiverName</i>	AMSTR	input
<i>PolicyName</i>	AMSTR	input
<i>options</i>	AMLONG	input
<i>SelMsgName</i>	AMSTR	input
<i>FileNameLen</i>	AMLONG	input
<i>FileName</i> (1)	AMSTR	input
<i>RcvMsgName</i>	AMSTR	output
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amReceiveMsg

Purpose: Receives a message.

Table 164. amReceiveMsg call

Parameter	Data type	Usage
<i>Session</i>	AMHSES	input
<i>ReceiverName</i>	AMSTR	input
<i>PolicyName</i>	AMSTR	input
<i>SelMsgName</i>	AMSTR	input
<i>BuffLen</i>	AMLONG	input
<i>DataLength</i> (1)	AMLONG	input
<i>Data</i> (1)	AMBYTE	output
<i>RcvMsgName</i>	AMSTR	output
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amReceivePublication

Purpose: Receives a publication from a publish/subscribe broker.

Table 165. amReceivePublication call

Parameter	Data type	Usage
<i>Session</i>	AMHSES	input
<i>SubscriberName</i>	AMSTR	input
<i>PolicyName</i>	AMSTR	input
<i>SelMsgName</i>	AMSTR	input
<i>TopicBuffLen</i>	AMLONG	input
<i>BuffLen</i>	AMLONG	input
<i>TopicCount</i> (1)	AMLONG	output
<i>TopicLen</i> (1)	AMLONG	output
<i>FirstTopic</i>	AMSTRING	output
<i>DataLen</i> (1)	AMLONG	output
<i>Data</i> (1)	AMBYTE	output
<i>RcvMsgName</i>	AMSTR	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amReceiveRequest

amReceiveRequest

Purpose: Receives a request message.

Table 166. amReceiveRequest call

Parameter	Data type	Usage
<i>Session</i>	AMHSES	input
<i>ReceiverName</i>	AMSTR	input
<i>PolicyName</i>	AMSTR	input
<i>BuffLen</i>	AMLONG	input
<i>DataLength</i> (1)	AMLONG	input
<i>Data</i> (1)	AMBYTE	output
<i>RcvMsgName</i>	AMSTR	output
<i>SenderName</i>	AMSTR	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amSendFile

Purpose: Sends data from a file.

Table 167. amSendFile call

Parameter	Data type	Usage
<i>Session</i>	AMHSES	input
<i>SenderName</i>	AMSTR	input
<i>PolicyName</i>	AMSTR	input
<i>options</i>	AMLONG	input
<i>fileNameLen</i>	AMLONG	input
<i>fileName</i> (1)	AMBYTE	input
<i>SndMsgName</i>	AMSTR	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amSendMsg

Purpose: Sends a datagram (send and forget) message.

Table 168. amSendMsg call

Parameter	Data type	Usage
<i>Session</i>	AMHSES	input
<i>SenderName</i>	AMSTR	input
<i>PolicyName</i>	AMSTR	input
<i>DataLength</i>	AMLONG	input
<i>Data</i> (1)	AMBYTE	output
<i>SndMsgName</i>	AMSTR	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amSendRequest

Purpose: Sends a request message.

Table 169. amSendRequest call

Parameter	Data type	Usage
<i>Session</i>	AMHSES	input
<i>SenderName</i>	AMSTR	input
<i>PolicyName</i>	AMSTR	input
<i>DataLength</i>	AMLONG	input
<i>ReceiverName</i>	AMSTR	input
<i>DataLength</i>	AMLONG	input
<i>Data</i> (1)	AMBYTE	output
<i>SndMsgName</i>	AMSTR	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amSendResponse

amSendResponse

Purpose: Sends a response to a request message.

Table 170. amSendResponse call

Parameter	Data type	Usage
<i>Session</i>	AMHSES	input
<i>SenderName</i>	AMSTR	input
<i>PolicyName</i>	AMSTR	input
<i>RcvMsgName</i>	AMSTR	input
<i>DataLength</i>	AMLONG	input
<i>DataLength</i>	AMLONG	input
<i>Data</i> (1)	AMBYTE	output
<i>SndMsgName</i>	AMSTR	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amSubscribe

Purpose: Register a subscription with a publish/subscribe broker.

Table 171. amSubscribe call

Parameter	Data type	Usage
<i>Session</i>	AMHSES	input
<i>SubscriberName</i>	AMSTR	input
<i>PolicyName</i>	AMSTR	input
<i>ResponseName</i>	AMSTR	input
<i>TopicLen</i>	AMLONG	output
<i>Topic(1)</i>	AMSTRING	output
<i>FilterLen</i>	AMLONG	input
<i>Filter (1)</i>	AMSTR	output
<i>SubMsgName</i>	AMSTR	input
<i>CompCode (1)</i>	AMLONG	output
<i>Reason (1)</i>	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amTerminate

Purpose: Closes the session, deletes any implicitly created objects, and deletes the session.

Table 172. amTerminate call

Parameter	Data type	Usage
<i>Session (1)</i>	AMHSES	input
<i>PolicyName</i>	AMSTR	input
<i>CompCode (1)</i>	AMLONG	output
<i>Reason (1)</i>	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amUnsubscribe

amUnsubscribe

Purpose: Remove a subscription from a publish/subscribe broker.

Table 173. amUnsubscribe call

Parameter	Data type	Usage
<i>Session</i>	AMHSES	input
<i>SubscriberName</i>	AMSTR	input
<i>PolicyName</i>	AMSTR	input
<i>ReceiverName</i>	AMSTR	input
<i>TopicLen</i>	AMLONG	output
<i>Topic(1)</i>	AMSTRING	output
<i>FilterLen</i>	AMLONG	input
<i>Filter(1)</i>	AMSTR	output
<i>SubMsgName</i>	AMSTR	input
<i>CompCode (1)</i>	AMLONG	output
<i>Reason (1)</i>	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

Chapter 8. AMI object interface calls

Full details of these calls can be found in the *MQSeries Application Messaging Interface* manual.

Note: The C++ and Java interfaces are documented only in the AMI reference manual.

Distribution List interface functions

amDstClearErrorCodes

Purpose: Clears the error codes in the distribution list object.

Table 174. *amDstClearErrorCodes* call

Parameter	Data type	Usage
<i>hDistList</i>	AMHDST	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amDstClose

Purpose: Closes the distribution list.

Table 175. *amDstClose* call

Parameter	Data type	Usage
<i>hDistList</i>	AMHDST	input
<i>hPolicy</i>	AMHPOL	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amDstGetLastError

Purpose: Gets the information (completion and reason codes) from the last error in the distribution list object.

amDstGetLastError

Table 176. *amDstGetLastError* call

Parameter	Data type	Usage
<i>hDistList</i>	AMHDST	input
<i>buffLen</i>	AMLONG	input
<i>StringLen</i> (1)	AMLONG	output
<i>ErrorText</i> (1)	AMSTR	output
<i>Reason2</i> (1)	AMLONG	output
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note: 1. In the C programming language this parameter is a pointer.		

amDstGetName

Purpose: Gets the name of the distribution list object.

Table 177. *amDstGetName* call

Parameter	Data type	Usage
<i>hDistList</i>	AMHDST	input
<i>buffLen</i>	AMLONG	input
<i>NameLen</i> (1)	AMLONG	output
<i>Name</i> (1)	AMSTR	output
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note: 1. In the C programming language this parameter is a pointer.		

amDstGetSenderCount

Purpose: Gets a count of the number of sender services in the distribution list.

Table 178. *amDstGetSenderCount* call

Parameter	Data type	Usage
<i>hDistList</i>	AMHDST	input
<i>Count</i> (1)	AMLONG	output
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note: 1. In the C programming language this parameter is a pointer.		

amDstGetSenderHandle

Purpose: Returns the handle (type AMHSND) of a sender service in the distribution list object with the specified index.

Table 179. *amDstGetSenderHandle* call

Parameter	Data type	Usage
<i>hDistList</i>	AMHDST	input
<i>handleIndex</i>	AMLONG	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amDstOpen

Purpose: Opens the distribution list object for each of the destinations in the distribution list.

Table 180. *amDstOpen* call

Parameter	Data type	Usage
<i>hDistList</i>	AMHDST	input
<i>hPolicy</i>	AMHPOL	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amDstSend

amDstSend

Purpose: Sends a message to each sender in the distribution list.

Table 181. *amDstSend* call

Parameter	Data type	Usage
<i>hDistList</i>	AMHDST	input
<i>hPolicy</i>	AMHPOL	input
<i>hReceiver</i>	AMHRCV	input
<i>dataLen</i>	AMLONG	input
<i>Data</i> (1)	AMBYTE	input
<i>hMsg</i>	AMHMSG	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amDstSendFile

Purpose: Sends data from a file to each sender in the distribution list.

Table 182. *amDstSendFile* call

Parameter	Data type	Usage
<i>hDistList</i>	AMHDST	input
<i>hPolicy</i>	AMHPOL	input
<i>options</i>	AMLONG	input
<i>directoryLen</i>	AMLONG	input
<i>directory</i> (1)	AMSTR	input
<i>fileNameLen</i>	AMLONG	input
<i>FileName</i> (1)	AMSTR	input
<i>hMsg</i>	AMHMSG	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

Message Interface functions

amMsgAddElement

Purpose: Adds a name/value element to a message.

Table 183. amMsgAddElement call

Parameter	Data type	Usage
<i>hMsg</i>	AMHMSG	input
<i>Elem</i> (1)	AMELEM	input
<i>Options</i>	AMLONG	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amMsgAddFilter

Purpose: Adds a filter to a publish/subscribe message.

Table 184. amMsgAddFilter call

Parameter	Data type	Usage
<i>hMsg</i>	AMHMSG	input
<i>FilterLen</i>	AMLONG	input
<i>Filter</i> (1)	AMSTR	output
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amMsgAddTopic

amMsgAddTopic

Purpose: Adds a topic to a publish/subscribe message.

Table 185. amMsgAddTopic call

Parameter	Data type	Usage
<i>hMsg</i>	AMHMSG	input
<i>topicLen</i>	AMLONG	input
<i>Topic</i> (1)	AMSTR	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note: 1. In the C programming language this parameter is a pointer.		

amMsgClearErrorCodes

Purpose: Clears the error codes in the message object.

Table 186. amMsgClearErrorCodes call

Parameter	Data type	Usage
<i>hMsg</i>	AMHMSG	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note: 1. In the C programming language this parameter is a pointer.		

amMsgDeleteElement

Purpose: Clears the error codes in the message object.

Table 187. amMsgDeleteElement call

Parameter	Data type	Usage
<i>hMsg</i>	AMHMSG	input
<i>elemIndex</i>	AMLONG	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note: 1. In the C programming language this parameter is a pointer.		

amMsgDeleteFilter

Purpose: Deletes a filter from a publish/subscribe message at the specified index. Indexing is within all filters.

Table 188. amMsgDeleteFilter call

Parameter	Data type	Usage
<i>hMsg</i>	AMHMSG	input
<i>filterIndex</i>	AMLONG	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amMsgDeleteNamedElement

Purpose: Deletes the element with the specified name and index from the message.

Table 189. amMsgDeleteNamedElement call

Parameter	Data type	Usage
<i>hMsg</i>	AMHMSG	input
<i>nameIndex</i>	AMLONG	input
<i>nameLen</i>	AMLONG	input
<i>Name</i> (1)	AMLONG	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amMsgDeleteTopic

Purpose: Deletes the topic with the specified index.

Table 190. amMsgDeleteTopic call

Parameter	Data type	Usage
<i>hMsg</i>	AMHMSG	input
<i>topicIndex</i>	AMLONG	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amMsgGetCCSID

amMsgGetCCSID

Purpose: Gets the coded character set identifier of the message.

Table 191. amMsgGetCCSID call

Parameter	Data type	Usage
<i>hMsg</i>	AMHMSG	input
<i>CCSID</i> (1)	AMLONG	output
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note: 1. In the C programming language this parameter is a pointer.		

amMsgGetCorrelId

Purpose: Gets the correlation identifier of the message.

Table 192. amMsgGetCorrelId call

Parameter	Data type	Usage
<i>hMsg</i>	AMHMSG	input
<i>buffLen</i>	AMLONG	input
<i>CorrelIdLen</i> (1)	AMLONG	output
<i>CorrelId</i> (1)	AMBYTE	output
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note: 1. In the C programming language this parameter is a pointer.		

amMsgGetDataLength

Purpose: Gets the data length of the message.

Table 193. amMsgGetDataLength call

Parameter	Data type	Usage
<i>hMsg</i>	AMHMSG	input
<i>Length</i> (1)	AMBYTE	output
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note: 1. In the C programming language this parameter is a pointer.		

amMsgGetDataOffset

Purpose: Gets the current offset in the message data for reading or writing data bytes.

Table 194. amMsgGetDataOffset call

Parameter	Data type	Usage
<i>hMsg</i>	AMHMSG	input
<i>Offset</i> (1)	AMLONG	output
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amMsgGetElement

Purpose: Gets an element from a message (such as a publish/subscribe message).

Table 195. amMsgGetElement call

Parameter	Data type	Usage
<i>hMsg</i>	AMHMSG	input
<i>ElemIndex</i>	AMLONG	input
<i>Elem</i> (1)	AMELEM	output
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amMsgGetElementCCSID

Purpose: Gets the coded character set identifier of the message elements (including topic and filter data).

amMsgGetElementCCSID

Table 196. *amMsgGetElementCCSID* call

Parameter	Data type	Usage
<i>hMsg</i>	AMHMSG	input
<i>ElementCCSID</i> (1)	AMLONG	output
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amMsgGetElementCount

Purpose: Gets the total number of elements in a message (such as a publish/subscribe message).

Table 197. *amMsgGetElementCount* call

Parameter	Data type	Usage
<i>hMsg</i>	AMHMSG	input
<i>Count</i> (1)	AMLONG	output
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amMsgGetEncoding

Purpose: Gets the value used to encode numeric data types for the message.

Table 198. *amMsgGetEncoding* call

Parameter	Data type	Usage
<i>hMsg</i>	AMHMSG	input
<i>Encoding</i> (1)	AMLONG	output
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amMsgGetFilter

Purpose: Gets a filter from a publish/subscribe message, at the specified index. Indexing is within all filters.

Table 199. amMsgGetFilter call

Parameter	Data type	Usage
<i>hMsg</i>	AMHMSG	input
<i>filterIndex</i>	AMLONG	input
<i>buffLen</i>	AMLONG	input
<i>FilterLen</i> (1)	AMLONG	output
<i>Filter</i> (1)	AMSTR	output
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amMsgGetFilterCount

Purpose: Gets the total number of filters in a publish/subscribe message.

Table 200. amMsgGetFilterCount call

Parameter	Data type	Usage
<i>hMsg</i>	AMHMSG	input
<i>Count</i> (1)	AMLONG	output
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amMsgGetFormat

Purpose: Gets the format of the message.

Table 201. amMsgGetFormat call

Parameter	Data type	Usage
<i>hMsg</i>	AMHMSG	input
<i>buffLen</i>	AMLONG	input
<i>FormatLen</i> (1)	AMLONG	output
<i>Format</i> (1)	AMSTR	output
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amMsgGetGroupStatus

Purpose: Gets the group status of the message. This indicates whether the message is in a group, and if it is the first, middle, last or only one in the group.

Table 202. *amMsgGetGroupStatus* call

Parameter	Data type	Usage
<i>hMsg</i>	AMHMSG	input
<i>Status</i> (1)	AMLONG	output
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note: 1. In the C programming language this parameter is a pointer.		

amMsgGetLastError

Purpose: Gets the information (completion and reason codes) from the last error for the message object.

Table 203. *amMsgGetLastError* call

Parameter	Data type	Usage
<i>hMsg</i>	AMHMSG	input
<i>buffLen</i>	AMLONG	input
<i>StringLen</i> (1)	AMLONG	output
<i>ErrorText</i> (1)	AMSTR	output
<i>Reason2</i> (1)	AMLONG	output
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note: 1. In the C programming language this parameter is a pointer.		

amMsgGetMsgId

Purpose: Gets the message identifier of the message.

Table 204. amMsgGetMsgId call

Parameter	Data type	Usage
<i>hMsg</i>	AMHMSG	input
<i>buffLen</i>	AMLONG	input
<i>MsgIdLen</i> (1)	AMLONG	output
<i>MsgId</i> (1)	AMBYTE	output
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amMsgGetName

Purpose: Gets the name of the message object.

Table 205. amMsgGetName call

Parameter	Data type	Usage
<i>hMsg</i>	AMHMSG	input
<i>buffLen</i>	AMLONG	input
<i>NameLen</i> (1)	AMLONG	output
<i>Name</i> (1)	AMSTR	output
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amMsgGetNamedElement

Purpose: Gets the named element from a message (such as a publish/subscribe message) with the specified index.

amMsgGetNamedElement

Table 206. *amMsgGetNamedElement* call

Parameter	Data type	Usage
<i>hMsg</i>	AMHMSG	input
<i>nameIndex</i>	AMLONG	input
<i>nameLen</i>	AMLONG	input
<i>Name</i> (1)	AMSTR	input
<i>Element</i> (1)	AMELEM	output
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note: 1. In the C programming language this parameter is a pointer.		

amMsgGetNamedElementCount

Purpose: Gets the number of elements in a message with a specified name.

Table 207. *amMsgGetNamedElementCount* call

Parameter	Data type	Usage
<i>hMsg</i>	AMHMSG	input
<i>nameLen</i>	AMLONG	input
<i>Name</i> (1)	AMSTR	input
<i>Count</i> (1)	AMLONG	output
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note: 1. In the C programming language this parameter is a pointer.		

amMsgGetReportCode

Purpose: Gets the feedback from a message of type MQMT_REPORT.

Table 208. *amMsgGetReportCode* call

Parameter	Data type	Usage
<i>hMsg</i>	AMHMSG	input
<i>Code</i> (1)	AMLONG	output
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note: 1. In the C programming language this parameter is a pointer.		

amMsgGetTopic

Purpose: Gets a topic from a publish/subscribe message, at the specified index. Indexing is within all topics.

Table 209. amMsgGetTopic call

Parameter	Data type	Usage
<i>hMsg</i>	AMHMSG	input
<i>topicIndex</i>	AMLONG	input
<i>buffLen</i>	AMLONG	input
<i>TopicLen</i> (1)	AMLONG	output
<i>Topic</i> (1)	AMSTR	output
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amMsgGetTopicCount

Purpose: Gets the total number of topics in a publish/subscribe message.

Table 210. amMsgGetTopicCount call

Parameter	Data type	Usage
<i>hMsg</i>	AMHMSG	input
<i>Count</i> (1)	AMLONG	output
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amMsgGetType

amMsgGetType

Purpose: Gets the message type from a message.

Table 211. *amMsgGetType* call

Parameter	Data type	Usage
<i>hMsg</i>	AMHMSG	input
<i>Type</i> (1)	AMLONG	output
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note: 1. In the C programming language this parameter is a pointer.		

amMsgReadBytes

Purpose: Reads up to the specified number of data bytes from the message object, starting at the current data offset (which must be positioned before the end of the data for the reading to be successful). Use **msgSetDataOffset** to set the data offset. **amMsgReadBytes** will advance the data offset by the number of bytes read, leaving the offset immediately after the last byte read.

Table 212. *amMsgReadBytes* call

Parameter	Data type	Usage
<i>hMsg</i>	AMHMSG	input
<i>readLen</i>	AMLONG	input
<i>BytesRead</i> (1)	AMLONG	output
<i>Data</i> (1)	AMBYTE	output
<i>CompCode</i> (1)	AMLONG	input
<i>Reason</i> (1)	AMLONG	output
Note: 1. In the C programming language this parameter is a pointer.		

amMsgReset

Purpose: Resets the message object to its initial state.

Table 213. amMsgReset call

Parameter	Data type	Usage
<i>hMsg</i>	AMHMSG	input
<i>options</i>	AMLONG	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amMsgSetCCSID

Purpose: Sets the coded character set identifier of the message.

Table 214. amMsgSetCCSID call

Parameter	Data type	Usage
<i>hMsg</i>	AMHMSG	input
<i>CCSID</i>	AMLONG	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amMsgSetCorrelId

Purpose: Sets the correlation identifier of the message.

Table 215. amMsgSetCorrelId call

Parameter	Data type	Usage
<i>hMsg</i>	AMHMSG	input
<i>CorrelIdLen</i>	AMLONG	input
<i>CorrelId</i> (1)	AMBYTE	Input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amMsgSetDataOffset

amMsgSetDataOffset

Purpose: Sets the data offset for reading or writing byte data. If the data offset is greater than the current data length, it is valid to write data into the message at that offset, but an attempt to read data will result in an error.

Table 216. *amMsgSetDataOffset* call

Parameter	Data type	Usage
<i>hMsg</i>	AMHMSG	input
<i>offset</i>	AMLONG	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note: 1. In the C programming language this parameter is a pointer.		

amMsgSetElementCCSID

Purpose: Sets the coded character set identifier of the message element.

Table 217. *amMsgSetElementCCSID* call

Parameter	Data type	Usage
<i>hMsg</i>	AMHMSG	input
<i>elementCCSID</i>	AMLONG	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note: 1. In the C programming language this parameter is a pointer.		

amMsgSetEncoding

Purpose: Sets the encoding of the data in the message.

Table 218. *amMsgSetEncoding* call

Parameter	Data type	Usage
<i>hMsg</i>	AMHMSG	input
<i>encoding</i>	AMLONG	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note: 1. In the C programming language this parameter is a pointer.		

amMsgSetFormat

Purpose: Sets the format of the message.

Table 219. *amMsgSetFormat* call

Parameter	Data type	Usage
<i>hMsg</i>	AMHMSG	input
<i>formatLen</i>	AMLONG	input
<i>Format</i> (1)	AMSTR	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amMsgSetGroupStatus

Purpose: Sets the group status of the message.

Table 220. *amMsgSetGroupStatus* call

Parameter	Data type	Usage
<i>hMsg</i>	AMHMSG	input
<i>status</i>	AMLONG	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amMsgWriteBytes

Purpose: Writes the specified number of data bytes to the message object.

Table 221. *amMsgWriteBytes* call

Parameter	Data type	Usage
<i>hMsg</i>	AMHMSG	input
<i>writeLen</i>	AMLONG	input
<i>ByteData</i> (1)	AMBYTE	Input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

Policy interface functions

Policy interface functions

amPolClearErrorCodes

Purpose: Clears the error codes in the policy object.

Table 222. amPolClearErrorCodes call

Parameter	Data type	Usage
<i>hPolicy</i>	AMHPOL	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note: 1. In the C programming language this parameter is a pointer.		

amPolGetLastError

Purpose: Gets the information (completion and reason codes) from the last error for the policy object.

Table 223. amPolGetLastError call

Parameter	Data type	Usage
<i>hPolicy</i>	AMHPOL	input
<i>buffLen</i>	AMLONG	input
<i>StringLen</i> (1)	AMLONG	output
<i>ErrorText</i> (1)	AMSTR	output
<i>Reason2</i> (1)	AMLONG	output
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note: 1. In the C programming language this parameter is a pointer.		

amPolGetName

Purpose: Gets the information (completion and reason codes) from the last error for the policy object.

Table 224. amPolGetName call

Parameter	Data type	Usage
<i>hPolicy</i>	AMHPOL	input
<i>buffLen</i>	AMLONG	input
<i>NameLen</i> (1)	AMLONG	output
<i>Name</i> (1)	AMSTR	output
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amPolGetWaitTime

Purpose: Returns the wait time (in ms) set for this policy.

Table 225. amPolGetWaitTime call

Parameter	Data type	Usage
<i>hPolicy</i>	AMHPOL	input
<i>WaitTime</i> (1)	AMLONG	output
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amPolSetWaitTime

Purpose: Sets the wait time (in ms) for any receive function using this policy.

Table 226. amPolSetWaitTime call

Parameter	Data type	Usage
<i>hPolicy</i>	AMHPOL	input
<i>waitTime</i>	AMLONG	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

Publisher interface functions

Publisher interface functions

amPubClearErrorCodes

Purpose: Clears the error codes in the publisher object.

Table 227. amPubClearErrorCodes call

Parameter	Data type	Usage
<i>hPublisher</i>	AMHPUB	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note: 1. In the C programming language this parameter is a pointer.		

amPubClose

Purpose: Closes the publisher service.

Table 228. amPubClose call

Parameter	Data type	Usage
<i>hPublisher</i>	AMHPUB	input
<i>hPolicy</i>	AMHPOL	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note: 1. In the C programming language this parameter is a pointer.		

amPubGetCCSID

Purpose: Gets the coded character set identifier to be used for the publisher service.

Table 229. amPubGetCCSID call

Parameter	Data type	Usage
<i>hPublisher</i>	AMHPUB	input
<i>CCSID</i> (1)	AMLONG	output
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note: 1. In the C programming language this parameter is a pointer.		

amPubGetEncoding

Purpose: Gets the value to be used to encode numeric data types for the publisher service.

Table 230. amPubGetEncoding call

Parameter	Data type	Usage
<i>hPublisher</i>	AMHPUB	input
<i>Encoding</i> (1)	AMLONG	output
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amPubGetLastError

Purpose: Gets the information (completion and reason codes) from the last error for the publisher object.

Table 231. amPubGetLastError call

Parameter	Data type	Usage
<i>hPublisher</i>	AMHPUB	input
<i>buffLen</i>	AMLONG	input
<i>StringLen</i> (1)	AMLONG	output
<i>ErrorText</i> (1)	AMSTR	output
<i>Reason2</i> (1)	AMLONG	output
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amPubGetName

amPubGetName

Purpose: Gets the name of the publisher service.

Table 232. amPubGetName call

Parameter	Data type	Usage
<i>hPublisher</i>	AMHPUB	input
<i>bufLen</i>	AMLONG	input
<i>NameLen</i> (1)	AMLONG	output
<i>Name</i> (1)	AMSTR	output
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note: 1. In the C programming language this parameter is a pointer.		

amPubOpen

Purpose: Opens the publisher service.

Table 233. amPubOpen call

Parameter	Data type	Usage
<i>hPublisher</i>	AMHPUB	input
<i>hPolicy</i>	AMHPOL	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note: 1. In the C programming language this parameter is a pointer.		

amPubPublish

Purpose: Publishes a message using the publisher service.

Table 234. amPubPublish call

Parameter	Data type	Usage
<i>hPublisher</i>	AMHPUB	input
<i>hPolicy</i>	AMHPOL	input
<i>hResponse</i>	AMHRCV	input
<i>hPubMsg</i>	AMHMSG	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

Receiver interface functions

Receiver interface functions

amRcvBrowse

Purpose: Browses a message.

Table 235. amRcvBrowse call

Parameter	Data type	Usage
<i>hReceiver</i>	AMHRCV	input
<i>hPolicy</i>	AMHPOL	input
<i>options</i>	AMLONG	input
<i>buffLen</i>	AMLONG	input
<i>DataLen</i> (1)	AMLONG	output
<i>Data</i> (1)	AMBYTE	output
<i>hRcvMsg</i>	AMHMSG	output
<i>hSender</i>	AMHSND	output
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note: 1. In the C programming language this parameter is a pointer.		

amRcvBrowseSelect

Purpose: Browses a message with a selection message used to provide the selection criterion.

Table 236. amRcvBrowseSelect call

Parameter	Data type	Usage
<i>hReceiver</i>	AMHRCV	input
<i>hPolicy</i>	AMHPOL	input
<i>options</i>	AMLONG	input
<i>selMsg</i>	AMHMSG	input
<i>buffLen</i>	AMLONG	input
<i>DataLen</i> (1)	AMLONG	output
<i>Data</i> (1)	AMBYTE	output
<i>hRcvMsg</i>	AMHMSG	output
<i>hSender</i>	AMHSND	output
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amRcvClearErrorCodes

Purpose: Clears the error codes in the receiver service object.

Table 237. amRcvClearErrorCodes call

Parameter	Data type	Usage
<i>hReceiver</i>	AMHRCV	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amRcvClose

amRcvClose

Purpose: Closes the receiver service.

Table 238. amRcvClose call

Parameter	Data type	Usage
<i>hReceiver</i>	AMHRCV	input
<i>hPolicy</i>	AMHPOL	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note: 1. In the C programming language this parameter is a pointer.		

amRcvGetDefnType

Purpose: Gets the definition type of the receiver service.

Table 239. amRcvGetDefnType call

Parameter	Data type	Usage
<i>hReceiver</i>	AMHRCV	input
<i>Type</i> (1)	AMLONG	output
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note: 1. In the C programming language this parameter is a pointer.		

amRcvGetLastError

Purpose: Gets the information (completion and reason codes) from the last error for the receiver object.

Table 240. amRcvGetLastError call

Parameter	Data type	Usage
<i>hReceiver</i>	AMHRCV	input
<i>buffLen</i>	AMLONG	input
<i>StringLen(1)</i>	AMLONG	output
<i>ErrorText(1)</i>	AMSTR	output
<i>Reason2(1)</i>	AMLONG	output
<i>CompCode(1)</i>	AMLONG	output
<i>Reason(1)</i>	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amRcvGetName

Purpose: Gets the name of the receiver service.

Table 241. amRcvGetName call

Parameter	Data type	Usage
<i>hReceiver</i>	AMHRCV	input
<i>buffLen</i>	AMLONG	input
<i>NameLen(1)</i>	AMLONG	output
<i>Name(1)</i>	AMSTR	output
<i>CompCode(1)</i>	AMLONG	output
<i>Reason(1)</i>	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amRcvGetQueueName

amRcvGetQueueName

Purpose: Gets the queue name of the receiver service.

Table 242. amRcvGetQueueName call

Parameter	Data type	Usage
<i>hReceiver</i>	AMHRCV	input
<i>buffLen</i>	AMLONG	input
<i>NameLen(1)</i>	AMLONG	output
<i>QueueName(1)</i>	AMSTR	output
<i>CompCode(1)</i>	AMLONG	output
<i>Reason(1)</i>	AMLONG	output
Note: 1. In the C programming language this parameter is a pointer.		

amRcvOpen

Purpose: Opens the receiver service.

Table 243. amRcvOpen call

Parameter	Data type	Usage
<i>hReceiver</i>	AMHRCV	input
<i>hPolicy</i>	AMHPOL	input
<i>CompCode(1)</i>	AMLONG	output
<i>Reason(1)</i>	AMLONG	output
Note: 1. In the C programming language this parameter is a pointer.		

amRcvReceive

Purpose: Receives a message.

Table 244. *amRcvReceive* call

Parameter	Data type	Usage
<i>hReceiver</i>	AMHRCV	input
<i>hPolicy</i>	AMHPOL	input
<i>hSelMsg</i>	AMHMSG	input
<i>buffLen</i>	AMLONG	input
<i>DataLen(1)</i>	AMLONG	output
<i>Data(1)</i>	AMBYTE	output
<i>hRcvMsg</i>	AMHMSG	output
<i>hSender</i>	AMHSND	output
<i>CompCode(1)</i>	AMLONG	output
<i>Reason(1)</i>	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amRcvReceiveFile

Purpose: Receives file message data into a file.

Table 245. *amRcvReceiveFile* call

Parameter	Data type	Usage
<i>hReceiver</i>	AMHRCV	input
<i>hPolicy</i>	AMHPOL	input
<i>options</i>	AMLONG	input
<i>hSelMsg</i>	AMHMSG	input
<i>directoryLen</i>	AMLONG	input
<i>directory</i>	AMSTR	input
<i>fileNameLen</i>	AMLONG	input
<i>FileName(1)</i>	AMSTR	input
<i>hRcvMsg</i>	AMHMSG	output
<i>CompCode(1)</i>	AMLONG	output
<i>Reason(1)</i>	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amRcvSetQueueName

amRcvSetQueueName

Purpose: Sets the queue name of the receiver service, when this encapsulates a model queue.

Table 246. amRcvSetQueueName call

Parameter	Data type	Usage
<i>hReceiver</i>	AMHRCV	input
<i>hNameLen</i>	AMLONG	input
<i>QueueName(1)</i>	AMSTR	input
<i>CompCode(1)</i>	AMLONG	output
<i>Reason(1)</i>	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

Session interface functions
amSesBegin

Purpose: Begins a unit of work.

Table 247. amSesBegin call

Parameter	Data type	Usage
<i>hSess</i>	AMHSES	input
<i>hPolicy</i>	AMHPOL	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amSesClearErrorCodes

Purpose: Clears the error codes in the session object.

Table 248. amSesClearErrorCodes call

Parameter	Data type	Usage
<i>hSes</i>	AMHSES	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amSesClose

Purpose: Closes the session object and all open objects owned by the session, and disconnects from the underlying message transport (MQSeries).

Table 249. amSesClose call

Parameter	Data type	Usage
<i>hSess</i>	AMHSES	input
<i>hPolicy</i>	AMHPOL	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amSesCommit

amSesCommit

Purpose: Commits a unit of work that was started by **amSesBegin**, or by sending or receiving a message under syncpoint control as defined in the policy options for the send or receive request.

Table 250. *amSesCommit* call

Parameter	Data type	Usage
<i>hSess</i>	AMHSES	input
<i>hPolicy</i>	AMHPOL	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note: 1. In the C programming language this parameter is a pointer.		

amSesCreate

Purpose: Creates the session and system default objects.

Table 251. *amSesCreate* call

Parameter	Data type	Usage
<i>name</i> (1)	AMSTR	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note: 1. In the C programming language this parameter is a pointer.		

amSesCreateDistList

Purpose: Creates a distribution list object. A distribution list handle (of type AMHDST) is returned.

Table 252. *amSesCreateDistList* call

Parameter	Data type	Usage
<i>hSess</i>	AMHSES	input
<i>name</i> (1)	AMSTR	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note: 1. In the C programming language this parameter is a pointer.		

amSesCreateMessage

Purpose: Creates a message object. A message handle (of type AMHMSG) is returned.

Table 253. *amSesCreateMessage* call

Parameter	Data type	Usage
<i>hSess</i>	AMHSES	input
<i>name</i> (1)	AMSTR	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amSesCreatePolicy

Purpose: Creates a policy object. A policy handle (of type AMHPOL) is returned.

Table 254. *amSesCreatePolicy* call

Parameter	Data type	Usage
<i>hSess</i>	AMHSES	input
<i>name</i> (1)	AMSTR	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amSesCreatePublisher

Purpose: Creates a publisher object. A publisher handle (of type AMHPUB) is returned.

Table 255. *amSesCreatePublisher* call

Parameter	Data type	Usage
<i>hSess</i>	AMHSES	input
<i>name</i> (1)	AMSTR	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amSesCreateReceiver

amSesCreateReceiver

Purpose: Creates a receiver service object. A receiver handle (of type AMHRCV) is returned.

Table 256. amSesCreateReceiver call

Parameter	Data type	Usage
<i>hSess</i>	AMHSES	input
<i>name</i> (1)	AMSTR	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note: 1. In the C programming language this parameter is a pointer.		

amSesCreateSender

Purpose: Creates a sender service object. A sender handle (of type AMHSND) is returned.

Table 257. amSesCreateSender call

Parameter	Data type	Usage
<i>hSess</i>	AMHSES	input
<i>name</i> (1)	AMSTR	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note: 1. In the C programming language this parameter is a pointer.		

amSesCreateSubscriber

Purpose: Creates a subscriber object. A subscriber handle (of type AMHSUB) is returned.

Table 258. amSesCreateSubscriber call

Parameter	Data type	Usage
<i>hSess</i>	AMHSES	input
<i>name</i> (1)	AMSTR	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note: 1. In the C programming language this parameter is a pointer.		

amSesDelete

Purpose: Deletes the session object. Performs an implicit close if the session is open. This closes and deletes the session and all objects owned by it.

Table 259. *amSesDelete* call

Parameter	Data type	Usage
<i>hSess</i> (1)	AMHSES	input/output
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amSesDeleteDistList

Purpose: Deletes a distribution list object, and performs an implicit close if the distribution list is open.

Table 260. *amSesDeleteDistList* call

Parameter	Data type	Usage
<i>hSess</i>	AMHSES	input
<i>hDistList</i> (1)	AMHDST	input/output
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amSesDeleteMessage

Purpose: Deletes a message object.

Table 261. *amSesDeleteMessage* call

Parameter	Data type	Usage
<i>hSess</i>	AMHSES	input
<i>hMsg</i> (1)	AMHMSG	input/output
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amSesDeletePolicy

amSesDeletePolicy

Purpose: Deletes a policy object.

Table 262. amSesDeletePolicy call

Parameter	Data type	Usage
<i>hSess</i>	AMHSES	input
<i>hPolicy</i> (1)	AMHPOL	input/output
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note: 1. In the C programming language this parameter is a pointer.		

amSesDeletePublisher

Purpose: Deletes a publisher object, and performs an implicit close if the publisher is open.

Table 263. amSesDeletePublisher call

Parameter	Data type	Usage
<i>hSess</i>	AMHSES	input
<i>hPub</i> (1)	AMHPUB	input/output
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note: 1. In the C programming language this parameter is a pointer.		

amSesDeleteReceiver

Purpose: Deletes a receiver object, and performs an implicit close if the receiver is open.

Table 264. amSesDeleteReceiver call

Parameter	Data type	Usage
<i>hSess</i>	AMHSES	input
<i>hReceiver</i> (1)	AMHPUB	input/output
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note: 1. In the C programming language this parameter is a pointer.		

amSesDeleteSender

Purpose: Deletes a sender object, and performs an implicit close if the sender is open.

Table 265. amSesDeleteSender call

Parameter	Data type	Usage
<i>hSess</i>	AMHSES	input
<i>hSender</i> (1)	AMHSND	input/output
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amSesDeleteSubscriber

Purpose: Deletes a subscriber object, and performs an implicit close if the subscriber is open.

Table 266. amSesDeleteSubscriber call

Parameter	Data type	Usage
<i>hSess</i>	AMHSES	input
<i>hSubscriber</i> (1)	AMHSUB	input/output
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amSesGetDistListHandle

Purpose: Returns the handle of the distribution list object (of type AMHDST) with the specified name.

Table 267. amSesGetDistListHandle call

Parameter	Data type	Usage
<i>hSess</i>	AMHSES	input
<i>name</i> (1)	AMSTR	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amSesGetLastError

amSesGetLastError

Purpose: Gets the information (completion and reason codes) from the last error for the session.

Table 268. amSesGetLastError call

Parameter	Data type	Usage
<i>hSess</i>	AMHSES	input
<i>buffLen</i>	AMLONG	input
<i>StringLen</i> (1)	AMLONG	input
<i>ErrorText</i> (1)	AMSTR	input
<i>Reason2</i> (1)	AMLONG	output
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note: 1. In the C programming language this parameter is a pointer.		

amSesGetMessageHandle

Purpose: Returns the handle of the message object (of type AMHMSG) with the specified name.

Table 269. amSesGetMessageHandle call

Parameter	Data type	Usage
<i>hSess</i>	AMHSES	input
<i>name</i> (1)	AMSTR	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note: 1. In the C programming language this parameter is a pointer.		

amSesGetPolicyHandle

Purpose: Returns the handle of the policy object (of type AMHPOL) with the specified name.

Table 270. amSesGetPolicyHandle call

Parameter	Data type	Usage
<i>hSess</i>	AMHSES	input
<i>name</i> (1)	AMSTR	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amSesGetPublisherHandle

Purpose: Returns the handle of the publisher object (of type AMHPUB) with the specified name.

Table 271. amSesGetPublisherHandle call

Parameter	Data type	Usage
<i>hSess</i>	AMHSES	input
<i>name</i> (1)	AMSTR	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amSesGetReceiverHandle

Purpose: Returns the handle of the receiver service object (of type AMHRCV) with the specified name.

Table 272. amSesGetReceiverHandle call

Parameter	Data type	Usage
<i>hSess</i>	AMHSES	input
<i>name</i> (1)	AMSTR	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amSesGetSenderHandle

Purpose: Returns the handle of the sender service object (of type AMHSND) with the specified name.

amSesGetSenderHandle

Table 273. amSesGetSenderHandle call

Parameter	Data type	Usage
<i>hSess</i>	AMHSES	input
<i>name</i> (1)	AMSTR	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note: 1. In the C programming language this parameter is a pointer.		

amSesGetSubscriberHandle

Purpose: Returns the handle of the subscriber service object (of type AMHSUB) with the specified name.

Table 274. amSesGetSubscriberHandle call

Parameter	Data type	Usage
<i>hSess</i>	AMHSES	input
<i>name</i> (1)	AMSTR	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note: 1. In the C programming language this parameter is a pointer.		

amSesOpen

Purpose: Opens the session object using the specified policy options.

Table 275. amSesOpen call

Parameter	Data type	Usage
<i>hSess</i>	AMHSES	input
<i>hPolicy</i>	AMHPOL	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note: 1. In the C programming language this parameter is a pointer.		

amSesRollback

Purpose: Rolls back a unit of work.

Table 276. amSesRollback call

Parameter	Data type	Usage
<i>hSess</i>	AMHSES	input
<i>hPolicy</i>	AMHPOL	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note: 1. In the C programming language this parameter is a pointer.		

Sender interface functions

Sender interface functions

amSndClearErrorCodes

Purpose: Clears the error codes in the sender object.

Table 277. amSndClearErrorCodes call

Parameter	Data type	Usage
<i>hSender</i>	AMHSND	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note: 1. In the C programming language this parameter is a pointer.		

amSndClose

Purpose: Closes the sender service.

Table 278. amSndClose call

Parameter	Data type	Usage
<i>hSender</i>	AMHSND	input
<i>hPolicy</i>	AMHPOL	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note: 1. In the C programming language this parameter is a pointer.		

amSndGetCCSID

Purpose: Gets the coded character set identifier to be used for the sender service.

Table 279. amSndGetCCSID call

Parameter	Data type	Usage
<i>hSender</i>	AMHSND	input
<i>CCSID</i> (1)	AMLONG	output
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note: 1. In the C programming language this parameter is a pointer.		

amSndGetEncoding

Purpose: Gets the value to be used to encode numeric data types for the sender service.

Table 280. *amSndGetEncoding* call

Parameter	Data type	Usage
<i>hSender</i>	AMHSND	input
<i>Encoding</i> (1)	AMLONG	output
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amSndGetLastError

Purpose: Gets the information (completion and reason codes) from the last error for the sender object.

Table 281. *amSndGetLastError* call

Parameter	Data type	Usage
<i>hSender</i>	AMHSND	input
<i>buffLen</i>	AMLONG	input
<i>StringLen</i> (1)	AMLONG	output
<i>ErrorText</i> (1)	AMSTR	output
<i>Reason2</i> (1)	AMLONG	output
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amSndGetName

amSndGetName

Purpose: Gets the name of the sender service.

Table 282. amSndGetName call

Parameter	Data type	Usage
<i>hSender</i>	AMHSND	input
<i>buffLen</i>	AMLONG	input
<i>NameLen</i> (1)	AMLONG	output
<i>Name</i> (1)	AMSTR	output
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note: 1. In the C programming language this parameter is a pointer.		

amSndOpen

Purpose: Opens the sender service.

Table 283. amSndOpen call

Parameter	Data type	Usage
<i>hSender</i>	AMHSND	input
<i>hPolicy</i>	AMHPOL	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note: 1. In the C programming language this parameter is a pointer.		

amSndSend

Purpose: Sends a message to the destination specified by the sender service.

Table 284. amSndSend call

Parameter	Data type	Usage
<i>hSender</i>	AMHSND	input
<i>hPolicy</i>	AMHPOL	input
<i>hReceiver</i>	AMHRCV	input
<i>hRcvMsg</i>	AMHMSG	input
<i>dataLen</i>	AMLONG	input
<i>Data</i> (1)	AMBYTE	input
<i>hSndMsg</i>	AMHMSG	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amSndSendFile

Purpose: Sends a file to the destination specified by the sender service.

Table 285. amSndSendFile call

Parameter	Data type	Usage
<i>hSender</i>	AMHSND	input
<i>hPolicy</i>	AMHPOL	input
<i>options</i>	AMLONG	input
<i>directoryLen</i>	AMLONG	input
<i>directory</i> (1)	AMSTR	input
<i>fileNameLen</i>	AMLONG	input
<i>FileName</i> (1)	AMSTR	input
<i>hSndMsg</i>	AMHMSG	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

Subscriber interface functions

Subscriber interface functions

amSubClearErrorCodes

Purpose: Clears the error codes in the subscriber object.

Table 286. amSubClearErrorCodes call

Parameter	Data type	Usage
<i>hSubscriber</i>	AMHSUB	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note: 1. In the C programming language this parameter is a pointer.		

amSubClose

Purpose: Closes the subscriber service.

Table 287. amSubClose call

Parameter	Data type	Usage
<i>hSubscriber</i>	AMHSUB	input
<i>hPolicy</i>	AMHPOL	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note: 1. In the C programming language this parameter is a pointer.		

amSubGetCCSID

Purpose: Gets the coded character set identifier to be used for the subscriber's sender service.

Table 288. amSubGetCCSID call

Parameter	Data type	Usage
<i>hSubscriber</i>	AMHSUB	input
<i>CCSID</i> (1)	AMLONG	output
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note: 1. In the C programming language this parameter is a pointer.		

amSubGetDefnType

Purpose: Gets the definition type of the subscriber's receiver service.

Table 289. amSubGetDefnType call

Parameter	Data type	Usage
<i>hSubscriber</i>	AMHSUB	input
<i>Type</i> (1)	AMLONG	output
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amSubGetEncoding

Purpose: Gets the value to be used to encode numeric data types for the subscriber's sender service.

Table 290. amSubGetEncoding call

Parameter	Data type	Usage
<i>hSubscriber</i>	AMHSUB	input
<i>Encoding</i> (1)	AMLONG	output
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amSubGetLastError

Purpose: Gets the information (completion and reason codes) from the last error for the subscriber object.

amSubGetLastError

Table 291. *amSubGetLastError* call

Parameter	Data type	Usage
<i>hSubscriber</i>	AMHSUB	input
<i>buffLen</i>	AMLONG	input
<i>StringLen</i> (1)	AMLONG	output
<i>ErrorText</i> (1)	AMSTR	output
<i>Reason2</i> (1)	AMLONG	output
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note: 1. In the C programming language this parameter is a pointer.		

amSubGetName

Purpose: Gets the name of the subscriber object.

Table 292. *amSubGetName* call

Parameter	Data type	Usage
<i>hSubscriber</i>	AMHSUB	input
<i>buffLen</i>	AMLONG	input
<i>NameLen</i> (1)	AMLONG	output
<i>Name</i> (1)	AMSTR	output
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note: 1. In the C programming language this parameter is a pointer.		

amSubGetQueueName

Purpose: Gets the queue name of the subscriber's receiver service object.

Table 293. amSubGetQueueName call

Parameter	Data type	Usage
<i>hSubscriber</i>	AMHSUB	input
<i>buffLen</i>	AMLONG	input
<i>StringLen</i> (1)	AMLONG	output
<i>QueueName</i> (1)	AMSTR	output
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amSubOpen

Purpose: Opens the subscriber service.

Table 294. amSubOpen call

Parameter	Data type	Usage
<i>hSubscriber</i>	AMHSUB	input
<i>hPolicy</i>	AMHPOL	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amSubReceive

Purpose: Receives a message, normally a publication, using the subscriber service.

amSubReceive

Table 295. *amSubReceive* call

Parameter	Data type	Usage
<i>hSubscriber</i>	AMHSUB	input
<i>hPolicy</i>	AMHPOL	input
<i>hSelMsg</i>	AMHMSG	input
<i>hRcvMsg</i>	AMHMSG	output
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note: 1. In the C programming language this parameter is a pointer.		

amSubSetQueueName

Purpose: Sets the queue name of the subscriber's receiver object, when this encapsulates a model queue.

Table 296. *amSubSetQueueName* call

Parameter	Data type	Usage
<i>hSubscriber</i>	AMHSUB	input
<i>Name</i>	AMLONG	input
<i>QueueName</i> (1)	AMSTR	input
<i>CompCode</i> (1)	AMLONG	output
<i>Reason</i> (1)	AMLONG	output
Note: 1. In the C programming language this parameter is a pointer.		

amSubSubscribe

Purpose: Sends a subscribe message to a publish/subscribe broker using the subscriber service, to register a subscription.

Table 297. amSubSubscribe call

Parameter	Data type	Usage
<i>hSubscriber</i>	AMHSUB	input
<i>hPolicy</i>	AMHPOL	input
<i>hResponse</i>	AMHRCV	input
<i>hSubMsg</i>	AMHMSG	input
<i>CompCode (1)</i>	AMLONG	output
<i>Reason (1)</i>	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amSubUnsubscribe

Purpose: Sends an unsubscribe message to a publish/subscribe broker using the subscriber service, to deregister a subscription.

Table 298. amSubUnsubscribe call

Parameter	Data type	Usage
<i>hSubscriber</i>	AMHSUB	input
<i>hPolicy</i>	AMHPOL	input
<i>hResponse</i>	AMHRCV	input
<i>hUnsubMsg</i>	AMHMSG	input
<i>CompCode (1)</i>	AMLONG	output
<i>Reason (1)</i>	AMLONG	output
Note:		
1. In the C programming language this parameter is a pointer.		

amSubUnsubscribe

Chapter 9. Elementary data types

MQI calls use both elementary and structure data types. These correspond to data types that can be declared in a language that supports user-defined data types (for example, the C programming language). However, all user-defined data types ultimately resolve to elementary data types, or to aggregates of elementary data types (known as arrays or structures).

The elementary data types are:

MQBYTE	Byte
MQBYTE16	String of 16 bytes
MQBYTE24	String of 24 bytes
MQBYTE32	String of 32 bytes
MQBYTE40	String of 40 bytes
MQBYTE64	String of 64 bytes
MQCHAR	Single-byte character
MQCHARn	String of n single-byte characters
MQHBAG	Bag handle
MQHCONFIG	Configuration handle
MQHCONN	Connection handle
MQHOBJ	Object handle
MLONG	Long integer
MQPTR	Pointer
PMQCHAR	Pointer to character
PMQFUNC	Pointer to function
PMQLONG	Pointer to data of type MLONG
PMQMD	Pointer to MQMD structure
PMQVOID	Pointer to void
PPMQWDR	Pointer to PMQWDR structure
PPMQWQR	Pointer to PMQWQR structure

All other data types equate either directly to these elementary data types or to aggregates of them (that is, arrays or structures).

In C language, the parameters of some of the MQI calls are defined as being pointers to the relevant data type or structure.

Elementary data types - C

Elementary data types - C

These are typical declarations of the data types in C:

Table 299. C declarations of data types

Data type	Representation
MQBYTE	<code>typedef unsigned char MQBYTE;</code>
MQBYTE16	<code>typedef MQBYTE MQBYTE16[16];</code>
MQBYTE24	<code>typedef MQBYTE MQBYTE24[24];</code>
MQBYTE32	<code>typedef MQBYTE MQBYTE32[32];</code>
MQBYTE40	<code>typedef MQBYTE MQBYTE40[40];</code>
MQBYTE64	<code>typedef MQBYTE MQBYTE64[64];</code>
MQCHAR	<code>typedef char MQCHAR;</code>
MQCHAR4	<code>typedef MQCHAR MQCHAR4[4];</code>
MQCHAR8	<code>typedef MQCHAR MQCHAR8[8];</code>
MQCHAR12	<code>typedef MQCHAR MQCHAR12[12];</code>
MQCHAR16	<code>typedef MQCHAR MQCHAR16[16];</code>
MQCHAR20	<code>typedef MQCHAR MQCHAR20[20];</code>
MQCHAR28	<code>typedef MQCHAR MQCHAR28[28];</code>
MQCHAR32	<code>typedef MQCHAR MQCHAR32[32];</code>
MQCHAR48	<code>typedef MQCHAR MQCHAR48[48];</code>
MQCHAR64	<code>typedef MQCHAR MQCHAR64[64];</code>
MQCHAR128	<code>typedef MQCHAR MQCHAR128[128];</code>
MQCHAR256	<code>typedef MQCHAR MQCHAR256[256];</code>
MQHBAG	<code>typedef MQLONG MQHBAG;</code>
MQHCONFIG	<code>typedef MQLONG MQHCONFIG;</code>
MQHCONN	<code>typedef MQLONG MQHCONN;</code>
MQHOBJ	<code>typedef MQLONG MQHOBJ;</code>
MQLONG	<code>typedef long MQLONG;</code>
MQPTR	<code>typedef void MQPOINTER MQPTR;</code>
PMQCHAR	<code>typedef void MQPOINTER PMQCHAR;</code>

Table 299. C declarations of data types (continued)

Data type	Representation
PMQFUNC	typedef void MQPOINTER PMQFUNC;
PMQLONG	typedef MQLONG MQPOINTER PMQLONG;
PMQMD	typedef MQMD MQPOINTER PMQMD;
PMQVOID	typedef void MQPOINTER PMQVOID;
PPMQWDR	typedef PMQWDR MQPOINTER PPMQWDR;
PPMQWQR	typedef PMQWDR MQPOINTER PPMQWQR;

Elementary data types - COBOL

These are typical declarations of the data types in COBOL:

Table 300. COBOL declarations of data types

Data type	Representation
MQBYTE	PIC X
MQBYTE16	PIC X(16)
MQBYTE24	PIC X(24)
MQBYTE32	PIC X(32)
MQBYTE40	PIC X(40)
MQBYTE64	PIC X(64)
MQCHAR	PIC X
MQCHAR4	PIC X(4)
MQCHAR8	PIC X(8)
MQCHAR12	PIC X(12)
MQCHAR16	PIC X(16)
MQCHAR20	PIC X(20)
MQCHAR28	PIC X(28)
MQCHAR32	PIC X(32)
MQCHAR48	PIC X(48)
MQCHAR64	PIC X(64)

Elementary data types - COBOL

Table 300. COBOL declarations of data types (continued)

Data type	Representation
MQCHAR128	PIC X(128)
MQCHAR256	PIC X(256)
MQHCONN	PIC S9(9) BINARY
MQHOBJ	PIC S9(9) BINARY
MQLONG	PIC S9(9) BINARY
MQPTR	POINTER
PMQLONG	POINTER

Elementary data types - PL/I

These are typical declarations of the data types in PL/I:

Table 301. PL/I declarations of data types

Data type	Representation
MQBYTE	char(1)
MQBYTE16	char(16)
MQBYTE24	char(24)
MQBYTE32	char(32)
MQBYTE40	char(40)
MQBYTE64	char(64)
MQCHAR	char(1)
MQCHAR4	char(4)
MQCHAR8	char(8)
MQCHAR12	char(12)
MQCHAR16	char(16)
MQCHAR20	char(20)
MQCHAR28	char(28)
MQCHAR32	char(32)
MQCHAR48	char(48)

Table 301. PL/I declarations of data types (continued)

Data type	Representation
MQCHAR64	char(64)
MQCHAR128	char(128)
MQCHAR256	char(256)
MQHCONN	fixed bin(31)
MQHOBJ	fixed bin(31)
MQLONG	fixed bin(31)
PMQLONG	pointer

Elementary data types - System/390[®] assembler

These are typical declarations of the data types in System/390 assembler.

Table 302. S/390[®] assembler declarations of data types

Data type	Representation
MQBYTE	DS XL1
MQBYTE16	DS XL16
MQBYTE24	DS XL24
MQBYTE32	DS XL32
MQBYTE40	DS XL40
MQBYTE64	DS XL64
MQCHAR	DS CL1
MQCHAR4	DS CL4
MQCHAR8	DS CL8
MQCHAR12	DS CL12
MQCHAR16	DS CL16
MQCHAR20	DS CL20
MQCHAR28	DS CL28
MQCHAR32	DS CL32
MQCHAR48	DS CL48

Elementary data types - S/390 assembler

Table 302. S/390[®] assembler declarations of data types (continued)

Data type	Representation
MQCHAR64	DS CL64
MQCHAR128	DS CL128
MQCHAR256	DS CL256
MQHCONN	DS F
MQHOBJ	DS F
MQLONG	DS F
PMQLONG	DS F

Elementary data types - TAL

The elementary data types for the TAL programming language are:

Table 303. TAL declarations of data types

Data Type	Representation
MQBYTE	STRING
MQBYTE24	BEGIN STRING BYTE [0:23];END
MQBYTE32	BEGIN STRING BYTE [0:31];END
MQCHAR	STRING
MQCHAR4	BEGIN STRING BYTE [0:3];END
MQCHAR8	BEGIN STRING BYTE [0:7]; END
MQCHAR12	BEGIN STRING BYTE [0:11];END
MQCHAR28	BEGIN STRING BYTE [0:27];END
MQCHAR32	BEGIN STRING BYTE [0:31];END
MQCHAR48	BEGIN STRING BYTE [0:47];END
MQCHAR64	BEGIN STRING BYTE [0:63];END
MQCHAR128	BEGIN STRING BYTE [0:127];END
MQCHAR256	BEGIN STRING BYTE [0:255];END
MQHCONN	INT(32)
MQHOBJ	INT(32)

Table 303. TAL declarations of data types (continued)

Data Type	Representation
MQLONG	INT(32)

Elementary data types - Visual Basic (Windows 95, Windows 98, and Windows NT)

The elementary data types for Visual Basic are:

Table 304. Visual Basic declarations of data types

Data Type	Representation
MQBYTE	String*1
MQBYTE24	String*24
MQBYTE32	String*32
MQCHAR	String*1
MQCHAR4	String*4
MQCHAR8	String*8
MQCHAR12	String*12
MQCHAR28	String*28
MQCHAR32	String*32
MQCHAR48	String*48
MQCHAR64	String*64
MQCHAR128	String*128
MQCHAR256	String*256
MQHCONN	Long
MQHOBJ	Long
MQLONG	Long

Elementary data types - Visual Basic

Chapter 10. Structure data types

The structure data types used in the programming interface are:

MQBO	Begin options
MQCD	Channel data
MQCFH	PCF header
MQCFIN	PCF integer parameter
MQCFST	PCF string parameter
MQCFIL	PCF integer list parameter
MQCFSL	PCF string list parameter
MQCIH	CICS® information header
MQCNO	Connect options
MQCXP	Channel exit parameter
MQDH	Distribution header
MQDLH	Dead-letter header
MQDXP	Data-conversion exit
MQGMO	Get-message options
MQIIH	IMS™ information header
MQMD	Message descriptor
MQMDE	Message descriptor extension
MQOD	Object descriptor
MQOR	Object record
MQPMO	Put-message options
MQPMR	Put-message record
MQRMH	Reference message header
MQRR	Response record
MQTM	Trigger message
MQTMC	Trigger message (character format)
MQTMC2	Trigger message (character format 2)
MQTXP	Transport exit parameter
MQWCR	Cluster workload cluster-record
MQWDR	Cluster workload destination-record
MQWIH	Work information header
MQWQR	Cluster workload queue-record
MQWXP	Cluster workload exit parameter
MQXP	API-crossing exit parameter
MQXQH	Transmission queue header
MQXWD	Exit wait descriptor
MQZED	Entity data structure

In the following tables, the initial values of the fields are shown.

MQBO

MQBO (begin options)

Table 305. MQBO structure

Field name	Data type	Name of constant	Value of constant
<i>StrucId</i>	MQCHAR4	MQBO_STRUC_ID	'B0bb'
<i>Version</i>	MLONG	MQBO_VERSION_1	1
<i>Options</i>	MLONG	MQBO_NONE	0

Note: This structure is supported on Version 5, or later, products only.

MQCD (Channel data structure)

Table 306. MQCD structure

Field name	Data type	Name of constant	Value of constant
<i>ChannelName</i>	MQCHAR20	MQ_CHANNEL_NAME_LENGTH	
<i>Version</i>	MLONG	MQCD_VERSION_1	1
<i>ChannelType</i>	MLONG		0
<i>TransportType</i>	MLONG		
<i>Desc</i>	MQCHAR64	MQ_CHANNEL_DESC_LENGTH	
<i>QMgrName</i>	MQCHAR48	MQ_Q_MGR_NAME_LENGTH	
<i>XmitQName</i>	MQCHAR48	MQ_Q_NAME_LENGTH	
<i>ShortConnectionName</i>	MQCHAR20		
<i>MCAName</i>	MQCHAR20	Reserved	'bbbbbbbb'
<i>ModeName</i>	MQCHAR8		
<i>TpName</i>	MQCHAR64		
<i>BatchSize</i>	MLONG		
<i>DiscInterval</i>	MLONG		
<i>ShortRetryCount</i>	MLONG	none	0
<i>ShortRetryInterval</i>	MLONG	none	0
<i>LongRetryCount</i>	MLONG	none	0
<i>LongRetryInterval</i>	MLONG	none	0
<i>SecurityExit</i>	MQCHARn	none	blanks
<i>MsgExit</i>	MQCHARn	none	blanks
<i>SendExit</i>	MQCHARn	none	blanks
<i>ReceiveExit</i>	MQCHARn	none	blanks

Table 306. MQCD structure (continued)

Field name	Data type	Name of constant	Value of constant
<i>SeqNumberWrap</i>	MQLONG	none	blanks
<i>MaxMsgLength</i>	MQLONG	none	blanks
<i>PutAuthority</i>	MQLONG		
<i>DataConversion</i>	MQLONG		
<i>SecurityUserData</i>	MQCHAR32		
<i>MsgUserData</i>	MQCHAR32		
<i>SendUserData</i>	MQCHAR32		
<i>ReceiveUserData</i>	MQCHAR32		
<i>UserIdentifier</i> (1)	MQCHAR12		
<i>Password</i> (1)	MQCHAR12	none	blanks
<i>MCAUserIdentifier</i> (1)	MQCHAR12	none	blanks
<i>MCAType</i> (1)	MQLONG	none	blanks
<i>ConnectionName</i> (1)	MQCHAR264		
<i>RemoteUserIdentifier</i> (1)	MQCHAR12	none	
<i>RemotePassword</i> (1)	MQCHAR12	none	
<i>MsgRetryExit</i> (2)	MQCHARn		
<i>MsgRetryUserData</i> (2)	MQCHAR32		
<i>MsgRetryCount</i> (2)	MQLONG		
<i>MsgRetryInterval</i> (2)	MQLONG		
<i>HeartbeatInterval</i> (3)	MQLONG		
<i>BatchInterval</i> (3)	MQLONG		
<i>NonPersistentMsgSpeed</i> (3)	MQLONG	MQNPMS_NORMAL	
<i>StrucLength</i> (3)	MQLONG	MQCD_CURRENT_LENGTH	
<i>ExitNameLength</i> (3)	MQLONG	MQCD_CURRENT_LENGTH	
<i>ExitDataLength</i> (3)	MQLONG		
<i>MsgExitsDefined</i> (3)	MQLONG	0	
<i>SendExitsDefined</i> (3)	MQLONG	0	
<i>ReceiveExitsDefined</i> (3)	MQLONG	0	
<i>MsgExitPtr</i> (3)	MQPTR	0	
<i>MsgUserDataPtr</i> (3)	MQPTR	0	
<i>SendExitPtr</i> (3)	MQPTR	0	
<i>SendUserDataPtr</i> (3)	MQPTR	0	

MQCD

Table 306. MQCD structure (continued)

Field name	Data type	Name of constant	Value of constant
<i>ReceiveExitPtr</i> (3)	MQPTR	0	
<i>ReceiveUserDataPtr</i> (3)	MQPTR	0	
<i>ClusterPtr</i> (4)	MQPTR	0	
<i>ClustersDefined</i> (4)	MQPTR	0	
<i>NetworkPriority</i> (4)	MQLONG	0	
<i>LongMCAUserIdLength</i> (5)	MQLONG	0	
<i>LongRemoteUserIdLength</i> (5)	MQPTR	0	
<i>LongMCAUserIdPtr</i> (5)	MQPTR	0	
<i>LongRemoteUserIdPtr</i> (5)	MQPTR	0	
<i>MCASecurityId</i> (5)	MQPTR	0	
<i>Remote SecurityId</i> (5)	MQPTR	0	

Notes:

1. This field is present only when *Version* is MQCD_VERSION_2 or later.
2. This field is present only when *Version* is MQCD_VERSION_3 or later.
3. This field is present only when *Version* is MQCD_VERSION_4 or later.
4. This field is present only when *Version* is MQCD_VERSION_5 or later.
5. This field is present only when *Version* is MQCD_VERSION_6 or later.

MQCFH (PCF header)

Table 307. MQCFH structure

Field name	Data type	Name of constant	Value of constant
<i>Type</i>	MQLONG	MQCFT_COMMAND	1
<i>StrucLength</i>	MQLONG	MQCFH_STRUC_LENGTH	36
<i>Version</i>	MQLONG	MQCFH_VERSION_1	1
<i>Command</i>	MQLONG	None	0
<i>MsgSeqNumber</i>	MQLONG	None	1
<i>Control</i>	MQLONG	MQCFC_LAST	1
<i>CompCode</i>	MQLONG	MQCC_OK	0
<i>Reason</i>	MQLONG	MQRC_NONE	0
<i>ParameterCount</i>	MQLONG	None	0

MQCFIL (PCF integer list parameter)

Table 308. MQCFIL structure

Field name	Data type	Name of constant	Value of constant
<i>Type</i>	MQLONG	MQCFT_INTEGER_LIST	5
<i>StrucLength</i>	MQLONG	MQCFIL_STRUC_LENGTH_FIXED	16
<i>Parameter</i>	MQLONG	None	0
<i>Count</i>	MQLONG	None	0
<i>Values (1)</i>	MQLONG x Count	None	0

Note:

1. This parameter is present in the C programming language only.

MQCFIN (PCF integer parameter)

Table 309. MQCFIN structure

Field name	Data type	Name of constant	Value of constant
<i>Type</i>	MQLONG	MQCFT_INTEGER	3
<i>StrucLength</i>	MQLONG	MQCFIN_STRUC_LENGTH	16
<i>Parameter</i>	MQLONG	None	0

MQCFSL

MQCFSL (PCF string list parameter)

Table 310. MQCFSL structure

Field name	Data type	Name of constant	Value of constant
Type	MQLONG	MQCFT_STRING_LIST	6
StrucLength	MQLONG	MQCFSL_STRUC_LENGTH_FIXED	24
Parameter	MQLONG	None	0
CodedCharSetId	MQLONG	MQCCSI_DEFAULT	0
Count	MQLONG	None	0
StringLength	MQLONG	None	0
Strings (1)	MQCHAR x StringLength x Count	None	Null string

Note:

1. This parameter is present in the C programming language only.

MQCFST (PCF string parameter)

Table 311. MQCFST structure

Field name	Data type	Name of constant	Value of constant
Type	MQLONG	MQCFT_STRING	4
StrucLength	MQLONG	MQCFST_STRUC_LENGTH_FIXED	20
Parameter	MQLONG	None	0
CodedCharSetId	MQLONG	MQCCSI_DEFAULT	0
StringLength	MQLONG	None	0
Buffer (1)	MQCHAR x StringLength	None	Null string

Note:

1. This parameter is present in the C programming language only.

MQCIH (CICS bridge header)

Table 312. MQCIH structure

Field name	Data type	Name of constant	Value of constant
StrucId	MQCHAR4	MQCIH_STRUC_ID	'CIHb'

Table 312. MQCIH structure (continued)

Field name	Data type	Name of constant	Value of constant
<i>Version</i>	MLONG	MQCIH_VERSION_1	1
<i>StrucLength</i>	MLONG	MQCIH_LENGTH_1	164
<i>Encoding</i>	MLONG	none	0
<i>CodedCharSetId</i>	MLONG	none	0
<i>Format</i>	MQCHAR8	MQFMT_NONE	'bbbbbbbb'
<i>Flags</i>	MLONG	MQCIH_NONE	0
<i>ReturnCode</i>	MLONG	MQCRC_OK	0
<i>CompCode</i>	MLONG	MQCC_OK	0
<i>Reason</i>	MLONG	MQRC_NONE	0
<i>UOWControl</i>	MLONG	MQCUOWC_ONLY	273
<i>GetWaitInterval</i>	MLONG	MQCGWI_DEFAULT	-2
<i>LinkType</i>	MLONG	MQCLT_PROGRAM	1
<i>OutputDataLength</i>	MLONG	MQCODL_AS_INPUT	-1
<i>FacilityKeepTime</i>	MLONG	none	0
<i>ADSDDescriptor</i>	MLONG	MQCADSD_NONE	0
<i>ConversationalTask</i>	MLONG	MQCCT_NO	0
<i>TaskEndStatus</i>	MLONG	MQCTES_NOSYNC	0
<i>Facility</i>	MQBYTE8	MQCFAC_NONE	Nulls
<i>Function</i>	MQCHAR4	MQCFUNC_NONE	'bbbb'
<i>AbendCode</i>	MQCHAR4	none	'bbbb'
<i>Authenticator</i>	MQCHAR8	none	'bbbbbbbb'
<i>Reserved1</i>	MQCHAR8	none	'bbbbbbbb'
<i>ReplyToFormat</i>	MQCHAR8	MQFMT_NONE	'bbbbbbbb'
<i>RemoteSysID</i>	MQCHAR4	none	'bbbb'
<i>RemoteTransID</i>	MQCHAR4	none	'bbbb'
<i>TransactionID</i>	MQCHAR4	none	'bbbb'
<i>FacilityLike</i>	MQCHAR4	none	'bbbb'
<i>AttentionID</i>	MQCHAR4	none	'bbbb'
<i>StartCode</i>	MQCHAR4	MQCSC_NONE	'bbbb'
<i>CancelCode</i>	MQCHAR4	none	'bbbb'
<i>NextTransactionID</i>	MQCHAR4	none	'bbbb'
<i>Reserved2</i>	MQCHAR8	none	'bbbbbbbb'

MQCIH

Table 312. MQCIH structure (continued)

Field name	Data type	Name of constant	Value of constant
<i>Reserved3</i>	MQCHAR8	none	'bbbbbbbb'
<i>CursorPosition</i>	MLONG	none	0
<i>ErrorOffset</i>	MLONG	none	0
<i>InputItem</i>	MLONG	none	0
<i>Reserved4</i>	MLONG	none	0

Note:

1. This structure is supported on Version 5.1 products only.

MQCNO (connect options)

Table 313. MQCNO structure

Field name	Data type	Name of constant	Value of constant
<i>StrucId</i>	MQCHAR4	MQCNO_STRUC_ID	'CNOb'
<i>Version</i>	MLONG	MQCNO_VERSION_1	1
<i>Options</i>	MLONG	MQCNO_NONE	0
<i>ClientConnOffset</i> (1)	MLONG	none	0
<i>ClientConnPtr</i> (1)	MQPTR	none	nulls
<i>ConnTag</i> (2)	MQBYTE128	MQCT_NONE	nulls

Notes::

1. This field is used on Version 5.1 products only.
2. This field is used on MQSeries for OS/390 V5.2 only.

MQCXP (Channel exit parameter structure)

Table 314. MQCXP structure

Field name	Data type	Name of constant	Value of constant
<i>StrucId</i>	MQCHAR4	MQCXP_STRUC_ID	CXPb
<i>Version</i>	MLONG	MQCXP_VERSION_1	1
<i>ExitId</i>	MLONG		0
<i>ExitReason</i>	MLONG		
<i>ExitResponse</i>	MLONG		

Table 314. MQCXP structure (continued)

Field name	Data type	Name of constant	Value of constant
<i>ExitResponse2</i>	MQLONG		
<i>Feedback</i>	MQLONG		
<i>MaxSegmentLength</i>	MQLONG		
<i>ExitUserArea</i>	MQBYTE16		
<i>ExitData</i>	MQCHAR32		
<i>MsgRetryCount</i> (1)	MQLONG	none	blanks
<i>MsgRetryInterval</i> (1)	MQLONG	none	0
<i>MsgRetryReason</i> (1)	MQLONG	none	0
<i>HeaderLength</i> (2)	MQLONG	none	0
<i>PartnerName</i> (2)	MQCHAR48		
<i>FAPLevel</i> (2)	MQLONG		
<i>CapabilityLevel</i> (2)	MQLONG		
<i>ExitNumber</i> (2)	MQLONG		

Notes:

1. This field is present only when *Version* is MQCXP_VERSION_2 or later.
2. This field is present only when *Version* is MQCXP_VERSION_3 or later.

MQDH

MQDH (distribution header)

Table 315. MQDH structure

Field name	Data type	Name of constant	Value of constant
<i>StrucId</i>	MQCHAR4	MQDH_STRUC_ID	'DHbb'
<i>Version</i>	MQLONG	MQDH_VERSION_1	1
<i>Struclength</i>	MQLONG	none	0
<i>Encoding</i>	MQLONG	none	0
<i>CodedCharSetId</i>	MQLONG	none	0
<i>Format</i>	MQCHAR8	MQFMT_NONE	'bbbbbbbb'
<i>Flags</i>	MQLONG	MQDHF_NONE	0
<i>PutMsgRecFields</i>	MQLONG	MQPMRF_NONE	0
<i>RecsPresent</i>	MQLONG	none	0
<i>ObjectRecOffset</i>	MQLONG	none	0
<i>PutMsgRecOffset</i>	MQLONG	none	0

Note: This structure is supported on Version 5, or later, products.

MQDLH (dead-letter header)

Table 316. MQDLH structure

Field name	Data type	Name of constant	Value of constant
<i>StrucId</i>	MQCHAR4	MQDLH_STRUC_ID	'DLHb'
<i>Version</i>	MQLONG	MQDLH_VERSION_1	1
<i>Reason</i>	MQLONG	MQRC_NONE	0
<i>DestQName</i>	MQCHAR48	none	blanks
<i>DestQMgrName</i>	MQCHAR48	none	blanks
<i>Encoding</i>	MQLONG	MQENC_NATIVE AS/400: OS/2(C): OS/2(COBOL): OS/390: UNIX systems: Windows NT(C): Windows NT(COBOL):	 273 546 17 785 273 546 17
<i>CodedCharSetId</i>	MQLONG	none	0
<i>Format</i>	MQCHAR8	MQFMT_NONE	'bbbbbbbb'

Table 316. MQDLH structure (continued)

Field name	Data type	Name of constant	Value of constant
<i>PutApplType</i>	MQLONG	none	0
<i>PutApplName</i>	MQCHAR28	none	blanks
<i>PutDate</i>	MQCHAR8	none	blanks
<i>PutTime</i>	MQCHAR8	none	blanks

Note: This structure is not supported on Windows.

MQDXP (data-conversion exit parameter)

Table 317. MQDXP structure

Field name	Data type	Name of constant	Value of constant
<i>StrucId</i>	MQCHAR4	MQDXP_STRUC_ID	'DXPb'
<i>Version</i>	MQLONG	MQDXP_VERSION_1	1
<i>ExitOptions</i>	MQLONG	none	0
<i>AddOptions</i>	MQLONG	none	0
<i>Encoding</i>	MQLONG	none	0
<i>CodedCharSetId</i>	MQLONG	none	0
<i>DataLength</i>	MQLONG	none	0
<i>CompCode</i>	MQLONG	MQCC_OK	0
<i>Reason</i>	MQLONG	MQRC_NONE	0
<i>ExitResponse</i>	MQLONG	MQXDR_OK	0
<i>Hconn</i>	MQHCONN	none	0

MQGMO

MQGMO (get-message options)

Table 318. MQGMO structure

Field name	Data type	Name of constant	Value of constant
<i>StrucId</i>	MQCHAR4	MQGMO_STRUC_ID	'GMOb'
<i>Version</i>	MLONG	MQGMO_VERSION_1	1
<i>Options</i>	MLONG	MQGMO_NO_WAIT	0
<i>WaitInterval</i>	MLONG	none	0
<i>Signal1</i>	OS/390: PMQLONG All others: MLONG	none	NULL 0
<i>Signal2</i>	MLONG	none	0
<i>ResolvedQName</i>	MQCHAR48	none	blanks
<i>MatchOptions</i> (1)	MLONG	MQMO_MATCH_MSG_ID MQMO_MATCH_CORREL_ID	3
<i>GroupStatus</i> (1)	MQCHAR	MQGS_NOT_IN_GROUP	blanks
<i>SegmentStatus</i> (1)	MQCHAR	MQSS_NOT_A_SEGMENT	blanks
<i>Segmentation</i> (1)	MQCHAR	MQSEG_INHIBITED	blanks
<i>Reserved1</i> (1)	MQCHAR	none	blanks
<i>MsgToken</i> (2)	MQBYTE16	MQMTOK_NONE	NULL
<i>ReturnedLength</i> (1)	MQBYTE16	MQRL_UNDEFINED	-1

Note:

1. This field is used on Version 5.1 products, and OS/390 only.
2. This field is used on OS/390 only.

MQIIH (IMS information header)

Table 319. MQIIH structure

Field name	Data type	Name of constant	Value of constant
<i>StrucId</i>	MQCHAR4	MQIIH_STRUC_ID	'IIHb'
<i>Version</i>	MQLONG	MQIIH_VERSION_1	1
<i>StrucLength</i>	MQLONG	MQIIH_LENGTH_1	84
<i>Encoding</i>	MQLONG	MQENC_NATIVE OS/390: OS/2(C): OS/2(COBOL): AS/400: UNIX [®] systems: Windows NT(C): Windows NT(COBOL):	785 546 17 273 273 546 17
<i>CodedCharSetId</i>	MQLONG	MQCCSI_Q_MGR	0
<i>Format</i>	MQCHAR8	MQFMT_NONE	'bbbbbbbb'
<i>Flags</i>	MQLONG	MQIIH_NONE	0
<i>LTermOverride</i>	MQCHAR8	none	blanks
<i>MFSMapName</i>	MQCHAR8	none	blanks
<i>ReplyToFormat</i>	MQCHAR8	MQFMT_NONE	'bbbbbbbb'
<i>Authenticator</i>	MQCHAR8	MQIAUT_NONE	'bbbbbbbb'
<i>TranInstanceId.</i>	MQBYTE16	MQITIL_NONE	nulls
<i>TranState</i>	MQCHAR	MQITS_NOT_IN_CONVERSATION	'b'
<i>CommitMode</i>	MQCHAR	MQICM_COMMIT_THEN_SEND	'0'
<i>SecurityScope</i>	MQCHAR	MQISS_CHECK	'c'
<i>Reserved</i>	MQCHAR	none	'b'

Note: This structure is not supported on Windows, Tandem NonStop Kernel, or VSE/ESA.

MQMD (message descriptor)

Table 320. MQMD structure

Field name	Data type	Name of constant	Value of constant
<i>StrucId</i>	MQCHAR4	MQMD_STRUC_ID	'MDbb'
<i>Version</i>	MQLONG	MQMD_VERSION_1	1
<i>Report</i>	MQLONG	MQRO_NONE	0

MQMD

Table 320. MQMD structure (continued)

Field name	Data type	Name of constant	Value of constant
<i>MsgType</i>	MQLONG	MQMT_DATAGRAM	8
<i>Expiry</i>	MQLONG	MQEI_UNLIMITED	-1
<i>Feedback</i>	MQLONG	MQFB_NONE	0
<i>Encoding</i>	MQLONG	MQENC_NATIVE OS/2(C): OS/2(COBOL): OS/390: AS/400: UNIX systems: Windows NT(C): Windows NT(COBOL):	785 546 17 273 273 546 17
<i>CodedCharSetId</i>	MQLONG	MQCCSI_Q_MGR	0
<i>Format</i>	MQCHAR8	MQFMT_NONE	'bbbbbbbb'
<i>Priority</i>	MQLONG	MQPRI_PRIORITY_AS_Q_DEF	-1
<i>Persistence</i>	MQLONG	MQPER_PERSISTENCE_AS_Q_DEF	2
<i>MsgId</i>	MQBYTE24	MQMI_NONE	nulls
<i>CoreId</i>	MQBYTE24	MQCI_NONE	nulls
<i>BackoutCount</i>	MQLONG	none	0
<i>ReplyToQ</i>	MQCHAR48	none	blanks
<i>ReplyToQMgr</i>	MQCHAR48	none	blanks
<i>UserIdentifier</i>	MQCHAR12	none	blanks
<i>AccountingToken</i>	MQBYTE32	MQACT_NONE	nulls
<i>ApplIdentityData</i>	MQCHAR32	none	blanks
<i>PutApplType</i>	MQLONG	MQAT_NO_CONTEXT	0
<i>PutApplName</i>	MQCHAR28	none	blanks
<i>PutDate</i>	MQCHAR8	none	blanks
<i>PutTime</i>	MQCHAR8	none	blanks
<i>ApplOriginData</i>	MQCHAR4	none	blanks
<i>GroupId</i> (1)	MQBYTE24	MQGI_NONE	nulls
<i>MsgSeqNumber</i> (1)	MQLONG	none	1
<i>Offset</i> (1)	MQLONG	none	0
<i>Msgflags</i> (1)	MQLONG	MQMF_NONE	0
<i>OriginalLength</i> (1)	MQLONG	MQOL_UNDEFINED	-1

Table 320. MQMD structure (continued)

Field name	Data type	Name of constant	Value of constant
Note:			
1. This field is used on Version 5 or later products; that is, only when MQMD_VERSION_2 is selected.			

MQMDE (message descriptor extension)

Table 321. MQMDE structure

Field name	Data type	Name of constant	Value of constant
<i>StrucId</i>	MQCHAR4	MQMDE_STRUC_ID	'MDEb'
<i>Version</i>	MQLONG	MQMDE_VERSION_2	2
<i>StrucLength</i>	MQLONG	MQMDE_STRUC_LENGTH_2	72
<i>Encoding</i>	MQLONG	MQENC_NATIVE OS/390: OS/2(C): OS/2(COBOL): AS/400: UNIX systems: Windows NT(C): Windows NT(COBOL):	785 546 17 273 273 546 17
<i>CodedCharSetId</i>	MQLONG	MQCCSI_Q_MGR	0
<i>Format</i>	MQCHAR8	MQFMT_NONE	'bbbbbbbb'
<i>Flags</i>	MQLONG	MQMDEF_NONE	0
<i>GroupId</i>	MQBYTE24	MQGI_NONE	nulls
<i>MsgSeqNumber</i>	MQLONG	none	1
<i>Offset</i>	MQLONG	none	0
<i>MsgFlags</i>	MQLONG	MQMF_NONE	0
<i>OriginalLength</i>	MQLONG	MQOL_UNDEFINED	-1
Note: This structure is supported on Version 5 or later products.			

MQOD

MQOD (object descriptor)

Table 322. MQOD structure

Field name	Data type	Name of constant	Value of constant
<i>StrucId</i>	MQCHAR4	MQOD_STRUC_ID	'0Dbb'
<i>Version</i>	MQLONG	MQOD_VERSION_1	1
<i>ObjectType</i>	MQLONG	MQOT_Q	1
<i>ObjectName</i>	MQCHAR48	none	blanks
<i>ObjectQMgrName</i>	MQCHAR48	none	blanks
<i>DynamicQName</i>	MQCHAR48	none OS/390: none All others:	'CSQ.*' 'AMQ.*'
<i>AlternateUserId</i>	MQCHAR12	none	blanks
<i>RecsPresent</i> (1)	MQLONG	none	0
<i>KnownDestCount</i> (1)	MQLONG	none	0
<i>UnknownDestCount</i> (1)	MQLONG	none	0
<i>InvalidDestCount</i> (1)	MQLONG	none	0
<i>ObjectRecOffset</i> (1)	MQLONG	none	0
<i>ResponseRecOffset</i> (1)	MQLONG	none	0
<i>ObjectRecPtr</i> (1)	MQPTR	none	null(2)
<i>ResponseRecPtr</i> (1)	MQPTR	none	null(2)
<i>AlternateSecurityId</i> (3)	MQCHAR40	MQSID_NONE	null
<i>ResolvedQName</i> (3)	MQCHAR48	none	blanks
<i>ResolvedQMgrName</i> (3)	MQCHAR48	none	blanks

Note:

1. This field is used on Version 5 or later products only.
2. This value is a null pointer in C and null bytes otherwise.
3. This field is used on Version 5.1 products only.

MQOR (object record)

Table 323. MQOR structure

Field name	Data type	Name of constant	Value of constant
<i>ObjectName</i>	MQCHAR48	none	blanks
<i>ObjectQMgrName</i>	MQCHAR48	none	blanks

Note: This structure is supported on Version 5 or later products only.

MQPMO (put-message options)

Table 324. MQPMO structure

Field name	Data type	Name of constant	Value of constant
<i>StrucId</i>	MQCHAR4	MQPMO_STRUC_ID	'PMOb'
<i>Version</i>	MLONG	MQPMO_VERSION_1	1
<i>Options</i>	MLONG	MQPMO_NONE	0
<i>Timeout</i>	MLONG	none	-1
<i>Context</i>	MQHOBJ	none	0
<i>KnownDestCount</i>	MLONG	none	0
<i>UnknownDestCount</i>	MLONG	none	0
<i>InvalidDestCount</i>	MLONG	none	0
<i>ResolvedQName</i>	MQCHAR48	none	blanks
<i>ResolvedQMgrName</i>	MQCHAR48	none	blanks
<i>RecsPresent</i> (1)	MLONG	none	0
<i>PutMsgRecFields</i> (1)	MLONG	MQPMRF_NONE	0
<i>PutMsgRecOffset</i> (1)	MLONG	none	0
<i>ResponseRecOffset</i> (1)	MLONG	none	0
<i>PutMsgRecPtr</i> (1)	none	nulls	
<i>ResponseRecPtr</i> (1)	MQPTR	none	nulls
Note:			
1. This field is used on Version 5 or later products; that is, only when MQMD_VERSION_2 is selected.			

MQPMR (put-message record)

Table 325. MQPMR structure

Field name	Data type	Name of constant	Value of constant
<i>Msgid</i>	MQBYTE24	MQMI_NONE	nulls
<i>CorrelId</i>	MQBYTE24	MQCI_NONE	nulls
<i>GroupId</i>	MQBYTE24	MQGI_NONE	nulls
<i>Feedback</i>	MLONG	MQFB_NONE	0
<i>AccountingToken</i>	MQBYTE32	MQACT_NONE	nulls

MQRFH

MQRFH & MQRFH2 (rules and formatting header)

Table 326. MQRFH and MQRFH2 structure

Field name	Data type	Name of constant	Value of constant
<i>StrucId</i>	MQCHAR4	MQRFH_STRUC_ID	'RFHb1'
<i>Version</i>	MQLONG	MQRFH_VERSION_2	2
<i>StrucLength</i>	MQLONG	MQRFH_STRUC_LENGTH_FIXED_2	36
<i>Encoding</i>	MQLONG	MQENC_NATIVE	
<i>CodedCharSetId</i>	MQLONG	MQCCSI_INHERIT	-2
<i>Format</i>	MQCHAR8	MQFMT_NONE	'bbbbbbbb'
<i>Flags</i>	MQLONG	MQRFH_NONE	0x00000000
<i>NameValueCCSID</i>	MQLONG	None	128
<i>NameValueLength(1)</i>	MQLONG	None	128
<i>NameValueData(1)</i>	MQCHARn	None	128

Note:

1. These two fields only apply to the MQRFH2 structure, and are optional.

MQRMH (reference-message header)

Table 327. MQRMH structure

Field name	Data type	Name of constant	Value of constant
<i>StrucId</i>	MQCHAR4	MQRMH_STRUC_ID	'RMHb'
<i>Version</i>	MQLONG	MQRMH_VERSION_1	1
<i>StrucLength</i>	MQLONG	none	0
<i>Encoding</i>	MQLONG	MQENC_NATIVE OS/390: OS/2(C): OS/2(COBOL): AS/400: UNIX systems: Windows NT(C): Windows NT(COBOL):	785 546 17 273 273 546 17
<i>CodedCharSetId</i>	MQLONG	MQCCSI_Q_MGR	0
<i>Format</i>	MQCHAR8	MQFMT_NONE	'bbbbbbbb'
<i>Flags</i>	MQLONG	MQRMHF_NOT_LAST	0
<i>ObjectType</i>	MQCHAR8	none	'bbbbbbbb'
<i>ObjectInstanceId</i>	MQBYTE24	MQOII_NONE	nulls
<i>SrcEnvLength</i>	MQLONG	none	0
<i>SrcEnvOffset</i>	MQLONG	none	0
<i>SrcNameLength</i>	MQLONG	none	0
<i>SrcNameOffset</i>	MQLONG	none	0
<i>DestEnvLength</i>	MQLONG	none	0
<i>DestEnvOffset</i>	MQLONG	none	0
<i>DestEnvLength</i>	MQLONG	none	0
<i>DestEnvOffset</i>	MQLONG	none	0
<i>DataLogicalLength</i>	MQLONG	none	0
<i>DataLogicalOffset</i>	MQLONG	none	0
<i>DataLogicalOffset2</i>	MQLONG	none	0

MQRR

MQRR (response record)

Table 328. MQRR structure

Field name	Data type	Name of constant	Value of constant
<i>CompCode</i>	MQLONG	MQCC_OK	0
<i>Reason</i>	MQLONG	MQRC_NONE	0

Note: This structure is supported on Version 5 or later products only.

MQTM (trigger message)

Table 329. MQTM structure

Field name	Data type	Name of constant	Value of constant
<i>StrucId</i>	MQCHAR4	MQTM_STRUC_ID	'TMbb'
<i>Version</i>	MQLONG	MQTM_VERSION_1	1
<i>QName</i>	MQCHAR48	none	blanks
<i>ProcessName</i>	MQCHAR48	none	blanks
<i>TriggerData</i>	MQCHAR64	none	blanks
<i>ApplType</i>	MQLONG	none	0
<i>ApplId</i>	MQCHAR256	none	blanks
<i>EnvData</i>	MQCHAR128	none	blanks
<i>UserData</i>	MQCHAR128	none	blanks

Note: This structure is not supported on Windows.

MQTMC (trigger message in character format)

Table 330. MQTMC structure

Field name	Data type	Name of constant	Value of constant
<i>StrucId</i>	MQCHAR4	MQTMC_STRUC_ID	'TMCb'
<i>Version</i>	MQCHAR4	MQTMC_VERSION_1	'bbb1'
<i>QName</i>	MQCHAR48	none	blanks
<i>ProcessName</i>	MQCHAR48	none	blanks
<i>TriggerData</i>	MQCHAR64	none	blanks
<i>ApplType</i>	MQCHAR4	none	blanks
<i>ApplId</i>	MQCHAR256	none	blanks
<i>EnvData</i>	MQCHAR128	none	blanks
<i>UserData</i>	MQCHAR128	none	blanks

Note: This structure is supported on AS/400 only.

MQTMC2 (trigger message—character format 2)

Table 331. MQTMC2 structure

Field name	Data type	Name of constant	Value of constant
<i>StrucId</i>	MQCHAR4	MQTMC_STRUC_ID	'TMCb'
<i>Version</i>	MQCHAR4	MQTMC_VERSION_2	'bbb2'
<i>QName</i>	MQCHAR48	none	blanks
<i>ProcessName</i>	MQCHAR48	none	blanks
<i>TriggerData</i>	MQCHAR64	none	blanks
<i>ApplType</i>	MQCHAR4	none	blanks
<i>ApplId</i>	MQCHAR256	none	blanks
<i>EnvData</i>	MQCHAR128	none	blanks
<i>UserData</i>	MQCHAR128	none	blanks
<i>QmgrName</i>	MQCHAR48	none	blanks

Note: This structure is not supported on Windows or VSE/ESA.

MQTXP

MQTXP (Transport exit parameter structure)

Table 332. MQTXP structure

Field name	Data type	Name of constant	Value of constant
<i>StrucId</i>	MQCHAR4	MQTXP_STRUC_ID	TXPb
<i>Version</i>	MQLONG	MQTXP_VERSION_1	1
<i>Reserved</i>	MQLONG	Reserved	0
<i>ExitReason</i>	MQLONG		
<i>ExitUserArea</i>	MQBYTE16		
<i>TransportType</i>	MQLONG		
<i>RetryCount</i>	MQLONG		
<i>DataLength</i>	MQLONG		
<i>SessionId</i>	MQLONG		
<i>GroupId</i>	MQLONG		
<i>DataId</i>	MQLONG		
<i>ExitResponse</i>	MQLONG		
<i>Feedback</i>	MQLONG	Reserved	0

MQWCR (Cluster workload cluster-record structure)

Table 333. MQWCR structure

Field name	Data type	Name of constant	Value of constant
<i>ClusterName</i>	MQCHAR48	none	0
<i>ClusterRecOffset</i>	MQLONG	none	0
<i>ClusterFlags</i>	MQLONG	none	0

MQWDR (Cluster workload destination-record structure)

Table 334. MQWDR structure

Field name	Data type	Name of constant	Value of constant
<i>StrucId</i>	MQCHAR4	MQWDR_STRUC_ID	'WDRb'
<i>Version</i>	MQLONG	MQWDR_VERSION_1	1
<i>StrucLength</i>	MQLONG	MQWDR_LENGTH_1	

Table 334. MQWDR structure (continued)

Field name	Data type	Name of constant	Value of constant
<i>QMgrFlags</i>	MLONG	none	0
<i>QMgrIdentifier</i>	MQCHAR48	none	0
<i>QMgrName</i>	MQCHAR48	none	0
<i>ClusterRecOffset</i>	MLONG	none	0
<i>ChannelState</i>	MLONG	none	0
<i>ChannelDefOffset</i>	PPMQWDR	none	0

MQWIH (Work Information Header)

Table 335. MQWIH structure

Field name	Data type	Name of constant	Value of constant
<i>StrucId</i>	MQCHAR4	MQWIH_STRUC_ID	'WIHb'
<i>Version</i>	MLONG	MQWIH_VERSION_1	1
<i>StrucLength</i>	MLONG	MQWIH_LENGTH_1	120
<i>Encoding</i>	MLONG	none	0
<i>CodedCharSetId</i>	MLONG	none	0
<i>Format</i>	MQCHAR8	MQFMT_NONE	'bbbbbbbb'
<i>Flags</i>	MLONG	MQWIH_NONE	0
<i>ServiceName</i>	MQCHAR32	none	blanks
<i>ServiceStep</i>	MQCHAR8	none	blanks
<i>Msgtoken</i>	MQBYTE16	MQMTOK_NONE	nulls
<i>Reserved</i>	MQCHAR32	none	blanks

Note: This structure is supported on Version 5.1 products and OS/390 only.

MQWQR (Cluster workload queue-record structure)

Table 336. MQWQR structure

Field name	Data type	Name of constant	Value of constant
<i>StrucId</i>	MQCHAR4	MQWQR_STRUC_ID	'WQRb'
<i>Version</i>	MLONG	MQWQR_VERSION_1	1

MQWQR

Table 336. MQWQR structure (continued)

Field name	Data type	Name of constant	Value of constant
<i>StrucLength</i>	MQLONG	MQWQR_LENGTH_1	
<i>QFlags</i>	MQLONG	none	0
<i>QName</i>	MQCHAR48	none	0
<i>QType</i>	MQLONG	none	0
<i>QDesc</i>	MQCHAR64	none	0
<i>DefBind</i>	MQLONG	none	0
<i>DefPersistence</i>	MQLONG	none	0
<i>DefPriority</i>	MQLONG	none	0
<i>InhibitPut</i>	MQLONG	none	0

MQWXP (Cluster workload exit parameter structure)

Table 337. MQWXP structure

Field name	Data type	Name of constant	Value of constant
<i>StrucId</i>	MQCHAR4	MQWXP_STRUC_ID	'WXPb'
<i>Version</i>	MQLONG	MQWXP_VERSION_1	1
<i>ExitId</i>	MQLONG	MQXT_CLUSTER_WORKLOAD_EXIT	
<i>ExitReason</i>	MQLONG	none	0
<i>ExitResponse</i>	MQLONG	none	0
<i>ExitResponse2</i>	MQLONG	none	0
<i>Feedback</i>	MQLONG	none	0
<i>Reserved</i>	MQLONG	none	0
<i>ExitUserArea</i>	MQBYTE16	MQXUA_NONE	0
<i>ExitData</i>	MQCHAR32	MQ_EXIT_DATA_LENGTH	
<i>MsgDescPtr</i>	PMQMD		
<i>MsgBufferPtr</i>	PMQVOID		
<i>MsgBufferLength</i>	MQLONG		
<i>MsgLength</i>	MQLONG		
<i>QMgrName</i>	MQCHAR48	MQ_Q_MGR_NAME_LENGTH	
<i>DestinationCount</i>	MQLONG		0
<i>DestinationChosen</i>	MQLONG	none	0

Table 337. MQWXP structure (continued)

Field name	Data type	Name of constant	Value of constant
<i>DestinationArrayPtr</i>	PPMQWDR	none	0
<i>QArrayPtr</i>	PPMQWQR	none	0

MQXP (API-crossing exit parameter block)

Table 338. MQXP structure

Field name	Data type	Name of constant	Value of constant
<i>StrucId</i>	MQCHAR4	MQXP_STRUC_ID	'XQHb'
<i>Version</i>	MLONG	MQXP_VERSION_1	1
<i>ExitId</i>	MLONG	MQXT_API_CROSSING_EXIT	1
<i>ExitReason</i>	MLONG	none	0
<i>ExitResponse</i>	MLONG	MQXCC_OK	0
<i>ExitCommand</i>	MLONG	none	0
<i>ExitParmCount</i>	MLONG	none	0
<i>Reserved</i>	MLONG	none	0
<i>ExitUserArea</i>	MQBYTE16	MQXUA_NONE	nulls

Note: This structure is supported on OS/390 only.

MQXQH (transmission-queue header)

Table 339. MQXQH structure

Field name	Data type	Name of constant	Value of constant
<i>StrucId</i>	MQCHAR4	MQXQH_STRUC_ID	'XQHb'
<i>Version</i>	MLONG	MQXQH_VERSION_1	1
<i>RemoteQName</i>	MQCHAR48	none	blanks
<i>RemoteQMgrName</i>	MQCHAR48	none	blanks
<i>MsgDesc</i>	MQMD	Names and values as in Table 320 on page 169	

MQXWD

MQXWD (Exit wait descriptor structure)

Table 340. MQXWD structure

Field name	Data type	Name of constant	Value of constant
<i>StrucId</i>	MQCHAR4	MQXWD_STRUC_ID	MQXWD_STRUC_ID
<i>Version</i>	MQLONG	MQXWD_VERSION_1	MQXWD_VERSION_1
<i>Reserved1</i>	MQLONG	Reserved	0
<i>Reserved2</i>	MQLONG	Reserved	0
<i>Reserved3</i>	MQLONG	Reserved	0
<i>ECB</i>	MQLONG	None	0

MQZED (Entity Data structure)

Table 341. MQZED structure

Field name	Data type	Name of constant	Value of constant
<i>StrucId</i>	MQCHAR4	MQZED_STRUC_ID	'ZEDb'
<i>Version</i>	MQLONG	MQZED_VERSION_1	1
<i>EntityNamePtr</i>	PMQCHAR	none	null(1)
<i>EntityDomainPtr</i>	PMQCHAR	none	null(1)
<i>SecurityId</i>	MQBYTE	none	

Note:

1. This value is a null-terminated string.

Chapter 11. Attributes of MQSeries objects

Full details of these attributes can be found in the *MQSeries Application Programming Reference* manual.

Local and model queue attributes

Table 342. Attributes of local and model queues

Attribute	Data type	Name of Selector
<i>AlterationDate</i> (8)	MQCHAR12	MQCA_ALTERATION_DATE
<i>AlterationTime</i> (8)	MQCHAR8	MQCA_ALTERATION_TIME
<i>Archive</i> (1, 9)	MLONG	MQIA_ARCHIVE
<i>BackoutRequeueQName</i> (10)	MQCHAR48	MQCA_BACKOUT_REQ_Q_NAME
<i>BackoutThreshold</i> (10)	MLONG	MQIA_BACKOUT_THRESHOLD
<i>CFStrucName</i> (3)	MQCHAR12	MQCACF_STRUC_NAME
<i>ClusterName</i> (1, 8)	MQCHAR48	MQCA_CLUSTER_NAME
<i>ClusterNamelist</i> (1, 8)	MQCHAR48	MQCA_CLUSTER_NAMELIST
<i>CreationDate</i>	MQCHAR12	MQCA_CREATION_DATE
<i>CreationTime</i>	MQCHAR8	MQCA_CREATION_TIME
<i>CurrentQDepth</i> (1)	MLONG	MQIA_CURRENT_Q_DEPTH
<i>DefBind</i> (1, 8)	MLONG	MQIA_DEF_BIND
<i>DefinitionType</i>	MLONG	MQIA_DEFINITION_TYPE
<i>DefInputOpenOption</i> (10)	MLONG	MQIA_DEF_INPUT_OPEN_OPTION
<i>DefPersistence</i>	MLONG	MQIA_DEF_PERSISTENCE
<i>DefPriority</i> (10)	MLONG	MQIA_DEF_PRIORITY
<i>DistLists</i> (2)	MLONG	MQIA_DIST_LISTS
<i>HardenGetBackout</i> (3)	MLONG	MQIA_HARDEN_GET_BACKOUT
<i>IndexType</i> (3)	MLONG	MQIA_INDEX_TYPE
<i>InhibitGet</i>	MLONG	MQIA_INHIBIT_GET
<i>InhibitPut</i>	MLONG	MQIA_INHIBIT_PUT
<i>InitiationQName</i> (4, 10)	MQCHAR48	MQCA_INITIATION_Q_NAME
<i>MaxMsgLength</i>	MLONG	MQIA_MAX_MSG_LENGTH
<i>MaxQDepth</i>	MLONG	MQIA_MAX_Q_DEPTH

Local and model queues

Table 342. Attributes of local and model queues (continued)

Attribute	Data type	Name of Selector
<i>MsgDeliverySequence</i> (10)	MQLONG	MQIA_MSG_DELIVERY_SEQUENCE
<i>OpenInputCount</i> (1)	MQLONG	MQIA_OPEN_INPUT_COUNT
<i>OpenOutputCount</i> (1)	MQLONG	MQIA_OPEN_OUTPUT_COUNT
<i>ProcessName</i> (4, 5, 10)	MQCHAR48	MQCA_PROCESS_NAME
<i>QDepthHighEvent</i> (6)	MQLONG	MQIA_Q_DEPTH_HIGH_EVENT (PCF)
<i>QDepthHighLimit</i> (6)	MQLONG	MQIA_Q_DEPTH_HIGH_LIMIT (PCF)
<i>QDepthLowEvent</i> (6)	MQLONG	MQIA_Q_DEPTH_LOW_EVENT (PCF)
<i>QDepthLowLimit</i> (6)	MQLONG	MQIA_Q_DEPTH_LOW_LIMIT (PCF)
<i>QDepthMaxEvent</i> (6)	MQLONG	MQIA_Q_DEPTH_MAX_EVENT (PCF)
<i>QDesc</i>	MQCHAR64	MQCA_Q_DESC
<i>QName</i>	MQCHAR48	MQCA_Q_NAME
<i>QServiceInterval</i> (6)	MQLONG	MQIA_Q_SERVICE_INTERVAL (PCF)
<i>QServiceIntervalEvent</i> (6)	MQLONG	MQIA_Q_SERVICE_INTERVAL_EVENT (PCF)
<i>QSGDisp</i> (1, 3)	MQLONG	MQIA_QSG_DISP
<i>QType</i>	MQLONG	MQIA_Q_TYPE
<i>RetentionInterval</i> (10)	MQLONG	MQIA_RETENTION_INTERVAL
<i>Scope</i> (4, 7, 10)	MQLONG	MQIA_SCOPE
<i>Shareability</i>	MQLONG	MQIA_SHAREABILITY
<i>StorageClass</i> (3)	MQCHAR8	MQCA_STORAGE_CLASS
<i>TriggerControl</i> (4, 10)	MQLONG	MQIA_TRIGGER_CONTROL
<i>TriggerData</i> (4, 10)	MQCHAR64	MQCA_TRIGGER_DATA
<i>TriggerDepth</i> (4, 10)	MQLONG	MQIA_TRIGGER_DEPTH
<i>TriggerMsgPriority</i> (4, 10)	MQLONG	MQIA_TRIGGER_MSG_PRIORITY
<i>TriggerType</i> (4)	MQLONG	MQIA_TRIGGER_TYPE
<i>Usage</i>	MQLONG	MQIA_USAGE

Table 342. Attributes of local and model queues (continued)

Attribute	Data type	Name of Selector
Note:		
1. Applies to local queues only		
2. Applies to OS/2, AS/400, UNIX systems, and Windows NT only		
3. Applies to OS/390 only		
4. Does not apply to Windows Version 2.0 or Version 2.1		
5. Optional in the case of triggering channels on Version 5 products, or later, and AS/400		
6. Applies to OS/2, AS/400, UNIX systems, Windows NT, and Windows Version 2.1 only		
7. Does not apply to OS/390		
8. Applies to Version 5.1 products and OS/390 only		
9. Applies to AIX, HP-UX, and Sun Solaris only		
10. Does not apply to VSE/ESA		

Local definition of remote queue attributes

Table 343. Attributes of local definitions of remote queues

Attribute	Data type	Name of Selector
<i>AlterationDate</i> (2)	MQCHAR12	MQCA_ALTERATION_DATE
<i>AlterationTime</i> (2)	MQCHAR8	MQCA_ALTERATION_TIME
<i>ClusterName</i> (2)	MQCHAR48	MQCA_CLUSTER_NAME
<i>ClusterNameList</i> (2)	MQCHAR48	MQCA_CLUSTER_NAMELIST
<i>DefBind</i> (2)	MQLONG	MQIA_DEF_BIND
<i>DefPersistence</i>	MQLONG	MQIA_DEF_PERSISTENCE
<i>DefPriority</i> (3)	MQLONG	MQIA_DEF_PRIORITY
<i>InhibitPut</i>	MQLONG	MQIA_INHIBIT_PUT
<i>QDesc</i>	MQCHAR64	MQCA_Q_DESC
<i>QName</i>	MQCHAR48	MQCA_Q_NAME
<i>QSGDisp</i> (4)	MQLONG	MQIA_QSG_DISP
<i>QType</i>	MQLONG	MQIA_Q_TYPE
<i>RemoteQMgrName</i>	MQCHAR48	MQCA_REMOTE_Q_MGR_NAME
<i>RemoteQName</i>	MQCHAR48	MQCA_REMOTE_Q_NAME
<i>Scope</i> (1)	MQLONG	MQIA_SCOPE
<i>XmitQName</i> (3)	MQCHAR48	MQCA_XMIT_Q_NAME
Note:		
1. Does not apply to OS/390, Windows Version 2.0, Windows Version 2.1, or VSE/ESA		
2. Applies to Version 5.1 products and OS/390 only		
3. Does not apply to VSE/ESA		
4. Applies to OS/390 only		

Remote queues

Alias queue attributes

Table 344. Attributes of alias queues

Attribute	Data type	Name of Selector
<i>AlterationDate</i> (1)	MQCHAR12	MQCA_ALTERATION_DATE
<i>AlterationTime</i> (1)	MQCHAR8	MQCA_ALTERATION_TIME
<i>BaseQName</i>	MQCHAR48	MQCA_BASE_Q_NAME
<i>ClusterName</i> (1)	MQCHAR48	MQCA_CLUSTER_NAME
<i>ClusterNameList</i> (1)	MQCHAR48	MQCA_CLUSTER_NAMELIST
<i>DefBind</i> (1)	MQLONG	MQIA_DEF_BIND
<i>DefPersistence</i>	MQLONG	MQIA_DEF_PERSISTENCE
<i>DefPriority</i>	MQLONG	MQIA_DEF_PRIORITY
<i>InhibitGet</i>	MQLONG	MQIA_INHIBIT_GET
<i>InhibitPut</i>	MQLONG	MQIA_INHIBIT_PUT
<i>QDesc</i>	MQCHAR64	MQCA_Q_DESC
<i>QName</i>	MQCHAR48	MQCA_Q_NAME
<i>QSGDisp</i> (3)	MQLONG	MQIA_QSG_DISP
<i>QType</i>	MQLONG	MQIA_Q_TYPE
<i>Scope</i> (2)	MQLONG	MQIA_SCOPE
Notes:: 1. Applies to Version 5.1 products and OS/390 only 2. Does not apply to OS/390, Windows Version 2.0, Windows Version 2.1, or VSE/ESA 3. Applies to OS/390 only		

Namelist attributes

Table 345. Attributes of namelists

Attribute	Data type	Name of Selector
<i>AlterationDate</i>	MQCHAR12	MQCA_ALTERATION_DATE
<i>AlterationTime</i>	MQCHAR8	MQCA_ALTERATION_TIME
<i>NameCount</i>	MQLONG	MQIA_NAME_COUNT
<i>NamelistDesc</i>	MQCHAR64	MQCA_NAMELIST_DESC
<i>NamelistName</i>	MQCHAR48	MQCA_NAMELIST_NAME
<i>Names</i>	MQCHAR48 x <i>NameCount</i>	MQCA_NAMES

Table 345. Attributes of namelists (continued)

Attribute	Data type	Name of Selector
<i>QSGDisp</i> (2)	MQLONG	MQIA_QSG_DISP
Notes:		
1. Namelists are supported on OS/390 and Version 5.1 products only.		
2. Applies to OS/390 only		

Process definition attributes

Table 346. Attributes of process definitions

Attribute	Data type	Name of Selector
<i>AlterationDate</i> (3)	MQCHAR12	MQCA_ALTERATION_DATE
<i>AlterationTime</i> (3)	MQCHAR8	MQCA_ALTERATION_TIME
<i>ApplId</i>	MQCHAR256	MQCA_APPL_ID
<i>ApplType</i>	MQLONG	MQIA_APPL_TYPE
<i>EnvData</i>	MQCHAR128	MQCA_ENV_DATA
<i>ProcessDesc</i>	MQCHAR64	MQCA_PROCESS_DESC
<i>ProcessName</i>	MQCHAR48	MQCA_PROCESS_NAME
<i>QSGDisp</i> (4)	MQLONG	MQIA_QSG_DISP
<i>UserData</i>	MQCHAR128	MQCA_USER_DATA
Notes::		
1. On Version 5 products, or later, the process definition object is optional in the case of triggering channels.		
2. Process definitions are not supported on Windows Version 2.0, Windows Version 2.1, or VSE/ESA.		
3. This attribute applies to Version 5.1 products and OS/390 only.		
4. Applies to OS/390 only.		

Queue manager attributes

Table 347. Attributes of a queue manager

Attribute	Data type	Name of Selector
<i>AlterationDate</i> (5)	MQCHAR12	MQCA_ALTERATION_DATE
<i>AlterationTime</i> (5)	MQCHAR8	MQCA_ALTERATION_TIME
<i>AuthorityEvent</i> (2)	MQLONG	MQIA_AUTHORITY_EVENT
<i>CodedCharSetId</i>	MQLONG	MQIA_CODED_CHAR_SET_ID
<i>ClusterWorkloadData</i> (5)	MQCHAR32	MQCA_CLUSTER_WORKLOAD_DATA

Queue manager

Table 347. Attributes of a queue manager (continued)

Attribute	Data type	Name of Selector
<i>ClusterWorkloadExit</i> (5)	MQCHARn	MQCA_CLUSTER_WORKLOAD_EXIT
<i>ClusterWorkloadLength</i> (5)	MQLONG	MQCA_CLUSTER_WORKLOAD_LENGTH
<i>CommandInputQName</i>	MQCHAR48	MQCA_COMMAND_INPUT_Q_NAME
<i>CommandLevel</i>	MQLONG	MQIA_COMMAND_LEVEL
<i>DeadLetterQName</i> (2, 3)	MQCHAR48	MQCA_DEAD_LETTER_Q_NAME
<i>DefXmitQName</i>	MQCHAR48	MQCA_DEF_XMIT_Q_NAME
<i>DistLists</i> (4)	MQLONG	MQIA_DIST_LISTS
<i>InhibitEvent</i> (2)	MQLONG	MQIA_INHIBIT_EVENT
<i>IntraGroupQueuing</i> (6)	MQLONG	MQIA_INTRA_GROUP_QUEUING
<i>LocalEvent</i> (2)	MQLONG	MQIA_LOCAL_EVENT
<i>MaxHandles</i>	MQLONG	MQIA_MAX_HANDLES
<i>MaxMsgLength</i>	MQLONG	MQIA_MAX_MSG_LENGTH
<i>MaxPriority</i>	MQLONG	MQIA_MAX_PRIORITY
<i>MaxUncommittedMsgs</i>	MQLONG	MQIA_MAX_UNCOMMITTED_MSGS
<i>PerformanceEvent</i> (2)	MQLONG	MQIA_PERFORMANCE_EVENT
<i>Platform</i>	MQLONG	MQIA_PLATFORM
<i>QMgrDesc</i>	MQCHAR64	MQCA_Q_MGR_DESC
<i>QMgrIdentifier</i> (5)	MQCHAR48	MQCA_Q_MGR_NAME
<i>QMgrName</i>	MQCHAR48	MQCA_Q_MGR_NAME
<i>QSGDisp</i> (6)	MQLONG	MQIA_QSG_DISP
<i>RemoteEvent</i> (2)	MQLONG	MQIA_REMOTE_EVENT
<i>RepositoryName</i> (5)	MQCHAR48	MQCA_REPOSITORY_NAME
<i>RepositoryNameList</i> (5)	MQCHAR48	MQCA_REPOSITORY_NAMELIST
<i>StartStopEvent</i> (2)	MQLONG	MQIA_START_STOP_EVENT
<i>SyncPoint</i>	MQLONG	MQIA_SYNCPOINT
<i>TriggerInterval</i> (2, 3)	MQLONG	MQIA_TRIGGER_INTERVAL

Note:

1. Queue manager attributes are not supported on VSE/ESA.
2. This attribute does not apply to Windows Version 2.0.
3. This attribute does not apply to Windows Version 2.1.
4. This attribute applies to Version 5 products, or later, and AS/400 only.
5. This attribute applies to Version 5.1 products and OS/390 only.
6. Applies to OS/390 only.

Chapter 12. MQI constants

This Chapter describes groups of MQI constants. Within each group, the constants are listed in numeric order.

For further information, see the following manuals:

- For MQI call constants, see the *MQSeries Application Programming Reference* manual.
- For events, commands, and responses, see the *MQSeries Programmable System Management* manual.

The information in these tables applies to all platforms except where indicated by an X.

Note: The presence or absence of an X is not an indication of whether the constant is supplied in the MQI header. In the case of input fields, for example, MQIA_*, the system does not *act on* those indicated with an X. In the case of output fields, for example, MQRC_*, the system does not *generate* those indicated with an X.

MQ_* (Lengths of character string and byte fields)

Table 348. MQ_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQ_ABEND_CODE_LENGTH				(1)	X	X	X	X		4	00000004
MQ_ACCOUNTING_TOKEN_LENGTH							X			32	00000020
MQ_APPL_IDENTITY_DATA_LENGTH							X			32	00000020
MQ_APPL_NAME_LENGTH							X			28	0000001C
MQ_APPL_ORIGIN_DATA_LENGTH							X			4	00000004
MQ_ATTENTION_ID_LENGTH				(1)	X	X	X	X		4	00000004
MQ_AUTHENTICATOR_LENGTH				(1)	X	X	X	X		8	00000008
MQ_BRIDGE_NAME_LENGTH				(1)	X	X	X	X		24	00000018
MQ_CANCEL_CODE_LENGTH				(1)	X	X	X	X		4	00000004
MQ_CF_STRUC_NAME_LENGTH		X	X	X	X	X	X	X	X	12	0000000C

MQ_*

Table 348. MQ_* constants (continued)

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQ_CHANNEL_DATE_LENGTH							X			12	0000000C
MQ_CHANNEL_DESC_LENGTH							X			64	00000040
MQ_CHANNEL_NAME_LENGTH							X			20	00000014
MQ_CHANNEL_TIME_LENGTH							X			8	00000008
MQ_CLUSTER_NAME_LENGTH				(1)	X	X	X	X		48	00000030
MQ_CONN_NAME_LENGTH							X			264	00000108
MQ_CONN_TAG_LENGTH		X	X	X	X	X	X	X	X	128	00000080
MQ_CORREL_ID_LENGTH							X			24	00000018
MQ_CREATION_DATE_LENGTH										12	0000000C
MQ_CREATION_TIME_LENGTH										8	00000008
MQ_DATE_LENGTH				(1)	X	X	X	X		12	0000000C
MQ_EXIT_DATA_LENGTH							X			32	00000020
MQ_EXIT_NAME_LENGTH	X X	X X	X X	X X	X X	X X	X X	X X	X X	8 20 128	00000008 00000014 00000080
MQ_EXIT_USER_AREA_LENGTH							X			16	00000010
MQ_FACILITY_LENGTH				(1)	X	X	X	X		8	00000008
MQ_FACILITY_LIKE_LENGTH				(1)	X	X	X	X		4	00000004
MQ_FORMAT_LENGTH							X			8	00000008
MQ_FUNCTION_LENGTH				(1)	X	X	X	X		4	00000004
MQ_GROUP_ID_LENGTH	X			(1)	X	X	X	X		24	00000018
MQ_LTERM_OVERRIDE_LENGTH				(1)	X	X	X	X		8	00000008
MQ_LUWID_LENGTH							X			16	00000010
MQ_MCA_JOB_NAME_LENGTH							X			28	0000001C
MQ_MCA_NAME_LENGTH							X			20	00000014
MQ_MFS_MAP_NAME_LENGTH				(1)	X	X	X	X		8	00000008
MQ_MODE_NAME_LENGTH							X			8	00000008
MQ_MSG_HEADER_LENGTH							X			4000	00000FA0
MQ_MSG_ID_LENGTH							X			24	00000018

Table 348. MQ_* constants (continued)

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQ_MSG_TOKEN_LENGTH				(1)	X	X	X	X		16	00000010
MQ_NAMELIST_DESC_LENGTH				(1)	X	X	X	X		64	00000040
MQ_NAMELIST_NAME_LENGTH				(1)	X	X	X	X		48	00000030
MQ_OBJECT_INSTANCE_ID_LENGTH				(1)	X	X	X	X		24	00000018
MQ_OBJECT_NAME_LENGTH							X			48	00000030
MQ_PASSWORD_LENGTH							X			12	0000000C
MQ_PROCESS_APPL_ID_LENGTH							X			256	00000100
MQ_PROCESS_DESC_LENGTH							X			64	00000040
MQ_PROCESS_ENV_DATA_LENGTH							X			128	00000080
MQ_PROCESS_NAME_LENGTH							X			48	00000030
MQ_PROCESS_USER_DATA_LENGTH							X			128	00000080
MQ_PUT_APPL_NAME_LENGTH							X			28	0000001C
MQ_PUT_DATE_LENGTH							X			8	00000008
MQ_PUT_TIME_LENGTH							X			8	00000008
MQ_Q_DESC_LENGTH										64	00000040
MQ_Q_MGR_DESC_LENGTH										64	00000040
MQ_Q_MGR_IDENTIFIER				(1)	X	X	X	X		48	00000030
MQ_Q_MGR_NAME_LENGTH										48	00000030
MQ_Q_NAME_LENGTH										48	00000030
MQ_REMOTE_SYS_ID_LENGTH				(1)	X	X	X	X		4	00000004
MQ_SECURITY_ID_LENGTH	X			(1)	X	X	X	X		32	00000020
MQ_SERVICE_NAME_LENGTH				(1)	X	X	X	X		32	00000020
MQ_SERVICE_STEP_LENGTH				(1)	X	X	X	X		8	00000008
MQ_SHORT_CONN_NAME_LENGTH							X			20	00000014
MQ_START_CODE_LENGTH				(1)	X	X	X	X		4	00000004
MQ_STORAGE_CLASS_LENGTH		X	X	X	X	X	X	X	X	8	00000008
MQ_TIME_LENGTH				(1)	X	X	X	X		8	00000008
MQ_TOTAL_EXIT_DATA_LENGTH	X						X			999	000003E7
MQ_TOTAL_EXIT_NAME_LENGTH	X						X			999	000003E7

MQ_*

Table 348. MQ_* constants (continued)

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQ_TP_NAME_LENGTH							X			64	00000040
MQ_TRAN_INSTANCE_ID_LENGTH				(1)	X	X	X	X		16	00000010
MQ_TRANSACTION_ID_LENGTH				(1)	X	X	X	X		4	00000004
MQ_TRIGGER_DATA_LENGTH										64	00000040
MQ_USER_ID_LENGTH							X			12	0000000C
Note: 1. Supported on AIX, HP-UX, and Sun Solaris only											

MQACT_* (Accounting token)

Table 349. MQACT_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Value
MQACT_NONE										32 nulls

MQACT_* (Action option)

Table 350. MQACT_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQACT_FORCE_REMOVE										1	00000001

MQACTT_* (Accounting token type)

Table 351. MQACTT_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQACTT_UNKNOWN	X			(1)	X	X	X	X		0	00000000
MQACTT_CICS_LUOW_ID	X			(1)	X	X	X	X		1	00000001
MQACTT_OS2_DEFAULT	X			(1)	X	X	X	X		4	00000004
MQACTT_DOS_DEFAULT	X			(1)	X	X	X	X		5	00000005
MQACTT_UNIX_NUMERIC_ID	X			(1)	X	X	X	X		6	00000006
MQACTT_OS400_ACCOUNT_TOKEN	X			(1)	X	X	X	X		8	00000008
MQACTT_WINDOWS_DEFAULT	X			(1)	X	X	X	X		9	00000009
MQACTT_NT_SECURITY_ID	X			(1)	X	X	X	X		11	0000000B
MQACTT_USER	X			(1)	X	X	X	X		25	00000019

Note:
1. Supported on AIX, HP-UX, and Sun Solaris only

MQAT_* (Application type)

Table 352. MQAT_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQAT_DEFAULT											(variable)
MQAT_UNKNOWN										-1	FFFFFFFF
MQAT_NO_CONTEXT										0	00000000
MQAT_CICS										1	00000001
MQAT_MVS										2	00000002
MQAT_IMS										3	00000003
MQAT_OS2										4	00000004
MQAT_DOS										5	00000005

MQAT_*

Table 352. MQAT_* constants (continued)

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQAT_AIX										6	00000006
MQAT_UNIX										6	00000006
MQAT_QMGR										7	00000007
MQAT_OS400										8	00000008
MQAT_WINDOWS										9	00000009
MQAT_CICS_VSE										10	0000000A
MQAT_WINDOWS_NT										11	0000000B
MQAT_VMS										12	0000000C
MQAT_NSK										13	0000000D
MQAT_GUARDIAN										13	0000000D
MQAT_VOS										14	0000000E
MQAT_IMS_BRIDGE					X	X	X	X		19	00000013
MQAT_XCF					X	X	X	X		20	00000014
MQAT_CICS_BRIDGE				(1)	X	X	X	X		21	00000015
MQAT_NOTES_AGENT				(1)	X	X	X	X		22	00000016
MQAT_BROKER				(1)	X	X	X	X		26	0000001A
MQAT_JAVA				(1)	X	X	X	X		28	0000001C
MQAT_DQM										29	0000001D
MQAT_USER_FIRST										65536	00010000
MQAT_USER_LAST										99999999	3B9AC9FF

MQBND_* (Binding options)

Table 353. MQBND_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQBND_BIND_ON_OPEN										0	00000000
MQBND_BIND_NOT_FIXED										1	00000001
Note: 1. Supported on AIX, HP-UX, and Sun Solaris only											

MQBO_* (Begin options)

Table 354. MQBO_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQBO_NONE	X			(1)	X	X	X	X		0	00000000
Note: 1. Supported on AIX, HP-UX, and Sun Solaris only											

MQBO_* (Begin options structure identifier)

Table 355. MQBO_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Value
MQBO_STRUC_ID	X			(1)	X	X	X	X		BObb
Note: 1. Supported on AIX, HP-UX, and Sun Solaris only										

MQBO_*

MQBO_* (Begin options version)

Table 356. MQBO_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQBO_VERSION_1	X			(1)	X	X	X	X		1	00000001
MQBO_VERSION_2	X			(1)	X	X	X	X		2	00000002
MQBO_CURRENT_VERSION	X			(1)	X	X	X	X		2	00000002
MQBO_SERIALIZE_UOW_TAG	X			(1)	X	X	X	X		1	00000001

Note:

1. Supported on AIX, HP-UX, and Sun Solaris only

MQCA_* (Character attribute selector)

Table 357. MQCA_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQCA_FIRST										2001	000007D1
MQCA_APPL_ID							X			2001	000007D1
MQCA_BASE_Q_NAME										2002	000007D2
MQCA_COMMAND_INPUT_Q_NAME							X			2003	000007D3
MQCA_CREATION_DATE										2004	000007D4
MQCA_CREATION_TIME										2005	000007D5
MQCA_DEAD_LETTER_Q_NAME							X	X		2006	000007D6
MQCA_ENV_DATA										2007	000007D7
MQCA_INITIATION_Q_NAME										2008	000007D8
MQCA_NAMELIST_DESC				(1)	X	X	X	X		2009	000007D9
MQCA_NAMELIST_NAME				(1)	X	X	X	X		2010	000007DA
MQCA_PROCESS_DESC							X			2011	000007DB
MQCA_PROCESS_NAME							X			2012	000007DC

Table 357. MQCA_* constants (continued)

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQCA_Q_DESC										2013	000007DD
MQCA_Q_MGR_DESC							X			2014	000007DE
MQCA_Q_MGR_NAME										2015	000007DF
MQCA_Q_NAME										2016	000007E0
MQCA_REMOTE_Q_MGR_NAME										2017	000007E1
MQCA_REMOTE_Q_NAME										2018	000007E2
MQCA_BACKOUT_REQ_Q_NAME							X			2019	000007E3
MQCA_NAMES				(1)	X	X	X	X		2020	000007E4
MQCA_USER_DATA							X			2021	000007E5
MQCA_STORAGE_CLASS		X	X	X	X	X	X	X	X	2022	000007E6
MQCA_TRIGGER_DATA							X	X		2023	000007E7
MQCA_XMIT_Q_NAME							X			2024	000007E8
MQCA_DEF_XMIT_Q_NAME							X			2025	000007E9
MQCA_CHANNEL_AUTO_DEF_EXIT				(1)	X	X	X	X		2026	000007EA
MQCA_ALTERATION_DATE				(1)	X	X	X	X		2027	000007EB
MQCA_ALTERATION_TIME				(1)	X	X	X	X		2028	000007EC
MQCA_CLUSTER_NAME				(1)	X	X	X	X		2029	000007ED
MQCA_CLUSTER_NAMELIST				(1)	X	X	X	X		2030	000007EE
MQCA_CLUSTER_Q_MGR_NAME				(1)	X	X	X	X		2031	000007EF
MQCA_CLUSTER_Q_MGR_IDENTIFIER				(1)	X	X	X	X		2032	000007F0
MQCA_CLUSTER_WORKLOAD_EXIT				(1)	X	X	X	X		2033	000007F1
MQCA_CLUSTER_WORKLOAD_DATA				(1)	X	X	X	X		2034	000007F2
MQCA_REPOSITORY_NAME				(1)	X	X	X	X		2035	000007F3
MQCA_REPOSITORY_NAMELIST				(1)	X	X	X	X		2036	000007F4
MQCA_CLUSTER_DATE				(1)	X	X	X	X		2037	000007F5
MQCA_CLUSTER_TIME				(1)	X	X	X	X		2038	000007F6
MQCA_CF_STRUC_NAME		X	X	X	X	X	X	X	X	2039	000007F7
MQCA_QSG_NAME		X	X	X	X	X	X	X	X	2040	000007F8
MQCA_LAST_USED											(variable)

MQCA_*

Table 357. MQCA_* constants (continued)

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQCA_LAST										4000	00000FA0

Note:
1. Supported on AIX, HP-UX, and Sun Solaris only

MQCACF_* (Character attribute command format parameter)

Table 358. MQCACF_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQCACF_CACF_FIRST							X	(1)		4	00000004
MQCACF_ACCOUNTING_TOKEN_LENGTH							X	(1)		3001	00000BB9
MQCACF_FROM_Q_NAME							X	(1)		3001	00000BB9
MQCACF_TO_Q_NAME							X	(1)		3002	00000BBA
MQCACF_FROM_PROCESS_NAME							X	(1)		3003	00000BBB
MQCACF_TO_PROCESS_NAME							X	(1)		3004	00000BBC
MQCACF_FROM_NAMELIST_NAME							X	(1)		3005	00000BBD
MQCACF_TO_NAMELIST_NAME							X	(1)		3006	00000BBE
MQCACF_FROM_CHANNEL_NAME							X	(1)		3007	00000BBF
MQCACF_TO_CHANNEL_NAME							X	(1)		3008	00000BC0
MQCACF_Q_NAMES							X	(1)		3011	00000BC3
MQCACF_PROCESS_NAMES							X	(1)		3012	00000BC4
MQCACF_NAMELIST_NAMES							X	(1)		3013	00000BC5
MQCACF_ESCAPE_TEXT							X	(1)		3014	00000BC6
MQCACF_LOCAL_Q_NAMES							X	(1)		3015	00000BC7
MQCACF_MODEL_Q_NAMES							X	(1)		3016	00000BC8
MQCACF_ALIAS_Q_NAMES							X	(1)		3017	00000BC9
MQCACF_REMOTE_Q_NAMES							X	(1)		3018	00000BCA

Table 358. MQCACF_* constants (continued)

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQCACF_SENDER_CHANNEL_NAMES							X	(1)		3019	00000BCB
MQCACF_SERVER_CHANNEL_NAMES							X	(1)		3020	00000BCC
MQCACF_REQUESTER_CHANNEL_NAMES							X	(1)		3021	00000BCD
MQCACF_RECEIVER_CHANNEL_NAMES							X	(1)		3022	00000BCE
MQCACF_OBJECT_Q_MGR_NAME							X	(1)		3022	00000BCF
MQCACF_APPL_NAME							X	(1)		3024	00000BD0
MQCACF_USER_IDENTIFIER							X	(1)		3025	00000BD1
MQCACF_AUX_ERROR_DATA_STR_1							X	(1)		3026	00000BD2
MQCACF_AUX_ERROR_DATA_STR_2							X	(1)		3027	00000BD3
MQCACF_AUX_ERROR_DATA_STR_3							X	(1)		3028	00000BD4
MQCACF_BRIDGE_NAME							X	(1)		3029	00000BD5
MQCACF_STREAM_NAME							X	(1)		3030	00000BD6
MQCACF_TOPIC							X	(1)		3031	00000BD7
MQCACF_PARENT_Q_MGR_NAME							X	(1)		3032	00000BD8
MQCACF_PUBLISH_TIMESTAMP							X	(1)		3034	00000BDA
MQCACF_STRING_DATA							X	(1)		3035	00000BDB
MQCACF_SUPPORTED_STREAM_NAME							X	(1)		3036	00000BDC
MQCACF_REG_TOPIC							X	(1)		3037	00000BDD
MQCACF_REG_TIME							X	(1)		3038	00000BDE
MQCACF_REG_USER_ID							X	(1)		3039	00000BDF
MQCACF_CHILD_Q_MGR_NAME							X	(1)		3040	00000BE0
MQCACF_REG_STREAM_NAME							X	(1)		3041	00000BE1
MQCACF_REG_Q_MGR_NAME							X	(1)		3042	00000BE2
MQCACF_REG_Q_NAME							X	(1)		3043	00000BE3
MQCACF_REG_CORREL_ID							X	(1)		3044	00000BE4
MQCACF_LAST_USED							X	(1)			(variable)
Note:											
1. Supported on Windows 2.1 only											

MQCACH_*

MQCACH_* (Channel character attribute command format parameter)

Table 359. MQCACH_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal Hex	
MQCACH_FIRST							X	(1)		3501	00000DAD
MQCACH_CHANNEL_NAME							X	(1)		3501	00000DAD
MQCACH_DESC							X	(1)		3502	00000DAE
MQCACH_MODE_NAME							X	(1)		3503	00000DAF
MQCACH_TP_NAME							X	(1)		3504	00000DB0
MQCACH_XMIT_Q_NAME							X	(1)		3505	00000DB1
MQCACH_CONNECTION_NAME							X	(1)		3506	00000DB2
MQCACH_MCA_NAME							X	(1)		3507	00000DB3
MQCACH_SEC_EXIT_NAME							X	(1)		3508	00000DB4
MQCACH_MSG_EXIT_NAME							X	(1)		3509	00000DB5
MQCACH_SEND_EXIT_NAME							X	(1)		3510	00000DB6
MQCAH_RCV_EXIT_NAME							X	(1)		3511	00000DB7
MQCACH_CHANNEL_NAMES							X	(1)		3512	00000DB8
MQCACH_SEC_EXIT_USER_DATA							X	(1)		3513	00000DB9
MQCACH_MSG_EXIT_USER_DATA							X	(1)		3514	00000DBA
MQCACH_SEND_EXIT_USER_DATA							X	(1)		3515	00000DBB
MQCACH_RCV_EXIT_USER_DATA							X	(1)		3516	00000DBC
MQCACH_USER_ID							X	(1)		3517	00000DBD
MQCACH_PASSWORD							X	(1)		3518	00000DBE
MQCACH_LAST_MSG_TIME							X	(1)		3524	00000DC4
MQCACH_LAST_MSG_DATE							X	(1)		3525	00000DC5
MQCACH_MCA_USER_ID							X	(1)		3527	00000DC7
MQCACH_CHANNEL_START_TIME							X	(1)		3528	00000DC8
MQCACH_CHANNEL_START_DATE							X	(1)		3529	00000DC9
MQCACH_MCA_JOB_NAME							X	(1)		3530	00000DCA
MQCACH_LAST_LUWID							X	(1)		3531	00000DCB
MQCACH_CURRENT_LUWID							X	(1)		3532	00000DCC

Table 359. MQCACH_* constants (continued)

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQCACH_FORMAT_NAME							X	(1)		3533	00000DCD
MQCACH_MR_EXIT_NAME							X	(1)		3534	00000DCE
MQCACH_MR_EXIT_USER_DATA							X	(1)		3535	00000DCF
MQCACH_LAST_USED							X	(1)			(variable)
Note:											
1. Supported on Windows 2.1 only											

MQCADSD_* (CICS header ADS description)

Table 360. MQCADSD_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQCADSD_NONE								X		0	00000000
MQCADSD_SEND										1	00000001
MQCADSD_RECV										16	00000010
MQCADSD_MSGFORMAT										256	00000100

MQCC_* (Completion code)

Table 361. MQCC_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQCC_UNKNOWN	X						X			-1	FFFFFFFF
MQCC_OK										0	00000000

MQCC_*

Table 361. MQCC_* constants (continued)

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQCC_WARNING										1	00000001
MQCC_FAILED										2	00000002

MQCCSI_* (Coded character set identifier)

Table 362. MQCCSI_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQCCSI_INHERIT	X									-2	FFFFFFFE
MQCCSI_EMBEDDED										-1	FFFFFFF
MQCCSI_DEFAULT										0	00000000
MQCCSI_UNDEFINED										0	00000000
MQCCSI_Q_MGR										0	00000000

MQCCT_* (CICS header conversational task)

Table 363. MQCCT_* constants

Constant	Decimal	Hex
MQCCT_NO	0	00000000
MQCCT_YES	1	00000001

MQCD_* (Channel definition structure length)

Table 364. MQCD_* constants

Constant	Decimal	Hex
MQCD_LENGTH_4		(variable)

Table 364. MQCD_* constants (continued)

Constant	Decimal	Hex
MQCD_LENGTH_5		(variable)
MQCD_LENGTH_6		(variable)
MQCD_CURRENT_LENGTH		(variable)

MQCD_* (Channel definition structure version)

Table 365. MQCD_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQCD_VRESION_1										1	00000001
MQCD_VERSION_2										2	00000002
MQCD_VERSION_3										3	00000003
MQCD_VERSION_4										4	00000004
MQCD_VERSION_5										5	00000005
MQCD_VERSION_6										6	00000006
MQCD_CURRENT_VERSION											(variable)

MQCDC_* (Channel data conversion)

Table 366. MQCDC_* constants

Constant	Decimal	Hex
MQCDC_NO_SENDER_CONVERSION	0	00000000
MQCDC_SENDER_CONVERSION	1	00000001

MQCF_* (Channel capability flags)

Table 367. MQCF_* constants

Constant	Decimal	Hex
MQCF_NONE	0	00000000
MQCF_DIST_LISTS	1	00000001

MQCFAC_*

MQCFAC_* (CICS header facility)

Table 368. MQCFAC_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Value
MQCFAC_NONE										8 nulls

MQCFH_* (Command format header structure length)

Table 369. MQCFH_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQCFH_STRUC_LENGTH										36	00000024

MQCFH_* (Command format header version)

Table 370. MQCFH_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQCFH_CURRENT_VERSION										1	00000001
MQCFH_VERSION_1										1	00000001

MQCFIL_* (Command format integer-list parameter structure length)

Table 371. MQCFIL_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQCFIL_STRUC_LENGTH_FIXED										16	00000010

MQCFIN_* (Command format integer parameter structure)

Table 372. MQCFH_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQCFIN_STRUC_LENGTH										16	00000010

MQCFSL_* (Command format string-list parameter structure)

Table 373. MQCFSL_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQCFSL_STRUC_LENGTH_FIXED										24	00000018

MQCFST_*

MQCFST_* (Command format string parameter structure length)

Table 374. MQCFST_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQCFST_STRUC_LENGTH_FIXED										20	00000014

MQCFT_* (Command structure type)

Table 375. MQCFT_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQCFT_COMMAND										1	00000001
MQCFT_RESPONSE										2	00000002
MQCFT_INTEGER										3	00000003
MQCFT_STRING										4	00000004
MQCFT_INTEGER_LIST										5	00000005
MQCFT_STRING_LIST										6	00000006
MQCFT_EVENT										7	00000007
MQCFT_USER										8	00000008

MQCFUNC_* (CICS header function name)

Table 376. MQCFUNC_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Value
MQCFUNC_MQCONN				(1)	X	X	X	X		CONN

Table 376. MQCFUNC_* constants (continued)

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Value
MQCFUNC_MQGET				(1)	X	X	X	X		GETb
MQCFUNC_MQINQ				(1)	X	X	X	X		INQb
MQCFUNC_MQOPEN				(1)	X	X	X	X		OPEN
MQCFUNC_MQPUT				(1)	X	X	X	X		PUTb
MQCFUNC_MQPUT1				(1)	X	X	X	X		PUT1
MQCFUNC_NONE				(1)	X	X	X	X		bbbb
Note:										
1. Supported on AIX, HP-UX, and Sun Solaris only										

MQCGWI_* (CICS header get-wait interval)

Table 377. MQCGWI_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQCGWI_DEFAULT		X	X	X	X	X	X	X	X	-2	FFFFFFFE

MQCHAD_* (Channel auto-definition event reporting)

Table 378. MQCHAD_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQCHAD_DISABLED				(1)	X	X	X	X		0	00000000
MQCHAD_ENABLED				(1)	X	X	X	X		1	00000001

MQCHIDS_*

MQCHIDS_* (Channel indoubt status)

Table 379. MQCHIDS_* constants

Constant										Decimal	Hex
MQCHIDS_NOT_INDOUBT							X	(1)		0	00000000
MQCHIDS_INDOUBT							X	(1)		1	00000001

Note:
1. Supported on Windows 2.1 only

MQCHS_* (Channel status)

Table 380. MQCHS_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMs	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQCHS_INACTIVE							X	(1)		0	00000000
MQCHS_BINDING							X	(1)		1	00000001
MQCHS_STARTING							X	(1)		2	00000002
MQCHS_RUNNING							X	(1)		3	00000003
MQCHS_STOPPING							X	(1)		4	00000004
MQCHS_RETRYING							X	(1)		5	00000005
MQCHS_STOPPED							X	(1)		6	00000006
MQCHS_REQUESTING							X	(1)		7	00000007
MQCHS_PAUSED							X	(1)		8	00000008
MQCHS_INITIALIZING							X	(1)		13	0000000D

Note:
1. Supported on Windows 2.1 only

MQCHSR_* (Channel stop requested)

Table 381. MQCHSR_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQCHSR_STOP_NOT_REQUESTED							X	(1)		0	00000000
MQCHSR_STOP_REQUESTED							X	(1)		1	00000001
Note:											
1. Supported on Windows 2.1 only											

MQCHT_* (Channel type)

Table 382. MQCHT_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQCHT_SENDER							X	(1)		1	00000001
MQCHT_SERVER							X	(1)		2	00000002
MQCHT_RECEIVER							X	(1)		3	00000003
MQCHT_REQUESTER							X	(1)		4	00000004
MQCHT_ALL							X	(1)		5	00000005
MQCHT_CLNTCONN							X	(1)		6	00000006
MQCHT_SRVCNN							X	(1)		7	00000007
MQCHT_CLUSRCVR							X	(1)		8	00000008
MQCHT_CLUSSDR							X	(1)		9	00000009
Note:											
1. Supported on Windows 2.1 only											

MQCHTAB_*

MQCHTAB_* (Channel table)

Table 383. MQCHTAB_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQCHTAB_Q_MGR							X	(1)		1	00000001
MQCHTAB_CLNTCONN							X	(1)		2	00000002
Note: 1. Supported on Windows 2.1 only											

MQCI_* (Correlation identifier)

Table 384. MQCI_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Value
MQCI_NONE										24 nulls
MQCI_NEW_SESSION							X		414D	51214E45575F534553

MQCIH_* (CICS header flags)

Table 385. MQCIH_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQCIH_NONE				(1)	X	X	X	X		0	00000000
Note: 1. Supported on AIX, HP-UX, and Sun Solaris only											

MQCIH_* (CICS header length)

Table 386. MQCIH_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQCIH_LENGTH_1				(1)	X	X	X	X		164	000000A4
MQCIH_LENGTH_2				(1)	X	X	X	X		180	000000B4
Note:											
1. Supported on AIX, HP-UX, and Sun Solaris only											

MQCIH_* (CICS header structure identifier)

Table 387. MQCIH_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Value
MQCIH_STRUC_ID				(1)	X	X	X	X		CIHb
Note:										
1. Supported on AIX, HP-UX, and Sun Solaris only										

MQCIH_* (CICS header version)

Table 388. MQCIH_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQCIH_VERSION_1				(1)	X	X	X	X		1	00000001
MQCIH_VERSION_2				(1)	X	X	X	X		2	00000002
Note:											
1. Supported on AIX, HP-UX, and Sun Solaris only											

MQCLT_*

MQCLT_* (CICS header link type)

Table 389. MQCLT_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQCLT_PROGRAM				(1)	X	X	X	X		1	00000001
MQCLT_TRANSACTION				(1)	X	X	X	X		2	00000002

Note:
1. Supported on AIX, HP-UX, and Sun Solaris only

MQCMD_* (Command identifier)

Table 390. MQCMD_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQCMD_NONE										0	00000000
MQCMD_CHANGE_Q_MGR										1	00000001
MQCMD_INQUIRE_Q_MGR										2	00000002
MQCMD_CHANGE_PROCESS										3	00000003
MQCMD_COPY_PROCESS										4	00000004
MQCMD_CREATE_PROCESS										5	00000005
MQCMD_DELETE_PROCESS										6	00000006
MQCMD_INQUIRE_PROCESS										7	00000007
MQCMD_CHANGE_Q										8	00000008
MQCMD_CLEAR_Q										9	00000009
MQCMD_COPY_Q										10	0000000A
MQCMD_CREATE_Q										11	0000000B
MQCMD_DELETE_Q										12	0000000C
MQCMD_INQUIRE_Q										13	0000000D
MQCMD_RESET_Q_STATS										17	00000011

Table 390. MQCMD_* constants (continued)

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQCMD_INQUIRE_Q_NAMES										18	00000012
MQCMD_INQUIRE_PROCESS_NAMES										19	00000013
MQCMD_INQUIRE_CHANNEL_NAMES										20	00000014
MQCMD_CHANGE_CHANNEL										21	00000015
MQCMD_COPY_CHANNEL										22	00000016
MQCMD_CREATE_CHANNEL										23	00000017
MQCMD_DELETE_CHANNEL										24	00000018
MQCMD_INQUIRE_CHANNEL										25	00000019
MQCMD_PING_CHANNEL										26	0000001A
MQCMD_RESET_CHANNEL										27	0000001B
MQCMD_START_CHANNEL										28	0000001C
MQCMD_STOP_CHANNEL										29	0000001D
MQCMD_START_CHANNEL_INIT										30	0000001E
MQCMD_START_CHANNEL_LISTENER										31	0000001F
MQCMD_CHANGE_NAMELIST										32	00000020
MQCMD_COPY_NAMELIST										33	00000021
MQCMD_CREATE_NAMELIST										34	00000022
MQCMD_DELETE_NAMELIST										35	00000023
MQCMD_INQUIRE_NAMELIST										36	00000024
MQCMD_INQUIRE_NAMELIST_NAMES										37	00000025
MQCMD_ESCAPE										38	00000026
MQCMD_RESOLVE_CHANNEL										39	00000027
MQCMD_PING_Q_MGR										40	00000028
MQCMD_INQUIRE_CHANNEL_STATUS										42	0000002A
MQCMD_Q_MGR_EVENT										44	0000002C
MQCMD_PERFM_EVENT										45	0000002D
MQCMD_CHANNEL_EVENT										46	0000002E
MQCMD_DELETE_PUBLICATION										60	0000003C
MQCMD_DEREGISTER_PUBLISHER										61	0000003D

MQCMD_*

Table 390. MQCMD_* constants (continued)

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQCMD_DEREGISTER_SUBSCRIBER										62	0000003E
MQCMD_PUBLISH										63	0000003F
MQCMD_REGISTER_PUBLISHER										64	00000040
MQCMD_REGISTER_SUBSCRIBER										65	00000041
MQCMD_REQUEST_UPDATE										66	00000042
MQCMD_BROKER_INTERNAL										67	00000043
MQCMD_INQUIRE_Q_CLUSTER_MANAGER										70	00000046
MQCMD_RESUME_Q_MGR_CLUSTER										71	00000047
MQCMD_SUSPEND_Q_MGR_CLUSTER										72	00000048
MQCMD_REFRESH_CLUSTER										73	00000049
MQCMD_RESET_CLUSTER										74	0000004A

MQCMDL_* (Command level)

Table 391. MQCNO_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQCMDL_LEVEL_1	X	X	X	X	X	X	X	X	X	100	00000064
MQCMDL_LEVEL_101					X	X	X		X	101	00000065
MQCMDL_LEVEL_110	X	X	X	X	X	X	X		X	110	0000006E
MQCMDL_LEVEL_114		X	X	X	X	X	X	X	X	114	00000072
MQCMDL_LEVEL_120		X	X	X	X	X	X	X	X	120	00000078
MQCMDL_LEVEL_200	X	X	X	X	X	X	X	X		200	000000C8
MQCMDL_LEVEL_201	X		X	X	X	X	X	X	X	201	000000C9
MQCMDL_LEVEL_210		X	X	X	X	X	X	X	X	210	000000D2
MQCMDL_LEVEL_220		X	X		X		X	X	X	220	000000DC

Table 391. MQCNO_* constants (continued)

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQCMDL_LEVEL_221	X	X	X			X	X	X	X	221	000000DD
MQCMDL_LEVEL_320	X	X		X	X	X	X	X	X	320	00000140
MQCMDL_LEVEL_420	X	X		X	X	X	X	X	X	420	000001A4
MQCMDL_LEVEL_500	X		X		X	X	X	X		500	000001F4
MQCMDL_LEVEL_510	X				X	X	X	X		510	000001FE
MQCMDL_LEVEL_520	X				X	X	X	X		520	00000208

MQCNO_* (Connect options)

Table 392. MQCNO_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQCNO_NONE				(1)	X	X	X	X		0	00000000
MQCNO_STANDARD_BINDING				(1)	X	X	X	X		0	00000000
MQCNO_FASTPATH_BINDING				(1)	X	X	X	X		1	00000001
MQCNO_SERIALIZE_CONN_TAG_Q_MGR		X	X	X	X	X	X	X	X	2	00000002
MQCNO_SERIALIZE_CONN_TAG_QSG		X	X	X	X	X	X	X	X	4	00000004
MQCNO_RESTRICT_CONN_TAG_Q_MGR		X	X	X	X	X	X	X	X	8	00000008
MQCNO_RESTRICT_CONN_TAG_QSG		X	X	X	X	X	X	X	X >	16	00000010
Note:											
1. Supported on AIX, HP-UX, and Sun Solaris only											

MQCNO_*

MQCNO_* (Connect options structure identifier)

Table 393. MQCNO_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Value
MQCNO_STRUC_ID				(1)	X	X	X	X		CNOB

Note:

1. Supported on AIX, HP-UX, and Sun Solaris only

MQCNO_* (Connect options version)

Table 394. MQCNO_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQCNO_VERSION_1				(1)	X	X	X	X		1	00000001
MQCNO_VERSION_2				(1)	X	X	X			2	00000002
MQCNO_VERSION_3		X	X	X	X	X	X	X	X	3	00000003
MQCNO_CURRENT_VERSION				(1)	X	X	X				(variable)

Note:

1. Supported on AIX, HP-UX, and Sun Solaris only

MQCO_* (Close options)

Table 395. MQCO_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQCO_NONE										0	00000000
MQCO_DELETE							X	X		1	00000001

Table 395. MQCO_* constants (continued)

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQCO_DELETE_PURGE							X	X		2	00000002

MQCODL_* (CICS header output data length)

Table 396. MQCODL_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQCODL_AS_INPUT				(1)	X	X	X	X		-1	FFFFFFF

Note:

- Supported on AIX, HP-UX, and Sun Solaris only

MQCQT_* (Cluster queue type)

Table 397. MQCQT_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQCQT_LOCAL_Q				(1)	X	X	X	X		1	00000001
MQCQT_ALIAS_Q				(1)	X	X	X	X		2	00000002
MQCQT_REMOTE_Q				(1)	X	X	X	X		3	00000003
MQCQT_Q_MGR_ALIAS				(1)	X	X	X	X		4	00000004

Note:

- Supported on AIX, HP-UX, and Sun Solaris only

MQCRC_*

MQCRC_* (CICS header return code)

Table 398. MQCRC_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQCRC_OK				(1)	X	X	X	X		0	00000000
MQCRC_CICS_EXEC_ERROR				(1)	X	X	X	X		1	00000001
MQCRC_MQ_API_ERROR				(1)	X	X	X	X		2	00000002
MQCRC_BRIDGE_ERROR				(1)	X	X	X	X		3	00000003
MQCRC_BRIDGE_ABEND				(1)	X	X	X	X		4	00000004
MQCRC_APPLICATION_ABEND				(1)	X	X	X	X		5	00000005
MQCRC_SECURITY_ERROR				(1)	X	X	X	X		6	00000006
MQCRC_PROGRAM_NOT_AVAILABLE				(1)	X	X	X	X		7	00000007
MQCRC_BRIDGE_TIMEOUT				(1)	X	X	X	X		8	00000008
MQCRC_TRANSID_NOT_AVAILABLE				(1)	X	X	X	X		9	00000009
Note: 1. Supported on AIX, HP-UX, and Sun Solaris only											

MQCSC_* (CICS header transaction start code)

Table 399. MQCSC_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Value
MQCSC_START										Sbbb
MQCSC_STARTDATA										SDbb
MQCSC_TERMINPUT										TDbb
MQCSC_NONE										bbbb

MQCT_* (Connection tag)

Table 400. MQCT_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQCT_NONE										128	nulls

MQCTES_* (CICS header task end status)

Table 401. MQCTES_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQCTES_NOSYNC										0	00000000
MQCTES_COMMIT										256	00000100
MQCTES_BACKOUT										4352	00001100
MQCTES_ENDTASK										65536	00010000

MQCUOW_* (CICS header unit-of-work control)

Table 402. MQCUOW_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQCUOWC_MIDDLE				(1)	X	X	X	X		16	00000016
MQCUOWC_FIRST				(1)	X	X	X	X		17	00000011
MQCUOWC_COMMIT				(1)	X	X	X	X		256	00000100
MQCUOWC_LAST				(1)	X	X	X	X		272	00000110
MQCUOWC_ONLY				(1)	X	X	X	X		273	00000111

MQCUOW_*

Table 402. MQCUOW_* constants (continued)

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQCUOWC_BACKOUT				(1)	X	X	X	X		4352	00001100
MQCUOWC_CONTINUE				(1)	X	X	X	X		65536	00010000
Note: 1. Supported on AIX, HP-UX, and Sun Solaris only											

MQCXP_* (Channel-exit parameter structure version)

Table 403. MQCXP_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQCXP_VERSION_1										1	00000001
MQCXP_VERSION_2										2	00000002
MQCXP_VERSION_3										3	00000003
MQCXP_VERSION_4										4	00000004
MQCXP_CURRENT_VERSION										(variable)	

MQDCC_* (Convert-characters masks and factors)

Table 404. MQDCC_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQDCC_SOURCE_ENC_FACTOR				(1)	X	X	X	X		16	00000010
MQDCC_SOURCE_ENC_MASK				(1)	X	X	X	X		240	000000F0
MQDCC_TARGET_ENC_FACTOR				(1)	X	X	X	X		256	00000100

Table 404. MQDCC_* constants (continued)

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQDCC_TARGET_ENC_MASK				(1)	X	X	X	X		3840	00000F00
Note:											
1. Supported on Windows 2.1 only											

MQDCC_* (Convert-characters option)

Table 405. MQDCC_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQDCC_NONE							X	X		0	00000000
MQDCC_SOURCE_ENC_UNDEFINED							X	X		0	00000000
MQDCC_TARGET_ENC_UNDEFINED							X	X		0	00000000
MQDCC_DEFAULT_CONVERSION				(1)	X	X	X	X		1	00000001
MQDCC_FILL_TARGET_BUFFER				(1)	X	X	X	X		2	00000002
MQDCC_SOURCE_ENC_NORMAL							X	X		16	00000010
MQDCC_SOURCE_ENC_REVERSED							X	X		32	00000020
MQDCC_TARGET_ENC_NORMAL							X	X		256	00000100
MQDCC_TARGET_ENC_REVERSED							X	X		512	00000200
MQDCC_SOURCE_ENC_NATIVE							X	X			
MQDCC_TARGET_ENC_NATIVE							X	X			
Note:											
1. Supported on Windows 2.1 only											

MQDELO_*

MQDELO_* (Delete options)

Table 406. MQDELO_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQDELO_NONE										0	00000000
MQDELO_LOCAL										4	00000004

MQDH_* (Distribution header structure identifier)

Table 407. MQDH_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Value
MQDH_STRUC_ID	X			(1)	X	X	X	X		DHbb
Note: 1. Supported on AIX, HP-UX, and Sun Solaris only										

MQDH_* (Distribution header version)

Table 408. MQDH_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQDH_VERSION_1	X			(1)	X	X	X	X		1	00000001
Note: 1. Supported on AIX, HP-UX, and Sun Solaris only											

MQDHF_* (Distribution header flags)

Table 409. MQDHF_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQDHF_NONE	X			(1)	X	X	X	X		0	00000000
MQDHF_NEW_MSG_IDS	X			(1)	X	X	X	X		1	00000001
Note: 1. Supported on AIX, HP-UX, and Sun Solaris only											

MQDL_* (Distribution list support)

Table 410. MQDL_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQDL_NOT_SUPPORTED	X			(1)	X	X	X	X		0	00000000
MQDL_SUPPORTED	X			(1)	X	X	X	X		1	00000001
Note: 1. Supported on AIX, HP-UX, and Sun Solaris only											

MQDLH_* (Dead-letter header structure identifier)

Table 411. MQDLH_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQDLH_STRUC_ID								X			DLHb

MQDLH_*

MQDLH_* (Dead-letter header structure version)

Table 412. MQDLH_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQDLH_VERSION_1								X		1	00000001

MQDT_* (Destination type)

Table 413. MQDT_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQDT_APPL										1	00000001
MQDT_BROKER										2	00000002

MQDXP_* (Data-conversion exit identifier)

Table 414. MQDXP_* constants

Constant	Value
MQDXP_STRUC_ID	DXPb

MQDXP_* (Data-conversion exit version)

Table 415. MQDXP_* constants

Constant	Decimal	Hex
MQDXP_VERSION_1	1	00000001

MQEC_* (Signal event control block completion code)

Table 416. MQEC_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQEC_MSG_ARRIVED		X	X	X	X	X	X	(1)	X	2	00000002
MQEC_WAIT_INTERVAL_EXPIRED		X	X	X	X	X	X	(1)	X	3	00000003
MQEC_WAIT_CANCELED		X	X	X	X	X	X	(1)	X	4	00000004
MQEC_Q_MGR QUIESCING		X	X	X	X	X	X	X	X	5	00000005
MQEC_CONNECTION QUIESCING		X	X	X	X	X	X	X	X	6	00000006
Note:											
1. Supported on Windows Version 2.1 only											

MQEI_* (Expiry interval)

Table 417. MQEI_* constants

Constant	Decimal	Hex
MQEI_UNLIMITED	-1	FFFFFFFF

MQENC_* (Encoding)

Table 418. MQENC_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQENC_NATIVE	X	(1)	X	X	X	X	X	(1)	(1)	17	00000011
	X	X						X	X	273	00000111
	X	(2)	X	X	X	X	X	(2)	(2)	546	00000222
	X	X	X	X	X	X	X	X	X	785	00000311
Notes::											
1. Applies to COBOL only											
2. Applies to C programs and also to DOS and Windows clients											

MQENC_*

MQENC_* (Encoding mask)

Table 419. MQENC_* constants

Constant		Decimal	Hex
MQENC_INTEGER_MASK		15	0000000F
MQENC_DECIMAL_MASK		240	000000F0
MQENC_FLOAT_MASK		3840	00000F00
MQENC_RESERVED_MASK		-4096	FFFFFF00

MQENC_* (Encoding for packed-decimal integers)

Table 420. MQENC_* constants

Constant		Decimal	Hex
MQENC_DECIMAL_UNDEFINED		0	00000000
MQENC_DECIMAL_NORMAL		16	00000010
MQENC_DECIMAL_REVERSED		32	00000020

MQENC_* (Encoding for floating-point numbers)

Table 421. MQENC_* constants

Constant		Decimal	Hex
MQENC_FLOAT_UNDEFINED		0	00000000
MQENC_FLOAT_IEEE_NORMAL		256	00000100
MQENC_FLOAT_IEEE_REVERSED		512	00000200
MQENC_FLOAT_S390		768	00000300

MQENC_* (Encoding for binary integers)

Table 422. MQENC_* constants

Constant		Decimal	Hex
MQENC_INTEGER_UNDEFINED		0	00000000
MQENC_INTEGER_NORMAL		1	00000001
MQENC_INTEGER_REVERSED		2	00000002

MQET_* (Escape type)

Table 423. MQET_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQET_MQSC	X									1	00000001

MQEVR_* (Event reporting)

Table 424. MQEVR_* constants

Constant	Decimal	Hex
MQEVR_DISABLED	0	00000000
MQEVR_ENABLED	1	00000001

MQFB_* (Feedback)

Table 425. MQFB_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQFB_NONE										0	00000000
MQFB_SYSTEM_FIRST							X			1	00000001
MQFB_QUIT							X			256	00000100
MQFB_EXPIRATION							X			258	00000102
MQFB_COA										259	00000103
MQFB_COD										260	00000104
MQFB_CHANNEL_COMPLETED										262	00000106
MQFB_CHANNEL_FAIL_RETRY							X			263	00000107
MQFB_CHANNEL_FAIL							X			264	00000108
MQFB_APPL_CANNOT_BE_STARTED							X			265	00000109
MQFB_TM_ERROR							X	X		266	0000010A

MQFB_*

Table 425. MQFB_* constants (continued)

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQFB_APPL_TYPE_ERROR							X			267	0000010B
MQFB_STOPPED_BY_MSG_EXIT							X			268	0000010C
MQFB_XMIT_Q_MSG_ERROR							X			271	0000010F
MQFB_PAN				(1)	X	X	X	X		275	0000010G
MQFB_NAN				(1)	X	X	X	X		276	0000010H
MQFB_STOPPED_BY_CHAD_EXIT				(1)	X	X	X	X		277	00000000
MQFB_DATA_LENGTH_ZERO				(1)	X	X	X	X		291	00000123
MQFB_DATA_LENGTH_NEGATIVE				(1)	X	X	X	X		292	00000124
MQFB_DATA_LENGTH_TOO_BIG				(1)	X	X	X	X		293	00000125
MQFB_BUFFER_OVERFLOW				(1)	X	X	X	X		294	00000126
MQFB_LENGTH_OFF_BY_ONE				(1)	X	X	X	X		295	00000127
MQFB_IIH_ERROR				(1)	X	X	X	X		296	00000128
MQFB_NOT_AUTHORIZED_FOR_IMS				(1)	X	X	X	X		298	0000012A
MQFB_IMS_ERROR				(1)	X	X	X	X		300	0000012C
MQFB_IMS_FIRST				(1)	X	X	X	X		301	0000012D
MQFB_IMS_LAST				(1)	X	X	X	X		399	0000018F
MQFB_CICS_INTERNAL_ERROR				(1)	X	X	X	X		401	00000191
MQFB_CICS_NOT_AUTHORIZED				(1)	X	X	X	X		402	00000192
MQFB_CICS_BRIDGE_FAILURE				(1)	X	X	X	X		403	00000193
MQFB_CICS_CORREL_ID_ERROR				(1)	X	X	X	X		404	00000194
MQFB_CICS_CCSID_ERROR				(1)	X	X	X	X		405	00000195
MQFB_CICS_ENCODING_ERROR				(1)	X	X	X	X		406	00000196
MQFB_CICS_CIH_ERROR				(1)	X	X	X	X		407	00000197
MQFB_CICS_UOW_ERROR				(1)	X	X	X	X		408	00000198
MQFB_CICS_COMMAREA_ERROR				(1)	X	X	X	X		409	00000199
MQFB_CICS_APPL_NOT_STARTED				(1)	X	X	X	X		410	0000019A
MQFB_CICS_APPL_ABENDED				(1)	X	X	X	X		411	0000019B
MQFB_CICS_DLQ_ERROR				(1)	X	X	X	X		412	0000019C
MQFB_CICS_UOW_BACKED_OUT				(1)	X	X	X	X		413	0000019D

Table 425. MQFB_* constants (continued)

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQFB_SYSTEM_LAST										65535	0000FFFF
MQFB_APPL_FIRST										65536	00010000
MQFB_APPL_LAST										999999999	3B9AC9FF
Note:											
1. Supported on AIX, HP-UX, and Sun Solaris only											

MQFC_* (Force option)

Table 426. MQFC_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQFC_NO										0	00000000
MQFC_YES										1	00000001

MQFMT_* (Format)

Table 427. MQFMT_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Value
MQFMT_NONE							X			bbbbbbbb
MQFMT_ADMIN							X			MQADMINb
MQFMT_CHANNEL_COMPLETED							X			MQCHCOMb
MQFMT_CICS				(1)	X	X	X	X		MQCICSbb
MQFMT_COMMAND_1				(1)			X	X		MQCMD1bb

MQFMT_*

Table 427. MQFMT_* constants (continued)

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Value
MQFMT_COMMAND_2							X			MQCMD2bb
MQFMT_DEAD_LETTER_HEADER							X	X		MQDEADbb
MQFMT_DIST_HEADER	X			(1)	X	X	X	X		MQDISTbb
MQFMT_EVENT				(1)			X			MQEVENTb
MQFMT_IMS				(1)	X	X	X	X		MQIMSbbb
MQFMT_IMS_VAR_STRING				(1)	X	X	X	X		MQIMSVSb
MQFMT_MD_EXTENSION	X			(1)	X	X	X	X		MQHMDEbb
MQFMT_PCF							X			MQPCFbbb
MQFMT_REF_MSG_HEADER				(1)	X	X	X	X		MQHREFbb
MQFMT_STRING							X			MQSTRbbb
MQFMT_TRIGGER								X		MQTRIGbb
MQFMT_XMIT_Q_HEADER							X			MQXMITbb
MQFMT_WORK_INFO_HEADER				(1)	X	X	X	X		MQHWIHbb
Note:										
1. Supported on AIX, HP-UX, and Sun Solaris only										

MQGI_* (Group identifier)

Table 428. MQGI_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Value
MQGI_NONE	X			(1)	X	X	X	X		24 nulls
Note:										
1. Supported on AIX, HP-UX, and Sun Solaris only										

MQGMO_* (Get-message options)

Table 429. MQGMO_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQGMO_NO_WAIT										0	00000000
MQGMO_NONE										0	00000000
MQGMO_WAIT										1	00000001
MQGMO_SYNCPOINT										2	00000002
MQGMO_NO_SYNCPOINT							X			4	00000004
MQGMO_SET_SIGNAL		X	X	X	X		X	(1)	X	8	00000008
MQGMO_BROWSE_FIRST										16	00000010
MQGMO_BROWSE_NEXT										32	00000020
MQGMO_ACCEPT_TRUNCATED_MSG										64	00000040
MQGMO_MARK_SKIP_BACKOUT		X	X	X	X	X	X	X	X	128	00000080
MQGMO_MSG_UNDER_CURSOR										256	00000100
MQGMO_LOCK	X					X	X	X		512	00000200
MQGMO_UNLOCK	X					X	X	X		1024	00000400
MQGMO_BROWSE_MSG_UNDER_CURSOR	X						X			2048	00000800
MQGMO_SYNCPOINT_IF_PERSISTENT				(2)	X	X	X	(1)		4096	00001000
MQGMO_FAIL_IF QUIESCING										8192	00002000
MQGMO_CONVERT								X		16384	00004000
MQGMO_LOGICAL_ORDER	X			(2)	X	X	X	X		32768	00008000
MQGMO_COMPLETE_MSG	X			(2)	X	X	X	X		65536	00010000
MQGMO_ALL_MSGS_AVAILABLE	X			(2)	X	X	X	X		131072	00020000
MQGMO_ALL_SEGMENTS_AVAILABLE	X			(2)	X	X	X	X		262144	00040000

Notes::

1. Supported on Windows Version 2.1 only
2. Supported on AIX, HP-UX, and Sun Solaris only

MQGMO_*

MQGMO_* (Get-message options structure identifier)

Table 430. MQGMO_* constants

Constant	Value
MQGMO_STRUC_ID	GMOb

MQGMO_* (Get-message options version)

Table 431. MQGMO_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQGMO_VERSION_1										1	00000001
MQGMO_VERSION_2	X			(1)	X	X	X	X		2	00000002
MQGMO_VERSION_3				(1)	X	X	X	X		3	00000003
MQGMO_CURRENT_VERSION				(1)	X	X	X	X		(variable)	

Note:
1. Supported on AIX, HP-UX, and Sun Solaris only

MQGS_* (Group status)

Table 432. MQGS_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Value
MQGS_NOT_IN_GROUP	X			(1)	X	X	X	X		b
MQGS_MSG_IN_GROUP	X			(1)	X	X	X	X		G
MQGS_LAST_MSG_IN_GROUP	X			(1)	X	X	X	X		L

Note:
1. Supported on AIX, HP-UX, and Sun Solaris only

MQHC_* (Connect handle)

Table 433. MQHC_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQHC_DEF_HCONN		X		X	X	X	X	X	X	0	00000000
MQHC_UNUSABLE_HCONN	X						X			-1	FFFFFFFF

MQHO_* (Object handle)

Table 434. MQHO_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQHO_UNUSABLE_HOBJ	X						X			-1	FFFFFFFF

MQIA_* (Integer attribute selector)

Table 435. MQIA_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal Hex	
MQIA_FIRST										1	00000001
MQIA_APPL_TYPE							X			1	00000001
MQIA_CODED_CHAR_SET_ID							X			2	00000002
MQIA_CURRENT_Q_DEPTH										3	00000003
MQIA_DEF_INPUT_OPEN_OPTION							X			4	00000004
MQIA_DEF_PERSISTENCE										5	00000005
MQIA_DEF_PRIORITY							X			6	00000006

MQIA_*

Table 435. MQIA_* constants (continued)

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal Hex
MQIA_DEFINITION_TYPE										7 00000007
MQIA_HARDEN_GET_BACKOUT							X			8 00000008
MQIA_INHIBIT_GET										9 00000009
MQIA_INHIBIT_PUT										10 0000000A
MQIA_MAX_HANDLES							X			11 0000000B
MQIA_USAGE							X			12 0000000C
MQIA_MAX_MSG_LENGTH										13 0000000D
MQIA_MAX_PRIORITY							X			14 0000000E
MQIA_MAX_Q_DEPTH										15 0000000F
MQIA_MSG_DELIVERY_SEQUENCE							X			16 00000010
MQIA_OPEN_INPUT_COUNT										17 00000011
MQIA_OPEN_OUTPUT_COUNT										18 00000012
MQIA_NAME_COUNT				(1)	X	X	X	X		19 00000013
MQIA_Q_TYPE										20 00000014
MQIA_RETENTION_INTERVAL							X			21 00000015
MQIA_BACKOUT_THRESHOLD							X			22 00000016
MQIA_SHAREABILITY										23 00000017
MQIA_TRIGGER_CONTROL								X		24 00000018
MQIA_TRIGGER_INTERVAL							X	X		25 00000019
MQIA_TRIGGER_MSG_PRIORITY							X	X		26 0000001A
MQIA_TRIGGER_TYPE								X		28 0000001C
MQIA_TRIGGER_DEPTH							X	X		29 0000001D
MQIA_SYNCPOINT										30 0000001E
MQIA_COMMAND_LEVEL							X			31 0000001F
MQIA_PLATFORM							X			32 00000020
MQIA_MAX_UNCOMMITTED_MSGS							X			33 00000021
MQIA_DIST_LISTS	X			(1)	X	X	X	X		34 00000022
MQIA_TIME_SINCE_RESET	X						X	(2)		35 00000023
MQIA_HIGH_Q_DEPTH	X						X	(2)		36 00000024

Table 435. MQIA_* constants (continued)

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal Hex
MQIA_MSG_ENQ_COUNT	X						X (2)			37 00000025
MQIA_MSG_DEQ_COUNT	X						X (2)			38 00000026
MQIA_Q_DEPTH_HIGH_LIMIT	X						X (2)			40 00000028
MQIA_Q_DEPTH_LOW_LIMIT	X						X (2)			41 00000029
MQIA_Q_DEPTH_MAX_EVENT	X						X (2)			42 0000002A
MQIA_Q_DEPTH_HIGH_EVENT	X						X (2)			43 0000002B
MQIA_Q_DEPTH_LOW_EVENT	X						X (2)			44 0000002C
MQIA_SCOPE	X		X			X	X X			45 0000002D
MQIA_Q_SERVICE_INTERVAL_EVENT	X						X (2)			46 0000002E
MQIA_AUTHORITY_EVENT	X						X (2)			47 0000002F
MQIA_INHIBIT_EVENT	X						X (2)			48 00000030
MQIA_LOCAL_EVENT	X						X (2)			49 00000031
MQIA_REMOTE_EVENT	X						X (2)			50 00000032
MQIA_START_STOP_EVENT	X						X (2)			52 00000034
MQIA_PERFORMANCE_EVENT	X						X (2)			53 00000035
MQIA_Q_SERVICE_INTERVAL	X						X (2)			54 00000036
MQIA_CHANNEL_AUTO_DEF				(1)	X	X	X X			55 00000037
MQIA_CHANNEL_AUTO_DEF_EVENT	X			(1)	X	X	X X			56 00000038
MQIA_INDEX_TYPE		X	X	X	X	X	X X	X	X	57 00000039
MQIA_CLUSTER_WORKLOAD_LENGTH				(1)	X	X	X X			58 0000003A
MQIA_CLUSTER_Q_TYPE				(1)	X	X	X X			59 0000003B
MQIA_DEF_BIND				(1)	X	X	X X			61 0000003D
MQIA_QSG_DISP		X	X	X	X	X	X X	X	X	63 0000003F
MQIA_INTRA_GROUP_QUEUEING		X	X	X	X	X	X X	X	X	64 00000040
MQIA_LAST_USED							X			(variable)
MQIA_LAST										2000 000007D0

Notes::

1. Supported on AIX, HP-UX, and Sun Solaris only
2. Supported on Windows Version 2.1 only

MQIACF_*

MQIACF_* (Integer attribute command format parameter)

Table 436. MQIACF_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal Hex
MQIACF_FIRST							X (1)			1001 000003E9
MQIACF_Q_MGR_ATTRS							X (1)			1001 000003E9
MQIACF_Q_ATTRS							X (1)			1002 000003EA
MQIACF_PROCESS_ATTRS							X (1)			1003 000003EB
MQIACF_NAMELIST_ATTRS							X (1)			1004 000003EC
MQIACF_FORCE							X (1)			1005 000003ED
MQIACF_REPLACE							X (1)			1006 000003EE
MQIACF_PURGE							X (1)			1007 000003EF
MQIACF_QUIESCE							X (1)			1008 000003F0
MQIACF_ALL							X (1)			1009 000003F1
MQIACF_PARAMETER_ID							X (1)			1012 000003F4
MQIACF_ERROR_ID							X (1)			1013 000003F5
MQIACF_ERROR_IDENTIFIER							X (1)			1013 000003F5
MQIACF_SELECTOR							X (1)			1014 000003F6
MQIACF_CHANNEL_ATTRS							X (1)			1015 000003F7
MQIACF_ESCAPE_TYPE							X (1)			1017 000003F9
MQIACF_ERROR_OFFSET							X (1)			1018 000003FA
MQIACF_REASON_QUALIFIER							X (1)			1020 000003FC
MQIACF_COMMAND							X (1)			1021 000003FD
MQIACF_OPEN_OPTIONS							X (1)			1022 000003FE
MQIACF_AUX_ERROR_DATA_INT_1							X (1)			1070 0000042E
MQIACF_AUX_ERROR_DATA_INT_2							X (1)			1071 0000042F
MQIACF_CONV_REASON_CODE							X (1)			1072 00000430
MQIACF_BRIDGE_TYPE							X (1)			1073 00000431
MQIACF_INQUIRY							X (1)			1074 00000432
MQIACF_WAIT_INTERVAL							X (1)			1075 00000433
MQIACF_OPTIONS							X (1)			1076 00000434

Table 436. MQIACF_* constants (continued)

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal Hex
MQIACF_BROKER_OPTIONS							X	(1)		1077 00000435
MQIACF_SEQUENCE_NUMBER							X	(1)		1079 00000437
MQIACF_INTEGER_DATA							X	(1)		1080 00000438
MQIACF_REGISTRATION_OPTIONS							X	(1)		1081 00000439
MQIACF_PUBLICATION_OPTIONS							X	(1)		1082 0000043A
MQIACF_CLUSTER_INFO							X	(1)		1083 0000043B
MQIACF_Q_MGR_DEFINITION_TYPE							X	(1)		1084 0000043C
MQIACF_Q_MGR_TYPE							X	(1)		1085 0000043D
MQIACF_ACTION							X	(1)		1086 0000043E
MQIACF_SUSPEND							X	(1)		1087 0000043F
MQIACF_BROKER_COUNT							X	(1)		1088 00000440
MQIACF_APPL_COUNT							X	(1)		1089 00000441
MQIACF_ANONYMOUS_COUNT							X	(1)		1090 00000442
MQIACF_REG_REG_OPTIONS							X	(1)		1091 00000443
MQIACF_DELETE_OPTIONS							X	(1)		1092 00000444
MQIACF_CLUSTER_Q_MGR_ATTRS							X	(1)		1092 00000444
MQIACF_LAST_USED							X	(1)		(variable)
Note:										
1. Supported on Windows 2.1 only										

MQIACH_* (Channel integer attribute command format parameter)

Table 437. MQIACH_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal Hex
MQIACH_FIRST							X	(1)		1501 000005DD
MQIACH_XMIT_PROTOCOL_TYPE							X	(1)		1501 000005DD

MQIACH_*

Table 437. MQIACH_* constants (continued)

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal Hex
MQIACH_BATCH_SIZE							X	(1)		1502 000003EA
MQIACH_DISC_INTERVAL							X	(1)		1503 000005DF
MQIACH_SHORT_TIMER							X	(1)		1504 000005E0
MQIACH_SHORT_RETRY							X	(1)		1505 000005E1
MQIACH_LONG_TIMER							X	(1)		1506 000005E2
MQIACH_LONG_RETRY							X	(1)		1507 000005E3
MQIACH_PUT_AUTHORITY							X	(1)		1508 000005E4
MQIACH_SEQUENCE_NUMBER_WRAP							X	(1)		1509 000005E5
MQIACH_MAX_MSG_LENGTH							X	(1)		1510 000005E6
MQIACH_CHANNEL_TYPE							X	(1)		1511 000005E7
MQIACH_DATA_COUNT							X	(1)		1512 000005E8
MQIACH_MSG_SEQUENCE_NUMBER							X	(1)		1514 000005EA
MQIACH_DATA_CONVERSION							X	(1)		1515 000005EB
MQIACH_IN_DOUBT							X	(1)		1516 000005EC
MQIACH_MCA_TYPE							X	(1)		1517 000005ED
MQIACH_CHANNEL_INSTANCE_TYPE							X	(1)		1523 000005F3
MQIACH_CHANNEL_INSTANCE_ATTRS							X	(1)		1524 000005F4
MQIACH_CHANNEL_ERROR_DATA							X	(1)		1525 000005F5
MQIACH_CHANNEL_TABLE							X	(1)		1526 000005F6
MQIACH_CHANNEL_STATUS							X	(1)		1527 000005F7
MQIACH_INDOUBT_STATUS							X	(1)		1528 000005F8
MQIACH_LAST_SEQ_NUMBER							X	(1)		1529 000005F9
MQIACH_CURRENT_MSGS							X	(1)		1531 000005FB
MQIACH_CURRENT_SEQ_NUMBER							X	(1)		1532 000005FC
MQIACH_MSGS							X	(1)		1534 000005FE
MQIACH_BYTES_SENT							X	(1)		1535 000005FF
MQIACH_BYTES_RCVD							X	(1)		1536 00000600
MQIACH_BATCHES							X	(1)		1537 00000601
MQIACH_BUFFERS_SENT							X	(1)		1538 00000602

Table 437. MQIACH_* constants (continued)

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal Hex
MQIACH_BUFFERS_RCVD							X	(1)		1539 00000603
MQIACH_LONG_RETRIES_LEFT							X	(1)		1540 00000604
MQIACH_SHORT_RETRIES_LEFT							X	(1)		1541 00000605
MQIACH_MCA_STATUS							X	(1)		1542 00000606
MQIACH_STOP_REQUESTED							X	(1)		1543 00000607
MQIACH_MR_COUNT							X	(1)		1544 00000608
MQIACH_MR_INTERVAL							X	(1)		1545 00000609
MQIACH_NPM_SPEED							X	(1)		1562 0000061A
MQIACH_HB_INTERVAL							X	(1)		1563 0000061B
MQIACH_BATCH_INTERVAL							X	(1)		1564 0000061C
MQIACH_NETWORK_PRIORITY							X	(1)		1565 0000061D
MQIACH_LAST_USED							X	(1)		(variable)
Note:										
1. Supported on Windows 2.1 only										

MQIAUT_* (IMS authenticator)

Table 438. MQIAUT_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Value
MQIAUT_NONE		X	X	X	X	X	X	X	X	bbbbbbbb

MQIAV_* (Integer attribute value)

Table 439. MQIAV_* constants

Constant	Decimal	Hex
MQIAV_NOT_APPLICABLE	-1	FFFFFFF

MQIAV_*

Table 439. MQIAV_* constants (continued)

Constant	Decimal	Hex
MQIAV_UNDEFINED	-2	FFFFFFFE

MQICM_* (IMS commit mode)

Table 440. MQICM_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Value
MQICM_COMMIT_THEN_SEND				(1)	X	X	X	X		0
MQICM_SEND_THEN_COMMIT				(1)	X	X	X	X		1

Note:
1. Supported on AIX, HP-UX, and Sun Solaris only

MQIDO_* (Indoubt resolution)

Table 441. MQIDO_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQIDO_COMMIT										1	00000001
MQIDO_BACKOUT										2	00000002

MQIGQ_* (Intra-group queuing)

Table 442. MQIGQ_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQIGQ_DISABLED										0	00000000
MQIGQ_ENABLED										1	00000001

MQIIH_* (IMS header flags)

Table 443. MQIIH_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQIIH_NONE				(1)	X	X	X	X		0	00000000

Note:
1. Supported on AIX, HP-UX, and Sun Solaris only

MQIIH_* (IMS header length)

Table 444. MQIIH_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQIIH_LENGTH_1				(1)	X	X	X	X		84	00000054

Note:
1. Supported on AIX, HP-UX, and Sun Solaris only

MQIIH_*

MQIIH_* (IMS header structure identifier)

Table 445. MQIIH_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Value
MQIIH_STRUC_ID				(1)	X	X	X	X		IIHb
Note: 1. Supported on AIX, HP-UX, and Sun Solaris only										

MQIIH_* (IMS header version)

Table 446. MQIIH_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQIIH_VERSION_1				(1)	X	X	X	X		1	00000001
Note: 1. Supported on AIX, HP-UX, and Sun Solaris only											

MQISS_* (IMS security scope)

Table 447. MQISS_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Value
MQISS_CHECK				(1)	X	X	X	X		C
MQISS_FULL				(1)	X	X	X	X		F
Note: 1. Supported on AIX, HP-UX, and Sun Solaris only										

MQIT_* (Index type)

Table 448. MQIT_* constants

Constant		OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQIT_NONE			X	X	X	X	X	X	X	X	0	00000000
MQIT_MSG_ID			X	X	X	X	X	X	X	X	1	00000001
MQIT_CORREL_ID			X	X	X	X	X	X	X	X	2	00000002
MQIT_MSG_TOKEN			X	X	X	X	X	X	X	X	4	00000004

MQITII_* (IMS transaction instance identifier)

Table 449. MQITII_* constants

Constant		OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Value
MQITII_NONE					(1)	X	X	X	X		16 nulls
Note:											
1. Supported on AIX, HP-UX, and Sun Solaris only											

MQITS_* (IMS transaction state)

Table 450. MQITS_* constants

Constant		OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Value
MQITS_IN_CONVERSATION					(1)	X	X	X	X		C
MQITS_NOT_IN_CONVERSATION					(1)	X	X	X	X		b
Note:											
1. Supported on AIX, HP-UX, and Sun Solaris only											

MQMCAS_*

MQMCAS_* (MCA status)

Table 451. MQMCAS_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQMCAS_STOPPED										0	00000000
MQMCAS_RUNNING										3	00000003

MQMCAT_* (MCA type)

Table 452. MQMCAT_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQMCAT_PROCESS										1	00000001
MQMCAT_THREAD										2	00000002

MQMD_* (Message descriptor structure identifier)

Table 453. MQMD_* constants

Constant	Value
MQMD_STRUC_ID	Mdbb

MQMD_* (Message descriptor version)

Table 454. MQMD_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQMD_VERSION_1										1	00000001
MQMD_VERSION_2	X			(1)	X	X	X	X		2	00000002
Note:											
1. Supported on AIX, HP-UX, and Sun Solaris only											

MQMDE_* (Message descriptor extension length)

Table 455. MQMDE_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQMDE_LENGTH_2	X			(1)	X	X	X	X		72	00000048

MQMDE_* (Message descriptor extension structure identifier)

Table 456. MQMDE_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Value
MQMDE_STRUC_ID	X			(1)	X	X	X	X		MDEb
Note:										
1. Supported on AIX, HP-UX, and Sun Solaris only										

MQMDE_*

MQMDE_* (Message descriptor extension version)

Table 457. MQMDE_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQMDE_VERSION_2	X			(1)	X	X	X	X		2	00000002

Note:

1. Supported on AIX, HP-UX, and Sun Solaris only

MQMDEF_* (Message descriptor extension flags)

Table 458. MQMDEF_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQMDEF_NONE	X			(1)	X	X	X	X		0	00000000

Note:

1. Supported on AIX, HP-UX, and Sun Solaris only

MQMDS_* (Message delivery sequence)

Table 459. MQMDS_* constants

Constant	Decimal	Hex
MQMDS_PRIORITY	0	00000000
MQMDS_FIFO	1	00000001

MQMF_* (Message flags)

Table 460. MQMF_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQMF_SEGMENTATION_INHIBITED	X			(1)	X	X	X	X		0	00000000
MQMF_NONE	X			(1)	X	X	X	X		0	00000000
MQMF_SEGMENTATION_ALLOWED	X			(1)	X	X	X	X		1	00000001
MQMF_SEGMENT	X			(1)	X	X	X	X		2	00000002
MQMF_LAST_SEGMENT	X			(1)	X	X	X	X		4	00000004
MQMF_MSG_IN_GROUP	X			(1)	X	X	X	X		8	00000008
MQMF_LAST_MSG_IN_GROUP	X			(1)	X	X	X	X		16	00000010

Note:
1. Supported on AIX, HP-UX, and Sun Solaris only

MQMF_* (Message flags masks)

Table 461. MQMF_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQMF_ACCEPT_UNSUP_MASK	X			(1)	X	X	X	X		-1048576	FFF00000
MQMF_ACCEPT_UNSUP_IF_XMIT_MASK	X			(1)	X	X	X	X		1044480	000FF000
MQMF_REJECT_UNSUP_MASK	X			(1)	X	X	X	X		4095	00000FFF

Note:
1. Supported on AIX, HP-UX, and Sun Solaris only

MQMI_* (Message identifier)

Table 462. MQMI_* constants

Constant	Value
MQMI_NONE	24 nulls

MQMO_* (Match options)

Table 463. MQMO_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMs	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQMO_NONE				(1)	X	X	X	X		0	00000000
MQMO_MATCH_MSG_ID				(1)	X	X	X	X		1	00000001
MQMO_MATCH_CORREL_ID				(1)	X	X	X	X		2	00000002
MQMO_MATCH_GROUP_ID	X			(1)	X	X	X	X		4	00000004
MQMO_MATCH_MSG_SEQ_NUMBER	X			(1)	X	X	X	X		8	00000008
MQMO_MATCH_OFFSET	X			(1)	X	X	X	X		16	00000010
MQMO_MATCH_MSG_TOKEN		X	X	X	X	X	X	X	X	32	00000020

Note:
1. Supported on AIX, HP-UX, and Sun Solaris only

MQMT_* (Message type)

Table 464. MQMT_* constants

Constant	Decimal	Hex
MQMT_SYSTEM_FIRST	1	00000001
MQMT_REQUEST	1	00000001
MQMT_REPLY	2	00000002
MQMT_REPORT	4	00000004
MQMT_DATAGRAM	8	00000008
MQMT_SYSTEM_LAST	65535	0000FFFF
MQMT_APPL_FIRST	65536	00010000
MQMT_APPL_LAST	99999999	3B9AC9FF

MQMTOK_* (Message token)

Table 465. MQMTOK_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Value
	MQMTOK_NONE		X	X	X	X	X	X	X	

MQNC_* (Name count)

Table 466. MQNC_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
	MQNC_MAX_NAMELIST_NAME_COUNT				(1)	X	X	X	X		256
Note:											
1. Supported on AIX, HP-UX, and Sun Solaris only											

MQNPMS_* (Nonpersistent message speed)

Table 467. MQNPMS_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
	MQNPMS_NORMAL				(1)	X	X	X	X		1
MQNPMS_FAST				(1)	X	X	X	X		2	00000002

MQOD_*

MQOD_* (Object descriptor length)

Table 468. MQOD_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQOD_CURRENT_LENGTH				(1)	X	X	X	X			(variable)
Note: 1. Supported on AIX, HP-UX, and Sun Solaris only											

MQOD_* (Object descriptor structure identifier)

Table 469. MQOD_* constants

Constant	Value
MQOD_STRUC_ID	0dbb

MQOD_* (Object descriptor version)

Table 470. MQOD_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQOD_VERSION_1										1	00000001
MQOD_VERSION_2				(1)	X	X	X	X		2	00000002
MQOD_VERSION_3				(1)	X	X	X	X		3	00000003
MQOD_CURRENT_VERSION				(1)	X	X	X	X		3	00000003
Note: 1. Supported on AIX, HP-UX, and Sun Solaris only											

MQOII_* (Object instance identifier)

Table 471. MQOII_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Value
MQOII_NONE				(1)	X	X	X	X		24 nulls

Note:
1. Supported on AIX, HP-UX, and Sun Solaris only

MQOL_* (Original length)

Table 472. MQOL_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQOL_UNDEFINED	X			(1)	X	X	X	X		-1	FFFFFFF

Note:
1. Supported on AIX, HP-UX, and Sun Solaris only

MQOO_* (Open options)

Table 473. MQOO_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Value
MQOO_INPUT_AS_Q_DEF							X			1 00000001
MQOO_INPUT_SHARED										2 00000002
MQOO_INPUT_EXCLUSIVE										4 00000004
MQOO_BROWSE										8 00000008
MQOO_OUTPUT										16 00000010

MQOO_*

Table 473. MQOO_* constants (continued)

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Value	
MQOO_INQUIRE										32	00000020
MQOO_SET										64	00000040
MQOO_SAVE_ALL_CONTEXT							X	X		128	00000080
MQOO_PASS_IDENTITY_CONTEXT							X	X		256	00000100
MQOO_PASS_ALL_CONTEXT							X	X		512	00000200
MQOO_SET_IDENTITY_CONTEXT							X			1024	00000400
MQOO_SET_ALL_CONTEXT							X			2048	00000800
MQOO_ALTERNATE_USER_AUTHORITY							X			4096	00001000
MQOO_FAIL_IF QUIESCING							X			8192	00002000
MQOO_BIND_AS_Q_DEF				(1)	X	X	X	X		0	00000000
MQOO_BIND_ON_OPEN				(1)	X	X	X	X		16384	00004000
MQOO_BIND_NOT_FIXED				(1)	X	X	X	X		32768	00008000
Note:											
1. Supported on AIX, HP-UX, and Sun Solaris only											

MQOT_* (Object type)

Table 474. MQOT_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQOT_Q										1	00000001
MQOT_NAMELIST				(1)	X	X	X	X		2	00000002
MQOT_PROCESS							X			3	00000003
MQOT_Q_MGR							X			5	00000005
MQOT_CHANNEL		X	X	X	X	X	X	X	X	6	00000006
MQOT_RESERVED_1		X	X	X	X	X	X	X	X	7	00000007
MQOT_ALL							X			1001	000003E9

Table 474. MQOT_* constants (continued)

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQOT_ALIAS_Q							X			1002	000003EA
MQOT_MODEL_Q							X			1003	000003EB
MQOT_LOCAL_Q							X			1004	000003EC
MQOT_REMOTE_Q							X			1005	000003ED
MQOT_SENDER_CHANNEL							X			1007	000003EF
MQOT_SERVER_CHANNEL							X			1008	000003F0
MQOT_REQUESTER_CHANNEL							X			1009	000003F1
MQOT_RECEIVER_CHANNEL							X			1010	000003F2
MQOT_CURRENT_CHANNEL							X			1011	000003F3
MQOT_SAVED_CHANNEL							X			1012	000003F4
Note:											
1. Supported on AIX, HP-UX, and Sun Solaris only											

MQPA_* (Put authority)

Table 475. MQPA_* constants

Constant	Decimal	Hex
MQPA_DEFAULT	1	00000001
MQPA_CONTEXT	2	00000002
MQPA_ONLY_MCA	3	00000003
MQPA_ALTERNATE_OR_MCA	4	00000004

MQPER_*

MQPER_* (Persistence)

Table 476. MQPER_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQPER_NOT_PERSISTENT							X			0	00000000
MQPER_PERSISTENT										1	00000001
MQPER_PERSISTENCE_AS_Q_DEF							X			2	00000002

MQPL_* (Platform)

Table 477. MQPL_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQPL_MVS		X	X	X	X	X	X	X	X	1	00000001
MQPL_OS2	X		X	X	X	X	X	X	X	2	00000002
MQPL_OS390		X	X	X	X	X	X	X	X	1	00000001
MQPL_AIX	X	X	X		X	X	X	X	X	3	00000003
MQPL_UNIX	X	X	X		X	X	X	X	X	3	00000003
MQPL_OS400	X	X		X	X	X	X	X	X	4	00000004
MQPL_ Windows	X	X	X	X	X	X	X		X	5	00000005
MQPL_WINDOWS_NT	X	X	X	X	X	X	X	X		11	0000000B
MQPL_VMS	X	X	X	X		X	X	X	X	12	0000000C
MQPL_NSK	X	X	X	X	X		X	X	X	13	0000000D

MQPMO_* (Put-message options)

Table 478. MQPMO_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal Hex
MQPMO_NONE										0 00000000
MQPMO_SYNCPOINT										2 00000002
MQPMO_NO_SYNCPOINT							X			4 00000004
MQPMO_DEFAULT_CONTEXT							X			32 00000020
MQPMO_NEW_MSG_ID	X			(1)	X	X	X	X		64 00000040
MQPMO_NEW_CORREL_ID	X			(1)	X	X	X	X		128 00000080
MQPMO_PASS_IDENTITY_CONTEXT							X	X		256 00000100
MQPMO_PASS_ALL_CONTEXT							X	X		512 00000200
MQPMO_SET_IDENTITY_CONTEXT							X			1024 00000400
MQPMO_SET_ALL_CONTEXT							X			2048 00000800
MQPMO_ALTERNATE_USER_AUTHORITY							X			4096 00001000
MQPMO_FAIL_IF QUIESCING							X			8192 00002000
MQPMO_NO_CONTEXT							X			16384 00004000
MQPMO_LOGICAL_ORDER	X			(1)	X	X	X	X		32768 00008000
Note:										
1. Supported on AIX, HP-UX, and Sun Solaris only										

MQPMO_* (Put-message options structure length)

Table 479. MQPMO_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal Hex
MQPMO_CURRENT_LENGTH				(1)	X	X	X	X		(variable)
Note:										
1. Supported on AIX, HP-UX, and Sun Solaris only										

MQPMO_*

MQPMO_* (Put-message options structure identifier)

Table 480. MQPMO_* constants

Constant	Value
MQPMO_STRUC_ID	PM0b

MQPMO_* (Put-message options structure version)

Table 481. MQPMO_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQPMO_VERSION_1										1	00000001
MQPMO_VERSION_2	X			(1)	X	X	X	X		2	00000002

Note:
1. Supported on AIX, HP-UX, and Sun Solaris only

MQPMRF_* (Put-message record field flags)

Table 482. MQPMRF_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQPMRF_NONE	X			(1)	X	X	X	X		0	00000000
MQPMRF_MSG_ID	X			(1)	X	X	X	X		1	00000001
MQPMRF_CORREL_ID	X			(1)	X	X	X	X		2	00000002
MQPMRF_GROUP_ID	X			(1)	X	X	X	X		4	00000004
MQPMRF_FEEDBACK	X			(1)	X	X	X	X		8	00000008
MQPMRF_ACCOUNTING_TOKEN	X			(1)	X	X	X	X		16	00000010

Note:
1. Supported on AIX, HP-UX, and Sun Solaris only

MQPO_* (Purge option)

Table 483. MQPO_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQPO_NO							X	(1)		0	00000000
MQPO_YES							X	(1)		1	00000001
Note:											
1. Supported on Windows 2.1 only											

MQPRI_* (Priority)

Table 484. MQPRI_* constants

Constant	Decimal	Hex
MQPRI_PRIORITY_AS_Q_DEF	-1	FFFFFFF

MQPS_* (Publish/subscribe tags)

Table 485. MQPS_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Value
MQPS_COMMAND				(1)	X	X	X	X		'MQPSCommand'
MQPS_COMP_CODE				(1)	X	X	X	X		'MQPSCompcode'
MQPS_DELETE_OPTIONS				(1)	X	X	X	X		'MQPSDelOpts'
MQPS_ERROR_ID				(1)	X	X	X	X		'MQPSErrorId'
MQPS_ERROR_POS				(1)	X	X	X	X		'MQPSErrorPos'
MQPS_INTEGER_DATA				(1)	X	X	X	X		'MQPSIntData'
MQPS_PARAMETER_ID				(1)	X	X	X	X		'MQPSParmId'
MQPS_PUBLICATION_OPTIONS				(1)	X	X	X	X		'MQSPubOpts'
MQPS_PUBLISH_TIMESTAMP				(1)	X	X	X	X		'MQSPubTime'

MQPS_*

Table 485. MQPS_* constants (continued)

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Value
MQPS_Q_MGR_NAME				(1)	X	X	X	X		'MQPSQMgrName'
MQPS_Q_NAME				(1)	X	X	X	X		'MQPSQName'
MQPS_REASON				(1)	X	X	X	X		'MQPSReason'
MQPS_REASON_TEXT				(1)	X	X	X	X		'MQPSReasonText'
MQPS_REGISTRATION_OPTIONS				(1)	X	X	X	X		'MQPSRegOpts'
MQPS_SEQUENCE_NUMBER				(1)	X	X	X	X		'MQPSSeqNum'
MQPS_STREAM_NAME				(1)	X	X	X	X		'MQPSStreamName'
MQPS_STRING_DATA				(1)	X	X	X	X		'MQPSStringData'
MQPS_TOPIC				(1)	X	X	X	X		'MQPSTopic'
MQPS_USED_ID				(1)	X	X	X	X		'MQPSUserId'
MQPS_COMMAND_B				(1)	X	X	X	X		'bMQPSCommandb'
MQPS_COMP_CODE_B				(1)	X	X	X	X		'bMQPSCompCodeb'
MQPS_DELETE_OPTIONS_B				(1)	X	X	X	X		'bMQPSDelOptsb'
MQPS_ERROR_ID_B				(1)	X	X	X	X		'bMQPSErrorIdb'
MQPS_ERROR_POS_B				(1)	X	X	X	X		'bMQPSErrorPosb'
MQPS_INTEGER_DATA_B				(1)	X	X	X	X		'bMQPSIntegerDatab'
MQPS_PARAMETER_ID_B				(1)	X	X	X	X		'bMQPSParameterIdb'
MQPS_PUBLICATION_OPTIONS_B				(1)	X	X	X	X		'bMQPSPubOptsb'
MQPS_PUBLISH_TIMESTAMP_B				(1)	X	X	X	X		'bMQPSPubTimeB'
MQPS_Q_MGR_NAME_B				(1)	X	X	X	X		'bMQPSQMgrNameb'
MQPS_Q_NAME_B				(1)	X	X	X	X		'bMQPSQNameb'
MQPS_REASON_B				(1)	X	X	X	X		'bMQPSReasonb'
MQPS_REASON_TEXT_B				(1)	X	X	X	X		'bMQPSReasonTextb'
MQPS_REGISTRATION_OPTIONS_B				(1)	X	X	X	X		'bMQPSRegOptsb'
MQPS_SEQUENCE_NUMBER_B				(1)	X	X	X	X		'bMQPSSeqNumb'
MQPS_STREAM_NAME_B				(1)	X	X	X	X		'bMQPSStreamNameb'
MQPS_STRING_DATA_B				(1)	X	X	X	X		'bMQPSStringDatab'
MQPS_TOPIC_B				(1)	X	X	X	X		'bMQPSTopicb'
MQPS_USER_ID_B				(1)	X	X	X	X		'bMQPSUserIdb'

Table 485. MQPS_* constants (continued)

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Value
Note: 1. Supported on AIX, HP-UX, and Sun Solaris only										

MQPS_* (Publish/subscribe tag commands)

Table 486. MQPS_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Value
MQPS_DELETE_PUBLICATION				(1) X	X	X	X	X		'DeletePub'
MQPS_DEREGISTER_PUBLISHER				(1) X	X	X	X	X		'DeregPub'
MQPS_DEREGISTER_SUBSCRIBER				(1) X	X	X	X	X		'DeregSub'
MQPS_PUBLISH				(1) X	X	X	X	X		'Publish'
MQPS_REGISTER_PUBLISHER				(1) X	X	X	X	X		'RegPub'
MQPS_REGISTER_SUBSCRIBER				(1) X	X	X	X	X		'RegSub'
MQPS_REQUEST_UPDATE				(1) X	X	X	X	X		'ReqUpdate'
MQPS_DELETE_PUBLICATION_B				(1) X	X	X	X	X		'bDeletePubb'
MQPS_DEREGISTER_PUBLISHER_B				(1) X	X	X	X	X		'bDeregPubb'
MQPS_DEREGISTER_SUBSCRIBER_B				(1) X	X	X	X	X		'bDeregSubb'
MQPS_PUBLISH_B				(1) X	X	X	X	X		'bPublishb'
MQPS_REGISTER_PUBLISHER_B				(1) X	X	X	X	X		'bRegPubb'
MQPS_REGISTER_SUBSCRIBER_B				(1) X	X	X	X	X		'bRegSubb'
MQPS_REQUEST_UPDATE-B				(1) X	X	X	X	X		'bReqUpdateb'
Note: 1. Supported on AIX, HP-UX, and Sun Solaris only										

MQPS_*

MQPS_* (Publish/subscribe tag options)

Table 487. MQPS_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMs	Tandem NSK	VSE/ESA	Windows	Windows NT	Value
MQPS_ANONYMOUS				(1)	X	X	X	X		'Anon'
MQPS_CORREL_ID_AS_IDENTITY				(1)	X	X	X	X		'CorrelAsId'
MQPS_DEREGISTER_ALL				(1)	X	X	X	X		'DeregAll'
MQPS_DIRECT_REQUESTS				(1)	X	X	X	X		'DirectReq'
MQPS_INCLUDE_STREAM_NAME				(1)	X	X	X	X		'InclStreamName'
MQPS_INFORM_IF_RETAINED				(1)	X	X	X	X		'InformIfRet'
MQPS_IS_RETAINED_PUBLICATION				(1)	X	X	X	X		'IsRetainedPub'
MQPS_LOCAL				(1)	X	X	X	X		'Local'
MQPS_NEW_PUBLICATIONS_ONLY				(1)	X	X	X	X		'NewPubsOnly'
MQPS_NO_REGISTRATION				(1)	X	X	X	X		'NoReg'
MQPS_NONE				(1)	X	X	X	X		'None'
MQPS_OTHER_SUBSCRIBERS_ONLY				(1)	X	X	X	X		'OtherSubsOnly'
MQPS_PUBLISH_ON_REQUEST_ONLY				(1)	X	X	X	X		'PubOnReqOnly'
MQPS_RETAIN_PUBLICATION				(1)	X	X	X	X		'RetainPub'
MQPS_ANONYMOUS_B				(1)	X	X	X	X		'bAnonb'
MQPS_CORREL_ID_AS_IDENTITY_B				(1)	X	X	X	X		'bCorrelAsIdb'
MQPS_DEREGISTER_ALL_B				(1)	X	X	X	X		'bDeregAllb'
MQPS_DIRECT_REQUESTS_B				(1)	X	X	X	X		'bDirectReqb'
MQPS_INCLUDE_STREAM_NAME_B				(1)	X	X	X	X		'bInclStreamNameb'
MQPS_INFORM_IF_RETAINED_B				(1)	X	X	X	X		'bInformIfRetb'
MQPS_IS_RETAINED_PUBLICATION_B				(1)	X	X	X	X		'bIsRetainedPub'
MQPS_LOCAL_B				(1)	X	X	X	X		'bLocalb'
MQPS_NEW_PUBLICATIONS_ONLY_B				(1)	X	X	X	X		'bNewPubsOnlyb'
MQPS_NO_REGISTRATION_B				(1)	X	X	X	X		'bNoRegb'
MQPS_NONE_B				(1)	X	X	X	X		'bNoneb'
MQPS_OTHER_SUBSCRIBERS_ONLY_B				(1)	X	X	X	X		'bOtherSubsOnlyb'
MQPS_PUBLISH_ON_REQUEST_ONLY_B				(1)	X	X	X	X		'bPubOnReqOnlyb'

Table 487. MQPS_* constants (continued)

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Value
MQPS_RETAIN_PUBLICATION_B				(1)	X	X	X	X		'bRetainPubb'
Note:										
1. Supported on AIX, HP-UX, and Sun Solaris only										

MQPUBO_* (Publication options)

Table 488. MQPUBO_* constants

Constant	Decimal	Hex
MQPUBO_NONE	0	00000000
MQPUBO_CORREL_ID_AS_IDENTITY	1	00000001
MQPUBO_RETAIN_PBLICATION	2	00000002
MQPUBO_OTHER_SUBSCRIBERS_ONLY	4	00000004
MQPUBO_NO_REGISTRATION	8	00000008
MQPUBO_IS_RETAINED_PUBLICATION	16	00000010

MQPXP_* (Publish/subscribe routing exit structure identifier)

Table 489. MQPXP_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Value
MQPXP_STRUC_ID										PXPb

MQPXP_* (Publish/subscribe routing exit version)

Table 490. MQPXP_* constants

Constant	Decimal	Hex
MQPXP_VERSION_1	1	00000001

MQPXP_*

Table 490. MQPXP_* constants (continued)

Constant	Decimal	Hex
MQPXP_CURRENT_VERSION	1	00000001

MQQA_* (Inhibit get)

Table 491. MQQA_* constants

Constant	Decimal	Hex
MQQA_GET_ALLOWED	0	00000000
MQQA_GET_INHIBITED	1	00000001

MQQA_* (Inhibit put)

Table 492. MQQA_* constants

Constant	Decimal	Hex
MQQA_PUT_ALLOWED	0	00000000
MQQA_PUT_INHIBITED	1	00000001

MQQA_* (Backout hardening)

Table 493. MQQA_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMs	Tandem NSK	VSE/ESA	Windows	Windows NT	Value
MQQA_BACKOUT_NOT_HARDENED							X			0 00000000
MQQA_BACKOUT_HARDENED							X			1 00000001

MQQA_* (Queue shareability)

Table 494. MQQA_* constants

Constant	Decimal	Hex
MQQA_NOT_SHAREABLE	0	00000000
MQQA_SHAREABLE	1	00000001

MQQDT_* (Queue definition type)

Table 495. MQQDT_* constants

Constant	Decimal	Hex
MQQDT_PREDEFINED	1	00000001
MQQDT_PERMANENT_DYNAMIC	2	00000002
MQQDT_TEMPORARY_DYNAMIC	3	00000003
MQQDT_SHARED_DYNAMIC	4	00000004

MQQF_* (Queue flags)

Table 496. MQQF_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQQF_LOCAL_Q				(1)	X	X	X	X		1	00000001

Note:
1. Supported on AIX, HP-UX, and Sun Solaris only

MQQMDT_* (Queue-manager definition type)

Table 497. MQQMDT_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQQMDT_EXPLICIT_CLUSTER_SENDER				(1)	X	X	X	X		1	00000001
MQQMDT_AUTO_CLUSTER_SENDER				(1)	X	X	X	X		2	00000002
MQQMDT_CLUSTER_RECEIVER				(1)	X	X	X	X		3	00000003
MQQMDT_AUTO_EXP_CLUSTER_SENDER				(1)	X	X	X	X		4	00000004

Note:
1. Supported on AIX, HP-UX, and Sun Solaris only

MQQMF_*

MQQMF_* (Queue-manager flags)

Table 498. MQQMF_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQQMF_REPOSITORY_Q_MGR				(1)	X	X	X	X		2	00000002
MQQMF_CLUSSDR_USER_DEFINED				(1)	X	X	X	X		8	00000008
MQQMF_CLUSSDR_AUTO_DEFINED				(1)	X	X	X	X		16	00000010
MQQMF_AVAILABLE				(1)	X	X	X	X		32	00000020

Note:

1. Supported on AIX, HP-UX, and Sun Solaris only

MQQMT_* (Queue—manager type)

Table 499. MQQMT_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQQMT_NORMAL				(1)	X	X	X	X		0	00000000
MQQMT_REPOSITORY				(1)	X	X	X	X		1	00000001

Note:

1. Supported on AIX, HP-UX, and Sun Solaris only

MQQO_* (Quiesce option)

Table 500. MQQF_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQQO_NO							X	(1)		0	00000000

Table 500. MQQF_* constants (continued)

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQQO_YES							X	(1)		0	00000000
Note:											
1. Supported on Windows 2.1 only											

MQQSG_* (Queue sharing group disposition)

Table 501. MQQSG_* constants

Constant	Decimal	Hex
MQQSG_Q_MGR	0	00000000
MQQSG_COPY	1	00000001
MQQSG_SHARED	2	00000002

MQQSIE_* (Service interval events)

Table 502. MQQSIE_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQQSIE_NONE	X						X	(1)		0	00000000
MQQSIE_HIGH	X						X	(1)		1	00000001
MQQSIE_OK	X						X	(1)		2	00000002
Note:											
1. Supported on Windows 2.1 only											

MQQT_*

MQQT_* (Queue type)

Table 503. MQQT_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal Hex
MQQT_LOCAL										1 00000001
MQQT_MODEL										2 00000002
MQQT_ALIAS										3 00000003
MQQT_REMOTE										6 00000006
MQQT_CLUSTER				(1)	X	X	X	X		7 00000007
MQQT_ALL					X	X	X	(2)		1001 000003E9

Notes::

1. Supported on AIX, HP-UX, and Sun Solaris only
2. Supported on Windows 2.1 only

MQRC_* (Reason code)

Table 504. MQRC_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal Hex
MQRC_NONE										0 00000000
MQRC_ALIAS_BASE_Q_TYPE_ERROR										2001 000007D1
MQRC_ALREADY_CONNECTED										2002 000007D2
MQRC_BACKED_OUT							X			2003 000007D3
MQRC_BUFFER_ERROR										2004 000007D4
MQRC_BUFFER_LENGTH_ERROR										2005 000007D5
MQRC_CHAR_ATTR_LENGTH_ERROR										2006 000007D6
MQRC_CHAR_ATTRS_ERROR										2007 000007D7
MQRC_CHAR_ATTRS_TOO_SHORT										2008 000007D8
MQRC_CONNECTION_BROKEN										2009 000007D9

Table 504. MQRC_* constants (continued)

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal Hex
MQRC_DATA_LENGTH_ERROR										2010 000007DA
MQRC_DYNAMIC_Q_NAME_ERROR										2011 000007DB
MQRC_ENVIRONMENT_ERROR							X			2012 000007DC
MQRC_EXPIRY_ERROR										2013 000007DD
MQRC_FEEDBACK_ERROR										2014 000007DE
MQRC_GET_INHIBITED										2016 000007E0
MQRC_HANDLE_NOT_AVAILABLE										2017 000007E1
MQRC_HCONN_ERROR										2018 000007E2
MQRC_HOBJ_ERROR										2019 000007E3
MQRC_INHIBIT_VALUE_ERROR										2020 000007E4
MQRC_INT_ATTR_COUNT_ERROR										2021 000007E5
MQRC_INT_ATTR_COUNT_TOO_SMALL										2022 000007E6
MQRC_INT_ATTRS_ARRAY_ERROR										2023 000007E7
MQRC_SYNCPOINT_LIMIT_REACHED							X			2024 000007E8
MQRC_MAX_CONNS_LIMIT_REACHED										2025 000007E9
MQRC_MD_ERROR										2026 000007EA
MQRC_MISSING_REPLY_TO_Q										2027 000007EB
MQRC_MSG_TYPE_ERROR										2029 000007ED
MQRC_MSG_TOO_BIG_FOR_Q										2030 000007EE
MQRC_MSG_TOO_BIG_FOR_Q_MGR							X			2031 000007EF
MQRC_NO_MSG_AVAILABLE										2033 000007F1
MQRC_NO_MSG_UNDER_CURSOR										2034 000007F2
MQRC_NOT_AUTHORIZED										2035 000007F3
MQRC_NOT_OPEN_FOR_BROWSE										2036 000007F4
MQRC_NOT_OPEN_FOR_INPUT										2037 000007F5
MQRC_NOT_OPEN_FOR_INQUIRE										2038 000007F6
MQRC_NOT_OPEN_FOR_OUTPUT										2039 000007F7
MQRC_NOT_OPEN_FOR_SET										2040 000007F8
MQRC_OBJECT_CHANGED										2041 000007F9

MQRC_*

Table 504. MQRC_* constants (continued)

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal Hex
MQRC_OBJECT_IN_USE										2042 000007FA
MQRC_OBJECT_TYPE_ERROR										2043 000007FB
MQRC_OD_ERROR										2044 000007FC
MQRC_OPTION_NOT_VALID_FOR_TYPE										2045 000007FD
MQRC_OPTIONS_ERROR										2046 000007FE
MQRC_PERSISTENCE_ERROR										2047 000007FF
MQRC_PERSISTENT_NOT_ALLOWED										2048 00000800
MQRC_PRIORITY_EXCEEDS_MAXIMUM										2049 00000801
MQRC_PRIORITY_ERROR										2050 00000802
MQRC_PUT_INHIBITED										2051 00000803
MQRC_Q_DELETED							X			2052 00000804
MQRC_Q_FULL										2053 00000805
MQRC_Q_NOT_EMPTY							X			2055 00000807
MQRC_Q_SPACE_NOT_AVAILABLE	X									2056 00000808
MQRC_Q_TYPE_ERROR							X			2057 00000809
MQRC_Q_MGR_NAME_ERROR										2058 0000080A
MQRC_Q_MGR_NOT_AVAILABLE										2059 0000080B
MQRC_REPORT_OPTIONS_ERROR										2061 0000080D
MQRC_SECOND_MARK_NOT_ALLOWED		X	X	X	X	X	X	X	X	2062 0000080E
MQRC_SECURITY_ERROR							X			2063 0000080F
MQRC_SELECTOR_COUNT_ERROR										2065 00000811
MQRC_SELECTOR_LIMIT_EXCEEDED										2066 00000812
MQRC_SELECTOR_ERROR										2067 00000813
MQRC_SELECTOR_NOT_FOR_TYPE										2068 00000814
MQRC_SIGNAL_OUTSTANDING		X	X	X	X	X	X	(2)	X	2069 00000815
MQRC_SIGNAL_REQUEST_ACCEPTED		X	X	X	X	X	X	(2)	X	2070 00000816
MQRC_STORAGE_NOT_AVAILABLE										2071 00000817
MQRC_SYNCPOINT_NOT_AVAILABLE	X									2072 00000818
MQRC_TRIGGER_CONTROL_ERROR							X	X		2075 0000081B

Table 504. MQRC_* constants (continued)

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal Hex
MQRC_TRIGGER_DEPTH_ERROR							X	X		2076 0000081C
MQRC_TRIGGER_MSG_PRIORITY_ERR							X	X		2077 0000081D
MQRC_TRIGGER_TYPE_ERROR							X	X		2078 0000081E
MQRC_TRUNCATED_MSG_ACCEPTED										2079 0000081F
MQRC_TRUNCATED_MSG_FAILED										2080 00000820
MQRC_UNKNOWN_ALIAS_BASE_Q										2082 00000822
MQRC_UNKNOWN_OBJECT_NAME										2085 00000825
MQRC_UNKNOWN_OBJECT_Q_MGR										2086 00000826
MQRC_UNKNOWN_REMOTE_Q_MGR										2087 00000827
MQRC_WAIT_INTERVAL_ERROR										2090 0000082A
MQRC_XMIT_Q_TYPE_ERROR										2091 0000082B
MQRC_XMIT_Q_USAGE_ERROR										2092 0000082C
MQRC_NOT_OPEN_FOR_PASS_ALL										2093 0000082D
MQRC_NOT_OPEN_FOR_PASS_IDENT										2094 0000082E
MQRC_NOT_OPEN_FOR_SET_ALL										2095 0000082F
MQRC_NOT_OPEN_FOR_SET_IDENT										2096 00000830
MQRC_CONTEXT_HANDLE_ERROR							X			2097 00000831
MQRC_CONTEXT_NOT_AVAILABLE							X			2098 00000832
MQRC_SIGNAL1_ERROR		X	X	X	X	X	X	(2)	X	2099 00000833
MQRC_OBJECT_ALREADY_EXISTS							X			2100 00000834
MQRC_OBJECT_DAMAGED	X									2101 00000835
MQRC_RESOURCE_PROBLEM	X						X			2102 00000836
MQRC_ANOTHER_Q_MGR_CONNECTED	X						X			2103 00000837
MQRC_UNKNOWN_REPORT_OPTION							X			2104 00000838
MQRC_STORAGE_CLASS_ERROR		X	X	X	X	X	X	X	X	2105 00000839
MQRC_COD_NOT_VALID_FOR_XCF_Q		X	X	X	X	X	X	X	X	2106 0000083A
MQRC_XWAIT_CANCELED		X	X	X	X	X	X	X	X	2107 0000083B
MQRC_XWAIT_ERROR		X	X	X	X	X	X	X	X	2108 0000083C
MQRC_SUPPRESSED_BY_EXIT		X	X	X	X	X	X	X	X	2109 0000083D

MQRC_*

Table 504. MQRC_* constants (continued)

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal Hex
MQRC_FORMAT_ERROR							X			2110 0000083E
MQRC_SOURCE_CCSDID_ERROR							X			2111 0000083F
MQRC_SOURCE_INTEGER_ENC_ERROR							X			2112 00000840
MQRC_SOURCE_DECIMAL_ENC_ERROR							X			2113 00000841
MQRC_SOURCE_FLOAT_ENC_ERROR							X			2114 00000842
MQRC_TARGET_CCSDID_ERROR							X			2115 00000843
MQRC_TARGET_INTEGER_ENC_ERROR							X			2116 00000844
MQRC_TARGET_DECIMAL_ENC_ERROR							X			2117 00000845
MQRC_TARGET_FLOAT_ENC_ERROR							X			2118 00000846
MQRC_NOT_CONVERTED							X			2119 00000847
MQRC_CONVERTED_MSG_TOO_BIG							X			2120 00000848
MQRC_NO_EXTERNAL_PARTICIPANTS	X			(1)	X	X	X	X		2121 00000849
MQRC_PARTICIPANT_NOT_AVAILABLE	X			(1)	X	X	X	X		2122 0000084A
MQRC_OUTCOME_MIXED	X			(1)	X	X	X	X		2123 0000084B
MQRC_OUTCOME_PENDING	X			(1)	X	X	X	X		2124 0000084C
MQRC_BRIDGE_STARTED		X	X	X	X	X	X	X	X	2125 0000084D
MQRC_BRIDGE_STOPPED		X	X	X	X	X	X	X	X	2126 0000084E
MQRC_ADAPTER_STORAGE_SHORTAGE		X	X	X	X	X	X	X	X	2127 0000084F
MQRC_UOW_IN_PROGRESS	X			(1)	X	X	X	X		2128 00000850
MQRC_ADAPTER_CONN_LOAD_ERROR		X	X	X	X	X	X	X	X	2129 00000851
MQRC_ADAPTER_SERV_LOAD_ERROR		X	X	X	X	X	X	X	X	2130 00000852
MQRC_ADAPTER_DEFS_ERROR		X	X	X	X	X	X	X	X	2131 00000853
MQRC_ADAPTER_DEFS_LOAD_ERROR		X	X	X	X	X	X	X	X	2132 00000854
MQRC_ADAPTER_CONV_LOAD_ERROR		X	X	X	X	X	X	X	X	2133 00000855
MQRC_BO_ERROR	X			(1)	X	X	X	X		2134 00000856
MQRC_DH_ERROR	X			(1)	X	X	X	X		2135 00000857
MQRC_MULTIPLE_REASONS	X			(1)	X	X	X	X		2136 00000858
MQRC_OPEN_FAILED	X			(1)	X	X	X	X		2137 00000859
MQRC_ADAPTER_DISC_LOAD_ERROR		X	X	X	X	X	X	X	X	2138 0000085A

Table 504. MQRC_* constants (continued)

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal Hex
MQRC_CNO_ERROR				(1)	X	X	X	X		2139 0000085B
MQRC_DB2_NOT_AVAILABLE		X	X	X	X	X	X	X	X	2342 0000085C
MQRC_DLH_ERROR	X			(1)	X	X	X	X		2141 0000085D
MQRC_HEADER_ERROR	X			(1)	X	X	X	X		2142 0000085E
MQRC_SOURCE_LENGTH_ERROR							X			2143 0000085F
MQRC_TARGET_LENGTH_ERROR							X			2144 00000860
MQRC_SOURCE_BUFFER_ERROR							X			2145 00000861
MQRC_TARGET_BUFFER_ERROR							X			2146 00000862
MQRC_IIH_ERROR	X				X	X	X	X		2148 00000864
MQRC_PCF_ERROR	X				X	X	X			2149 00000865
MQRC_DBCS_ERROR							X			2150 00000866
MQRC_OBJECT_NAME_ERROR	X			(1)	X	X	X	X		2152 00000868
MQRC_OBJECT_Q_MGR_NAME_ERROR	X			(1)	X	X	X	X		2153 00000869
MQRC_RECS_PRESENT_ERROR	X			(1)	X	X	X	X		2154 0000086A
MQRC_OBJECT_RECORDS_ERROR	X			(1)	X	X	X	X		2155 0000086B
MQRC_RESPONSE_RECORDS_ERROR	X			(1)	X	X	X	X		2156 0000086C
MQRC_ASID_MISMATCH		X	X	X	X	X	X	X	X	2157 0000086D
MQRC_PMO_RECORD_FLAGS_ERROR	X			(1)	X	X	X	X		2158 0000086E
MQRC_PUT_MSG_RECORDS_ERROR	X			(1)	X	X	X	X		2159 0000086F
MQRC_CONN_ID_IN_USE		X	X	X	X	X	X	X	X	2160 00000870
MQRC_Q_MGR QUIESCING							X			2161 00000871
MQRC_Q_MGR STOPPING							X			2162 00000872
MQRC_DUPLICATE_RECOV_COORD		X	X	X	X	X	X	X	X	2163 00000873
MQRC_PMO_ERROR										2173 0000087D
MQRC_API_EXIT_LOAD_ERROR		X	X	X	X	X	X	X	X	2183 00000887
MQRC_REMOTE_Q_NAME_ERROR							X			2184 00000888
MQRC_INCONSISTENT_PERSISTENCE	X				X	X	X			2185 00000889
MQRC_GMO_ERROR										2186 0000088A
MQRC_CICS_BRIDGE_RESTRICTION		X	X	X	X	X	X	X	X	2187 0000088B

MQRC_*

Table 504. MQRC_* constants (continued)

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital O/VMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal Hex
MQRC_STOPPED_BY_CLUSTER_EXIT				(1)	X	X	X	X		2188 0000088C
MQRC_CLUSTER_RESOLUTION_ERROR				(1)	X	X	X	X		2189 0000088D
MQRC_CONVERTED_STRING_TOO_BIG				(1)	X	X	X	X		2190 0000088E
MQRC_TMC_ERROR	X				X	X	X	X		2191 0000088F
MQRC_PAGESET_FULL		X	X	X	X	X	X	X	X	2192 00000890
MQRC_STORAGE_MEDIUM_FULL		X	X	X	X	X	X	X	X	2192 00000890
MQRC_PAGESET_ERROR		X	X	X	X	X	X	X	X	2193 00000891
MQRC_NAME_NOT_VALID_FOR_TYPE							X			2194 00000892
MQRC_UNEXPECTED_ERROR										2195 00000893
MQRC_UNKNOWN_XMIT_Q							X			2196 00000894
MQRC_UNKNOWN_DEF_XMIT_Q							X			2197 00000895
MQRC_DEF_XMIT_Q_TYPE_ERROR							X			2198 00000896
MQRC_DEF_XMIT_Q_USAGE_ERROR							X			2199 00000897
MQRC_NAME_IN_USE		X	X	X	X	X	X	X	X	2201 00000899
MQRC_CONNECTION QUIESCING		X	X	X	X	X	X	X	X	2202 0000089A
MQRC_CONNECTION_STOPPING		X	X	X	X	X	X	X	X	2203 0000089B
MQRC_ADAPTER_NOT_AVAILABLE		X	X	X	X	X	X	X	X	2204 0000089C
MQRC_NO_MSG_LOCKED	X									2209 000008A1
MQRC_CONNECTION_NOT_AUTHORIZED		X	X	X	X	X	X	X	X	2217 000008A9
MQRC_MSG_TOO_BIG_FOR_CHANNEL							X			2218 000008AA
MQRC_CALL_IN_PROGRESS							X			2219 000008AB
MQRC_RMH_ERROR	X			(1)	X	X	X	X		2220 000008AC
MQRC_Q_MGR_ACTIVE							X			2222 000008AE
MQRC_Q_MGR_NOT_ACTIVE							X			2223 000008AF
MQRC_Q_DEPTH_HIGH							X			2224 000008B0
MQRC_Q_DEPTH_LOW							X			2225 000008B1
MQRC_Q_SERVICE_INTERVAL_HIGH							X			2226 000008B2
MQRC_Q_SERVICE_INTERVAL_OK							X			2227 000008B3
MQRC_UNIT_OF_WORK_NOT_STARTED	X	X	X	X	X		X	X	X	2232 000008B8

Table 504. MQRC_* constants (continued)

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal Hex
MQRC_CHANNEL_AUTO_DEF_OK					X	X	X	X		2233 000008B9
MQRC_CHANNEL_AUTO_DEF_ERROR					X	X	X	X		2234 000008BA
MQRC_CFH_ERROR	X				X	X	X			2235 000008BB
MQRC_CFIL_ERROR	X				X	X	X			2236 000008BC
MQRC_CFIN_ERROR	X				X	X	X			2237 000008BD
MQRC_CFSL_ERROR	X				X	X	X			2238 000008BE
MQRC_CFST_ERROR	X				X	X	X			2239 000008BF
MQRC_INCOMPLETE_GROUP	X			(1)	X	X	X	X		2241 000008C1
MQRC_INCOMPLETE_MSG	X			(1)	X	X	X	X		2242 000008C2
MQRC_INCONSISTENT_CCSDS	X			(1)	X	X	X	X		2243 000008C3
MQRC_INCONSISTENT_ENCODINGS	X			(1)	X	X	X	X		2244 000008C4
MQRC_INCONSISTENT_UOW	X			(1)	X	X	X	X		2245 000008C5
MQRC_INVALID_MSG_UNDER_CURSOR	X			(1)	X	X	X	X		2246 000008C6
MQRC_MATCH_OPTIONS_ERROR				(1)	X	X	X	X		2247 000008C7
MQRC_MDE_ERROR	X			(1)	X	X	X	X		2248 000008C8
MQRC_MSG_FLAGS_ERROR	X			(1)	X	X	X	X		2249 000008C9
MQRC_MSG_SEQ_NUMBER_ERROR	X			(1)	X	X	X	X		2250 000008CA
MQRC_OFFSET_ERROR	X			(1)	X	X	X	X		2251 000008CB
MQRC_ORIGINAL_LENGTH_ERROR	X			(1)	X	X	X	X		2252 000008CC
MQRC_SEGMENT_LENGTH_ZERO	X			(1)	X	X	X	X		2253 000008CD
MQRC_UOW_NOT_AVAILABLE	X			(1)	X	X	X	X		2255 000008CF
MQRC_WRONG_GMO_VERSION	X			(1)	X	X	X	X		2256 000008D0
MQRC_WRONG_MD_VERSION	X			(1)	X	X	X	X		2257 000008D1
MQRC_GROUP_ID_ERROR	X			(1)	X	X	X	X		2258 000008D2
MQRC_INCONSISTENT_BROWSE	X			(1)	X	X	X	X		2259 000008D3
MQRC_XQH_ERROR	X				X	X	X			2260 000008D4
MQRC_SRC_ENV_ERROR	X			(1)	X	X	X	X		2261 000008D5
MQRC_SRC_NAME_ERROR	X			(1)	X	X	X	X		2262 000008D6
MQRC_DEST_ENV_ERROR	X			(1)	X	X	X	X		2263 000008D7

MQRC_*

Table 504. MQRC_* constants (continued)

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMs	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal Hex
MQRC_DEST_NAME_ERROR	X			(1)	X	X	X	X		2264 000008D8
MQRC_TM_ERROR	X				X	X	X	X		2265 000008D9
MQRC_CLUSTER_EXIT_ERROR				(1)	X	X	X	X		2266 000008E0
MQRC_CLUSTER_EXIT_LOAD_ERROR				(1)	X	X	X	X		2267 000008E1
MQRC_NO_DESTINATIONS_AVAILABLE				(1)	X	X	X	X		2270 000008E4
MQRC_CONN_TAG_IN_USE		X	X	X	X	X	X	X	X	2271 000008DF
MQRC_UOW_TAG_IN_USE		X	X	X	X	X	X	X	X	2271 000008DF
MQRC_CD_ERROR	X			(1)	X	X	X			2277 000008E8
MQRC_CLIENT_CONN_ERROR	X			(1)	X	X	X			2278 000008E8
MQRC_HCONFIG_ERROR	X		X				X	X		2280 000008E8
MQRC_FUNCTION_ERROR	X		X				X	X		2281 000008E9
MQRC_CHANNEL_STARTED							X			2282 000008EA
MQRC_CHANNEL_STOPPED							X			2283 000008EB
MQRC_CHANNEL_CONV_ERROR							X			2284 000008EC
MQRC_SERVICE_NOT_AVAILABLE	X		X				X	X		2285 000008ED
MQRC_INITIALIZATION_FAILED	X		X				X	X		2286 000008EE
MQRC_TERMINATION_FAILED	X		X				X	X		2287 000008EF
MQRC_UNKNOWN_Q_NAME	X		X				X	X		2288 000008F0
MQRC_SERVICE_ERROR	X		X				X	X		2289 000008F1
MQRC_Q_ALREADY_EXISTS	X		X				X	X		2290 000008F2
MQRC_USER_ID_NOT_AVAILABLE	X		X				X	X		2291 000008F3
MQRC_UNKNOWN_ENTITY	X		X					X		2292 000008F4
MQRC_UNKNOWN_AUTH_ENTITY	X		X				X	X		2293 000008F5
MQRC_UNKNOWN_REF_OBJECT	X		X				X	X		2294 000008F6
MQRC_CHANNEL_ACTIVATED							X			2295 000008F7
MQRC_CHANNEL_NOT_ACTIVATED							X			2296 000008F8
MQRC_UOW_CANCELED	X	X	X	X	X		X	X	X	2297 000008F9
MQRC_COMMAND_TYPE_ERROR	X			(1)	X	X	X	X		2300 000008FC
MQRC_MULTIPLE_INSTANCE_ERROR	X			(1)	X	X	X	X		2301 000008FD

Table 504. MQRC_* constants (continued)

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal Hex
MQRC_SYSTEM_ITEM_NOT_ALTERABLE	X			(1)	X	X	X	X		2302 000008FE
MQRC_BAG_CONVERSION_ERROR	X			(1)	X	X	X	X		2303 000008FF
MQRC_SELECTOR_OUT_OF_RANGE	X			(1)	X	X	X	X		2304 00000900
MQRC_SELECTOR_NOT_UNIQUE	X			(1)	X	X	X	X		2305 00000901
MQRC_INDEX_NOT_PRESENT	X			(1)	X	X	X	X		2306 00000902
MQRC_STRING_ERROR	X			(1)	X	X	X	X		2307 00000903
MQRC_ENCODING_NOT_SUPPORTED	X			(1)	X	X	X	X		2308 00000904
MQRC_SELECTOR_NOT_PRESENT	X			(1)	X	X	X	X		2309 00000905
MQRC_OUT_SELECTOR_ERROR	X			(1)	X	X	X	X		2310 00000906
MQRC_STRING_TRUNCATED	X			(1)	X	X	X	X		2311 00000907
MQRC_SELECTOR_WRONG_TYPE	X			(1)	X	X	X	X		2312 00000908
MQRC_INCONSISTENT_ITEM_TYPE	X			(1)	X	X	X	X		2313 00000909
MQRC_INDEX_ERROR	X			(1)	X	X	X	X		2314 0000090A
MQRC_SYSTEM_BAG_NOT_ALTERABLE	X			(1)	X	X	X	X		2315 0000090B
MQRC_ITEM_COUNT_ERROR	X			(1)	X	X	X	X		2316 0000090C
MQRC_FORMAT_NOT_SUPPORTED	X			(1)	X	X	X	X		2317 0000090D
MQRC_SELECTOR_NOT_SUPPORTED	X			(1)	X	X	X	X		2318 0000090E
MQRC_ITEM_VALUE_ERROR	X			(1)	X	X	X	X		2319 0000090F
MQRC_HBAG_ERROR	X			(1)	X	X	X	X		2320 00000910
MQRC_PARAMETER_MISSING	X			(1)	X	X	X	X		2321 00000911
MQRC_CMD_SERVER_NOT_AVAILABLE	X			(1)	X	X	X	X		2322 00000912
MQRC_STRING_LENGTH_ERROR	X			(1)	X	X	X	X		2323 00000913
MQRC_INQUIRY_COMMAND_ERROR	X			(1)	X	X	X	X		2324 00000914
MQRC_NESTED_BAG_NOT_SUPPORTED	X			(1)	X	X	X	X		2325 00000915
MQRC_BAG_WRONG_TYPE	X			(1)	X	X	X	X		2326 00000916
MQRC_ITEM_TYPE_ERROR	X			(1)	X	X	X	X		2327 00000917
MQRC_SYSTEM_BAG_NOT_DELETABLE	X			(1)	X	X	X	X		2328 00000918
MQRC_SYSTEM_ITEM_NOT_DELETABLE	X			(1)	X	X	X	X		2329 00000919
MQRC_CODED_CHAR_SET_ID_ERROR	X			(1)	X	X	X	X		2330 0000091A

MQRC_*

Table 504. MQRC_* constants (continued)

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal Hex
MQRC_MSG_TOKEN_ERROR		X	X	X	X	X	X	X	X	2331 0000091B
MQRC_MISSING_WIH		X	X	X	X	X	X	X	X	2332 0000091C
MQRC_WIH_ERROR		X	X	X	X	X	X	X	X	2333 0000091D
MQRC_RFH_ERROR		X	X	X	X	X	X	X	X	2334 0000091E
MQRC_RFH_STRING_ERROR		X	X	X	X	X	X	X	X	2335 0000091F
MQRC_RFH_COMMAND_ERROR		X	X	X	X	X	X	X	X	2336 00000920
MQRC_RFH_PARM_ERROR		X	X	X	X	X	X	X	X	2337 00000921
MQRC_RFH_DUPLICATE_PARM		X	X	X	X	X	X	X	X	2338 00000922
MQRC_RFH_PARM_MISSING		X	X	X	X	X	X	X	X	2339 00000923
MQRC_CHAR_CONVERSION_ERROR		X	X	X	X	X	X	X	X	2340 00000924
MQRC_UCS2_CONVERSION_ERROR		X	X	X	X	X	X	X	X	2341 00000925
MQRC_DB2_NOT_AVAILABLE							X			2342 00000926
MQRC_OBJECT_NOT_UNIQUE	X			(1)	X	X	X	X		2343 00000927
MQRC_CF_NOT_AVAILABLE	X			(1)	X	X	X	X		2345 00000929
MQRC_CF_STRUC_IN_USE	X			(1)	X	X	X	X		2346 0000092A
MQRC_CF_STRUC_LIST_HDR_IN_USE	X			(1)	X	X	X	X		2347 0000092B
MQRC_CF_STRUC_AUTH_FAILED	X			(1)	X	X	X	X		2348 0000092C
MQRC_CF_STRUC_ERROR	X			(1)	X	X	X	X		2349 0000092D
MQRC_CONN_TAG_NOT_USABLE		X	X	X	X	X	X	X	X	2350 0000092E

Notes:

1. Supported on AIX, HP-UX, and Sun Solaris only
2. Supported on Windows Version 2.1 only

MQRCCF_* (Reason code for command format)

Table 505. MQRCCF_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal Hex
MQRCCF_CFH_TYPE_ERROR	X						X (1)			3001 00000BB9
MQRCCF_CFH_LENGTH_ERROR	X						X (1)			3002 00000BBA
MQRCCF_CFH_VERSION_ERROR	X						X (1)			3003 00000BBB
MQRCCF_CFH_MSG_SEQ_NUMBER_ERROR	X						X (1)			3004 00000BBC
MQRCCF_CFH_CONTROL_ERROR	X						X (1)			3005 00000BBD
MQRCCF_CFH_PARM_COUNT_ERROR	X						X (1)			3006 00000BBE
MQRCCF_CFH_COMMAND_ERROR	X						X (1)			3007 00000BBF
MQRCCF_COMMAND_FAILED	X						X (1)			3008 00000BC0
MQRCCF_CFIN_LRNGTH_ERROR	X						X (1)			3009 00000BC1
MQRCCF_CFST_LENGTH_ERROR	X						X (1)			3010 00000BC2
MQRCCF_CFST_STRING_LENGTH_ERR	X						X (1)			3011 00000BC3
MQRCCF_FORCE_VALUE_ERROR	X						X (1)			3012 00000BC4
MQRCCF_STRUCTURE_TYPE_ERROR	X						X (1)			3013 00000BC5
MQRCCF_CFIN_PARM_ID_ERROR	X						X (1)			3014 00000BC6
MQRCCF_CFST_PARM_ID_ERROR	X						X (1)			3015 00000BC7
MQRCCF_MSG_LENGTH_ERROR	X						X (1)			3016 00000BC8
MQRCCF_CFIN_DUPLICATE_PARM	X						X (1)			3017 00000BC9
MQRCCF_CFST_DUPLICATE_PARM	X						X (1)			3018 00000BCA
MQRCCF_PARM_COUNT_TOO_SMALL	X						X (1)			3019 00000BCB
MQRCCF_PARM_COUNT_TOO_BIG	X						X (1)			3020 00000BCC
MQRCCF_Q_ALREADY_IN_CELL	X						X (1)			3021 00000BCD
MQRCCF_Q_TYPE_ERROR	X						X (1)			3022 00000BCE
MQRCCF_MD_FORMAT_ERROR	X						X (1)			3023 00000BCF
MQRCCF_REPLACE_VALUE_ERROR	X						X (1)			3025 00000BD1
MQRCCF_CFIL_DUPLICATE_ERROR	X						X (1)			3026 00000BD2
MQRCCF_CFIL_COUNT_ERROR	X						X (1)			3027 00000BD3
MQRCCF_CFIL_LENGTH_ERROR	X						X (1)			3028 00000BD4

MQRCCF_*

Table 505. MQRCCF_* constants (continued)

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal Hex
MQRCCF_QUIESCE_VALUE_ERROR	X						X (1)			3029 00000BD5
MQRCCF_MSG_SEQ_NUMBER_ERROR	X						X (1)			3030 00000BD6
MQRCCF_PING_DATA_COUNT_ERROR	X						X (1)			3031 00000BD7
MQRCCF_PING_DATA_COMPARE_ERROR	X						X (1)			3032 00000BD8
MQRCCF_CHANNEL_TYPE_ERROR	X						X (1)			3034 00000BDA
MQRCCF_PARM_SEQUENCE_ERROR	X						X (1)			3035 00000BDB
MQRCCF_XMIT_PROTOCOL_TYPE_ERROR	X						X (1)			3036 00000BDC
MQRCCF_BATCH_SIZE_ERROR	X						X (1)			3037 00000BDD
MQRCCF_DISC_INT_ERROR	X						X (1)			3038 00000BDE
MQRCCF_SHORT_RETRY_TIMER	X						X (1)			3039 00000BDF
MQRCCF_SHORT_TIMER_ERROR	X						X (1)			3040 00000BE0
MQRCCF_LONG_RETRY_ERROR	X						X (1)			3041 00000BE1
MQRCCF_LONG_TIMER_ERROR	X						X (1)			3042 00000BE2
MQRCCF_SEQ_NUMBER_WRAP_ERROR	X						X (1)			3043 00000BE3
MQRCCF_MAX_MSG_LENGTH_ERROR	X						X (1)			3044 00000BE4
MQRCCF_PUT_AUTH_ERROR	X						X (1)			3045 00000BE5
MQRCCF_PURGE_VALUE_ERROR	X						X (1)			3046 00000BE6
MQRCCF_CFIL_PARM_ID_ERROR	X						X (1)			3047 00000BE7
MQRCCF_MSG_TRUNCATED	X						X (1)			3048 00000BE8
MQRCCF_CCSID_ERROR	X						X (1)			3049 00000BE9
MQRCCF_ENCODING_ERROR	X						X (1)			3050 00000BEA
MQRCCF_DATA_CONV_VALUE_ERROR	X						X (1)			3052 00000BEC
MQRCCF_INDOUBT_VALUE_ERROR	X						X (1)			3053 00000BED
MQRCCF_ESCAPE_TYPE_ERROR	X						X (1)			3054 00000BEE
MQRCCF_CHANNEL_TABLE_ERROR	X						X (1)			3062 00000BF6
MQRCCF_MCA_TYPE_ERROR	X						X (1)			3063 00000BF7
MQRCCF_CHL_INST_TYPE_ERROR	X						X (1)			3064 00000BF8
MQRCCF_CHL_STATUS_NOT_FOUND	X						X (1)			3065 00000BF9
MQRCCF_CFSL_DUPLICATE_PARM	X						X (1)			3066 00000BFA

Table 505. MQRCCF_* constants (continued)

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal Hex
MQRCCF_CFSL_TOTAL_LENGTH_ERROR	X						X (1)			3067 00000BFB
MQRCCF_OBJECT_ALREADY_EXISTS	X						X (1)			4001 00000FA1
MQRCCF_OBJECT_WRONG_TYPE	X						X (1)			4002 00000FA2
MQRCCF_LIKE_OBJECT_WRONG_TYPE	X						X (1)			4003 00000FA3
MQRCCF_OBJECT_OPEN	X						X (1)			4004 00000FA4
MQRCCF_ATTR_VALUE_ERROR	X						X (1)			4005 00000FA5
MQRCCF_UNKNOWN_Q_MGR	X						X (1)			4006 00000FA6
MQRCCF_Q_WRONG_TYPE	X						X (1)			4007 00000FA7
MQRCCF_OBJECT_NAME_ERROR	X						X (1)			4008 00000FA8
MQRCCF_ALLOCATE_FAILED	X						X (1)			4009 00000FA9
MQRCCF_HOST_NOT_AVAILABLE	X						X (1)			4010 00000FAA
MQRCCF_CONFIGURATION_ERROR	X						X (1)			4011 00000FAB
MQRCCF_CONNECTION_REFUSED	X						X (1)			4012 00000FAC
MQRCCF_ENTRY_ERROR	X						X (1)			4013 00000FAD
MQRCCF_SEND_FAILED	X						X (1)			4014 00000FAE
MQRCCF_RECEIVED_DATA_ERROR	X						X (1)			4015 00000FAF
MQRCCF_RECEIVE_FAILED	X						X (1)			4016 00000FB0
MQRCCF_CONNECTION_CLOSED	X						X (1)			4017 00000FB1
MQRCCF_NO_STORAGE	X						X (1)			4018 00000FB2
MQRCCF_NO_COMMS_MANAGER	X						X (1)			4019 00000FB3
MQRCCF_LISTENER_NOT_STARTED	X						X (1)			4020 00000FB4
MQRCCF_BIND_FAILED	X						X (1)			4024 00000FB8
MQRCCF_CHANNEL_INDOUBT	X						X (1)			4025 00000FB9
MQRCCF_MQCONN_FAILED	X						X (1)			4026 00000FBA
MQRCCF_MQOPEN_FAILED	X						X (1)			4027 00000FBB
MQRCCF_MQGET_FAILED	X						X (1)			4028 00000FBC
MQRCCF_MQPUT_FAILED	X						X (1)			4029 00000FBD
MQRCCF_PING_ERROR	X						X (1)			4030 00000FBE
MQRCCF_CHANNEL_IN_USE	X						X (1)			4031 00000FBF

MQRCCF_*

Table 505. MQRCCF_* constants (continued)

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal Hex
MQRCCF_CHANNEL_NOT_FOUND	X						X (1)			4032 00000FC0
MQRCCF_UNKNOWN_REMOTE_CHANNEL	X						X (1)			4033 00000FC1
MQRCCF_REMOTE_QM_UNAVAILABLE	X						X (1)			4034 00000FC2
MQRCCF_REMOTE_QM_TERMINATING	X						X (1)			4035 00000FC3
MQRCCF_MQINQ_FAILED	X						X (1)			4036 00000FC4
MQRCCF_NOT_XMIT_Q	X						X (1)			4037 00000FC5
MQRCCF_CHANNEL_DISABLED	X						X (1)			4038 00000FC6
MQRCCF_USER_EXIT_NOT_AVAILABLE	X						X (1)			4039 00000FC7
MQRCCF_COMMIT_FAILED	X						X (1)			4040 00000FC8
MQRCCF_CHANNEL_ALREADY_EXISTS	X						X (1)			4042 00000FCA
MQRCCF_DATA_TOO_LARGE	X						X (1)			4043 00000FCB
MQRCCF_CHANNEL_NAME_ERROR	X						X (1)			4044 00000FCC
MQRCCF_XMIT_Q_NAME_ERROR	X						X (1)			4045 00000FCD
MQRCCF_MCA_NAME_ERROR	X						X (1)			4047 00000FCF
MQRCCF_SEND_EXIT_NAME_ERROR	X						X (1)			4048 00000FD0
MQRCCF_SEC_EXIT_NAME_ERROR	X						X (1)			4049 00000FD1
MQRCCF_MSG_EXIT_NAME_ERROR	X						X (1)			4050 00000FD2
MQRCCF_RCV_EXIT_NAME_ERROR	X						X (1)			4051 00000FD3
MQRCCF_XMIT_Q_NAME_WRONG_TYPE	X						X (1)			4052 00000FD4
MQRCCF_MCA_NAME_WRONG_TYPE	X						X (1)			4053 00000FD5
MQRCCF_DISC_INT_WRONG_TYPE	X						X (1)			4054 00000FD6
MQRCCF_SHORT_RETRY_WRONG_TYPE	X						X (1)			4055 00000FD7
MQRCCF_SHORT_TIMER_WRONG_TYPE	X						X (1)			4056 00000FD8
MQRCCF_LONG_RETRY_WRONG_TYPE	X						X (1)			4057 00000FD9
MQRCCF_LONG_TIMER_WRONG_TYPE	X						X (1)			4058 00000FDA
MQRCCF_PUT_AUTH_WRONG_TYPE	X				X	X	X (1)			4059 00000FDB
MQRCCF_MISSING_CONN_NAME	X						X (1)			4061 00000FDD
MQRCCF_CONN_NAME_ERROR	X						X (1)			4062 00000FDE
MQRCCF_MQSET_FAILED	X						X (1)			4063 00000FDF

Table 505. MQRCCF_* constants (continued)

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal Hex
MQRCCF_CHANNEL_NOT_ACTIVE	X						X	(1)		4064 00000FE0
MQRCCF_TERMINATED_BY_SEC_EXIT	X						X	(1)		4065 00000FE1
MQRCCF_DYNAMIC_Q_SCOPE_ERROR	X						X	(1)		4067 00000FE3
MQRCCF_CELL_DIR_NOT_AVAILABLE	X						X	(1)		4068 00000FE4
MQRCCF_MR_COUNT_ERROR	X						X	(1)		4069 00000FE5
MQRCCF_MR_COUNT_WRONG_TYPE	X						X	(1)		4070 00000FE6
MQRCCF_MR_EXIT_NAME_ERROR	X						X	(1)		4071 00000FE7
MQRCCF_MR_EXIT_NAME_WRONG_TYPE	X						X	(1)		4072 00000FE8
MQRCCF_MR_INTERVAL_ERROR	X						X	(1)		4073 00000FE9
MQRCCF_MR_INTERVAL_WRONG_TYPE	X						X	(1)		4074 00000FEA
MQRCCF_NPM_SPEED_ERROR	X						X	(1)		4075 00000FEB
MQRCCF_NPM_SPEED_WRONG_TYPE	X						X	(1)		4076 00000FEC
MQRCCF_HB_INTERVAL_ERROR	X						X	(1)		4077 00000FED
MQRCCF_HB_INTERVAL_WRONG_TYPE	X						X	(1)		4078 00000FEE
MQRCCF_CHAD_ERROR	X						X	(1)		4079 00000FEF
MQRCCF_CHAD_WRONG_TYPE	X						X	(1)		4080 00000FF0
MQRCCF_CHAD_EVENT_ERROR	X						X	(1)		4081 00000FF1
MQRCCF_CHAD_EVENT_WRONG_TYPE	X						X	(1)		4082 00000FF2
MQRCCF_CHAD_EXIT_ERROR	X						X	(1)		4083 00000FF3
MQRCCF_CHAD_EXIT_WRONG_TYPE	X						X	(1)		4084 00000FF4
MQRCCF_SUPPRESSED_BY_EXIT	X						X	(1)		4085 00000FF5
MQRCCF_BATCH_INT_ERROR	X						X	(1)		4086 00000FF6
MQRCCF_BATCH_INT_WRONG_TYPE	X						X	(1)		4087 00000FF7
Note:										
1. Supported on Windows 2.1 only										

MQREGO_*

MQREGO_* (Registration options)

Table 506. MQREGO_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQREGO_NONE										0	00000000
MQREGO_CORREL_IS_AS_IDENTITY										1	00000001
MQREGO_ANONYMOUS										2	00000002
MQREGO_LOCAL										4	00000004
MQREGO_DIRECT_REQUESTS										8	00000008
MQREGO_NEW_PUBLICATIONS_ONLY										16	00000010
MQREGO_PUBLISH_ON_REQUEST_ONLY										32	00000020
MQREGO_DEREGISTER_ALL										64	00000040
MQREGO_INCLUDE_STREAM_NAME										128	00000080
MQREGO_INFORM_IF_RETAINED										256	00000100

MQRFH_* (Rules and formatting header flags)

Table 507. MQRFH_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQRFH_NONE				(1)	X	X	X	X		0	00000000
Note: 1. Supported on AIX, HP-UX, and Sun Solaris only											

MQRFH_* (Rules and formatting header length)

Table 508. MQRFH_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQRFH_STRUC_LENGTH_FIXED				(1)	X	X	X	X		32	00000020
MQRFH_STRUC_LENGTH_FIXED_2				(1)	X	X	X	X		36	00000024
Note:											
1. Supported on AIX, HP-UX, and Sun Solaris only											

MQRFH_* (Rules and formatting header structure identifier)

Table 509. MQRFH_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Value
MQRFH_STRUC_ID				(1)	X	X	X	X		RFhb
Note:										
1. Supported on AIX, HP-UX, and Sun Solaris only										

MQRFH_* (Rules and formatting header version)

Table 510. MQRFH_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK NSKlines> VSE/ESA	Windows	Windows NT	Decimal	Hex
MQRFH_VERSION_1				(1)	X	X	X	X	1	00000001
MQRFH_VERSION_2				(1)	X	X	X	X	2	00000002
Note:										
1. Supported on AIX, HP-UX, and Sun Solaris only										

MQRL_*

MQRL_* (Returned length)

Table 511. MQRL_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Value
MQRL_UNDEFINED				(1)	X	X	X	X		-1

Note:

1. Supported on AIX, HP-UX, and Sun Solaris only

MQRMH_* (Reference message header structure identifier)

Table 512. MQRMH_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Value
MQRMH_STRUC_ID				(1)	X	X	X	X		RMHb

Note:

1. Supported on AIX, HP-UX, and Sun Solaris only

MQRMH_* (Reference message header version)

Table 513. MQRMH_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal Hex
MQRMH_VERSION_1				(1)	X	X	X	X		1 00000001

Note:

1. Supported on AIX, HP-UX, and Sun Solaris only

MQRMHF_* (Reference message header flags)

Table 514. MQRMH_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQRMHF_NOT_LAST				(1)	X	X	X	X		0	00000000
MQRMHF_LAST				(1)	X	X	X	X		1	00000001

Note:
1. Supported on AIX, HP-UX, and Sun Solaris only

MQRO_* (Report options)

Table 515. MQRO_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQRO_NONE										0	00000000
MQRO_COPY_MSG_ID_TO_CORREL_ID							X			0	00000000
MQRO_NEW_MSG_ID							X			0	00000000
MQRO_DEAD_LETTER_Q	X						X			0	00000000
MQRO_PAN					X	X	X	X		1	00000001
MQRO_NAN					X	X	X	X		2	00000002
MQRO_PASS_CORREL_ID							X			64	00000040
MQRO_PASS_MSG_ID							X			128	00000080
MQRO_COA										256	00000100
MQRO_COA_WITH_DATA										768	00000300
MQRO_COA_WITH_FULL_DATA	X									1792	00000700
MQRO_COD										2048	00000800
MQRO_COD_WITH_DATA										6144	00001800
MQRO_COD_WITH_FULL_DATA	X									14336	00003800
MQRO_EXPIRATION							X			2097152	00200000

MQRO_*

Table 515. MQRO_* constants (continued)

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal Hex	
MQRO_EXPIRATION_WITH_DATA							X			6291456	00600000
MQRO_EXPIRATION_WITH_FULL_DATA	X						X			14680064	00E00000
MQRO_EXCEPTION							X			16777216	01000000
MQRO_EXCEPTION_WITH_DATA							X			50331648	03000000
MQRO_EXCEPTION_WITH_FULL_DATA	X						X			117440512	07000000
MQRO_DISCARD_MSG	X						X			134217728	08000000

MQRO_* (Report-options masks)

Table 516. MQRO_* constants

Constant	Decimal	Hex
MQRO_REJECT_UNSUP_MASK	270270464	101C0000
MQRO_ACCEPT_UNSUP_MASK	-270532353	EFE000FF
MQRO_ACCEPT_UNSUP_IF_XMIT_MASK	261888	0003FF00

MQRP_* (Replace option)

Table 517. MQRP_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal Hex	
MQRP_NO										0	00000000
MQRP_YES										1	00000001

MQRQ_* (Reason qualifier)

Table 518. MQRQ_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQRQ_CONN_NOT_AUTHORIZED										1	00000001
MQRQ_OPEN_NOT_AUTHORIZED										2	00000002
MQRQ_CLOSE_NOT_AUTHORIZED										3	00000003
MQRQ_CMD_NOT_AUTHORIZED										4	00000004
MQRQ_Q_MGR_STOPPING										5	00000005
MQRQ_Q_MGR_QUIESCING										6	00000006
MQRQ_CHANNEL_STOPPED_OK										7	00000007
MQRQ_CHANNEL_STOPPED_ERROR										8	00000008
MQRQ_CHANNEL_STOPPED_RETRY										9	00000009
MQRQ_CHANNEL_STOPPED_DISABLED										10	0000000A
MQRQ_BRIDGE_STOPPED_OK										11	0000000B
MQRQ_BRIDGEL_STOPPED_ERROR										12	0000000C

MQSCO_* (Queue scope)

Table 519. MQSCO_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQSCO_Q_MGR	X					X	X	X		1	00000001
MQSCO_CELL	X					X	X	X		2	00000002

MQSEG_*

MQSEG_* (Segmentation)

Table 520. MQSEG_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Value
MQSEG_INHIBITED	X			(1)	X	X	X	X		b
MQSEG_ALLOWED	X			(1)	X	X	X	X		A

Note:
1. Supported on AIX, HP-UX, and Sun Solaris only

MQSID_* (Security identifier)

Table 521. MQSID_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Value
MQSID_NONE	X			(1)	X	X	X	X		32 nulls
MQSID_NONE_ARRAY	X			(1)	X	X	X	X		32 nulls

Note:
1. Supported on AIX, HP-UX, and Sun Solaris only

MQSIDT_* (Security identifier type)

Table 522. MQSIDT_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQSIDT_NONE	X			(1)	X	X	X	X		0	00000000
MQSIDT_NT_SECURITY_ID	X			(1)	X	X	X	X		1	00000001

Table 522. MQSIDT_* constants (continued)

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
Note:											
1. Supported on AIX, HP-UX, and Sun Solaris only											

MQSP_* (Syncpoint)

Table 523. MQSP_* constants

Constant	Decimal	Hex
MQSP_NOT_AVAILABLE	0	00000000
MQSP_AVAILABLE	1	00000001

MQSS_* (Segment status)

Table 524. MQSS_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Value
MQSS_NOT_A_SEGMENT	X			(1)	X	X	X	X		b
MQSS_LAST_SEGMENT	X			(1)	X	X	X	X		L
MQSS_SEGMENT	X			(1)	X	X	X	X		S
Note:										
1. Supported on AIX, HP-UX, and Sun Solaris only										

MQSUS_* (Suspend status)

Table 525. MQSUS_* constants

Constant	Decimal	Hex
MQSUS_NO	0	00000000
MQSUS_YES	1	00000001

MQTC_*

MQTC_* (Trigger control)

Table 526. MQTC_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQTC_OFF								X		0	00000000
MQTC_ON								X		1	00000001

MQTM_* (Trigger message structure identifier)

Table 527. MQTM_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Value
MQTM_STRUC_ID								X		TMbb

MQTM_* (Trigger message structure version)

Table 528. MQTM_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQTM_VERSION_1								X		1	00000001

MQTMC_* (Trigger message character format identifier)

Table 529. MQTMC_* constants

Constant										Value
	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	
MQTMC_STRUC_ID								X		TMCb

MQTMC_* (Trigger message character format structure)

Table 530. MQTMC_* constants

Constant										Value
	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	
MQTMC_VERSION_1	X	X		X	X	X	X	X	X	bbb1
MQTMC_VERSION_2								X		bbb2

MQTT_* (Trigger type)

Table 531. MQTT_* constants

Constant										Decimal	Hex
	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT		
MQTT_NONE								X		0	00000000
MQTT_FIRST								X		1	00000001
MQTT_EVERY								X		2	00000002
MQTT_DEPTH							X	X		3	00000003

MQTXP_*

MQTXP_* (Transport retry exit structure identifier)

Table 532. MQTXP_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Value
MQTXP_STRUC_ID										TXPb

MQTXP_* (Transport retry exit version)

Table 533. MQTXP_* constants

Constant	Decimal	Hex
MQTXP_VERSION_1	1	00000001
MQTXP_CURRENT_VERSION	1	00000001

MQUA_* (User-attribute selectors)

Table 534. MQUA_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQUA_FIRST	X			(1)	X	X	X	X		65536	00010000
MQUA_LAST	X			(1)	X	X	X	X		999999999	3B9AC9FF

Note:

1. Supported on AIX, HP-UX, and Sun Solaris only

MQUS_* (Usage)

Table 535. MQUS_* constants

Constant	Decimal	Hex
MQUS_NORMAL	0	00000000
MQUS_TRANSMISSION	1	00000001

MQWDR_* (Cluster workload exit destination-record length)

Table 536. MQWDR_* constants

Constant	Decimal	Hex
MQWDR_LENGTH_1	124	0000007C
MQWDR_CURRENT_LENGTH	124	0000007C

MQWDR_* (Cluster workload exit destination-record structure identifier)

Table 537. MQWDR_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Value
MQWDR_STRUC_ID										WDRb

MQWDR_* (Cluster workload exit destination-record structure identifier)

Table 538. MQWDR_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Value
MQWDR_STRUC_ID										WDRb

MQWDR_* (Cluster workload exit destination-record version)

Table 539. MQWDR_* constants

Constant	Decimal	Hex
MQWDR_VERSION_1	1	00000001
MQWDR_CURRENT_VERSION	1	00000001

MQWI_*

MQWI_* (Wait interval)

Table 540. MQWI_* constants

Constant	Decimal	Hex
MQWI_UNLIMITED	-1	FFFFFFFF

MQWIH_* (Workload information header flags)

Table 541. MQWIH_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Value
MQWIH_NONE				(1)	X	X	X	X		0

Note:
1. Supported on AIX, HP-UX, and Sun Solaris only

MQWIH_* (Workload information header structure length)

Table 542. MQWIH_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Value
MQWIH_LENGTH_1				(1)	X	X	X	X		120

Note:
1. Supported on AIX, HP-UX, and Sun Solaris only

MQWIH_* (Workload information header structure identifier)

Table 543. MQWIH_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Value
MQWIH_STRUC_ID				(1)	X	X	X	X		WIHb
Note:										
1. Supported on AIX, HP-UX, and Sun Solaris only										

MQWIH_* (Workload information header version)

Table 544. MQWIH_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Value
MQWIH_VERSION_1				(1)	X	X	X	X		1
Note:										
1. Supported on AIX, HP-UX, and Sun Solaris only										

MQWQR_* (Cluster workload exit queue-record length)

Table 545. MQWQR_* constants

Constant	Decimal	Hex
MQWQR_LENGTH_1	200	000000C8
MQWQR_CURRENT_LENGTH	200	000000C8

MQWQR_*

MQWQR_* (Cluster workload exit queue-record structure identifier)

Table 546. MQWQR_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Value
MQWQR_STRUC_ID										WQRb

MQWQR_* (Cluster workload exit queue-record version)

Table 547. MQWQR_* constants

Constant	Decimal	Hex
MQWQR_VERSION_1	1	00000001
MQWQR_CURRENT_VERSION	1	00000001

MQWXP_* (Cluster workload exit structure identifier)

Table 548. MQWXP_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Value
MQWXP_STRUC_ID										WXPb

MQWXP_* (Cluster workload exit version)

Table 549. MQWXP_* constants

Constant	Decimal	Hex
MQWXP_VERSION_1	1	00000001
MQWXP_CURRENT_VERSION	1	00000001

MQXC_* (Exit command identifier)

Table 550. MQXC_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQXC_MQOPEN		X	X	X	X	X	X	X	X	1	00000001
MQXC_MQCLOSE		X	X	X	X	X	X	X	X	2	00000002
MQXC_MQGET		X	X	X	X	X	X	X	X	3	00000003
MQXC_MQPUT		X	X	X	X	X	X	X	X	4	00000004
MQXC_MQPUT1		X	X	X	X	X	X	X	X	5	00000005
MQXC_MQINQ		X	X	X	X	X	X	X	X	6	00000006
MQXC_MQSET		X	X	X	X	X	X	X	X	8	00000008
MQXC_MQBACK		X	X	X	X	X	X	X	X	9	00000009
MQXC_MQCMIT		X	X	X	X	X	X	X	X	10	0000000A

MQXCC_* (Exit response)

Table 551. MQXCC_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQXCC_SKIP_FUNCTION		X	X	X	X	X	X	X	X	-2	FFFFFFFE
MQXCC_SUPPRESS_FUNCTION							X			-1	FFFFFFF
MQXCC_OK							X			0	00000000
MQXCC_SEND_AND_REQUEST_SEC_MSG							X			-3	FFFFFFFD
MQXCC_SEND_SEC_MSG							X			-4	FFFFFFFC
MQXCC_SUPPRESS_EXIT							X			-5	FFFFFFFB
MQXCC_CLOSE_CHANNEL							X			-6	FFFFFFFA
MQXCC_REQUEST_ACK	X	X	X	(1)	X	X	X	X		-7	FFFFFFF9

Note:

1. Supported on AIX, HP-UX, and Sun Solaris only

MQXDR_*

MQXDR_* (Data-conversion exit response)

Table 552. MQXDR_* constants

Constant	Decimal	Hex
MQXDR_OK	0	00000000
MQXDR_CONVERSION_FAILED	1	00000001

MQXP_* (Exit parameter block)

Table 553. MQXP_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Value
MQXP_STRUC_ID		X	X	X	X	X	X	X	X	XPbb

MQXP_* (Exit parameter block version)

Table 554. MQXP_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQXP_VERSION_1		X	X	X	X	X	X	X	X	1	00000001

MQXPT_* (Transmission protocol type)

Table 555. MQXPT_* constants

Constant	Decimal	Hex
MQXPT_LOCAL	0	00000000
MQXPT_LU62	1	00000001
MQXPT_TCP	2	00000002
MQXPT_NETBIOS	3	00000003
MQXPT_SPX	4	00000004

Table 555. MQXPT_* constants (continued)

Constant	Decimal	Hex
MQXPT_DECNET	5	00000005
MQXPT_UDP	6	00000006

MQXQH_* (Transmission queue header structure identifier)

Table 556. MQXQH_* constants

Constant	Value
MQXQH_STRUC_ID	XQHb

MQXQH_* (Transmission queue header structure version)

Table 557. MQXQH_* constants

Constant	Decimal	Hex
MQXQH_VERSION_1	1	00000001

MQXR_* (Exit reason)

Table 558. MQXR_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQXR_BEFORE		X	X	X	X	X	X	X	X	1	00000001
MQXR_AFTER		X	X	X	X	X	X	X	X	2	00000002
MQXR_INIT							X			11	0000000B
MQXR_TERM							X			12	0000000C
MQXR_MSG							X			13	0000000D
MQXR_XMIT							X			14	0000000E
MQXR_SEC_MSG							X			15	0000000F
MQXR_INIT_SEC							X			16	00000010
MQXR_RETRY	X						X	X		17	00000011
MQXR_AUTO_CLSSDR				(1)	X	X	X	X		18	00000012

MQXR_*

Table 558. MQXR_* constants (continued)

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQXR_AUTO_RECEIVER				(1)	X	X	X	X		19	00000013
MQXR_CLWL_OPEN				(1)	X	X	X	X		20	00000014
MQXR_CLWL_PUT				(1)	X	X	X	X		21	00000015
MQXR_CLWL_MOVE				(1)	X	X	X	X		22	00000016
MQXR_CLWL_REPOS				(1)	X	X	X	X		23	00000017
MQXR_CLWL_REPOS_MOVE				(1)	X	X	X	X		24	00000018
MQXR_END_BATCH	X	X	X	(1)	X	X	X	X		25	00000019
MQXR_ACK_RECEIVED	X	X	X	(1)	X	X	X	X		26	0000001A
MQXR_AUTO_SVRCONN				(1)	X	X	X	X		27	0000001B
MQXR_AUTO_CLUSRCVR				(1)	X	X	X	X		28	0000001C
Note:											
1. Supported on AIX, HP-UX, and Sun Solaris only											

MQXR2_* (Secondary exit response)

Table 559. MQXR2_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQXR2_PUT_WITH_DEF_ACTION										0	00000000
MQXR2_USE_AGENT_BUFFER										0	00000000
MQXR2_DEFAULT_CONTINUATION										0	00000000
MQXR2_STATIC_CACHE										0	00000000
MQXR2_PUT_WITH_DEF_USERID										1	00000001
MQXR2_PUT_WITH_MSG_USERID										2	00000002
MQXR2_USE_EXIT_BUFFER										4	00000004
MQXR2_CONTINUE_CHAIN										8	00000008
MQXR2_SUPPRESS_CHAIN										16	00000010

Table 559. MQXR2_* constants (continued)

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQXR2_DYNAMIC_CACHE										32	00000020

MQXT_* (Exit identifier)

Table 560. MQXT_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQXT_API_CROSSING_EXIT		X	X	X	X	X	X	X	X	1	00000001
MQXT_CHANNEL_SEC_EXIT							X			11	0000000B
MQXT_CHANNEL_MSG_EXIT							X			12	0000000C
MQXT_CHANNEL_SEND_EXIT							X			13	0000000D
MQXT_CHANNEL_RCV_EXIT							X			14	0000000E
MQXT_CHANNEL_MSG_RETRY_EXIT	X						X	X		15	0000000F
MQXT_CHANNEL_AUTO_DEF_EXIT							X	X		16	00000010
MQXT_CLUSTER_WORKLOAD_EXIT				(1)	X	X	X	X		20	00000014
MQXT_PUBSUB_ROUTING_EXIT	X			(1)	X	X	X	X		21	00000015

Note:

1. Supported on AIX, HP-UX, and Sun Solaris only

MQXUA_*

MQXUA_* (Exit user area)

Table 561. MQXUA_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Value	
										Decimal	Hex
MQXUA_NONE							X				16 nulls

MQXWD_* (Exit wait descriptor structure identifier)

Table 562. MQXWD_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Value	
										Decimal	Hex
MQXWD_STRUC_ID											XWDb

MQXWD_* (Exit wait descriptor version)

Table 563. MQXWD_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQXWD_VERSION_1										1	00000001

MQZAET_* (Authority service entity type)

Table 564. MQZAET_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQZAET_PRINCIPAL										1	00000001
MQZAET_GROUP										2	00000002

MQZAO_* (Authority service authorization type)

Table 565. MQZAO_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQZAO_CONNECT										1	00000001
MQZAO_BROWSE										2	00000002
MQZAO_INPUT										4	00000004
MQZAO_OUTPUT										8	00000008
MQZAO_INQUIRE										16	00000010
MQZAO_SET										32	00000020
MQZAO_PASS_IDENTITY_CONTEXT										64	00000040
MQZAO_PASS_ALL_CONTEXT										128	00000080
MQZAO_SET_IDENTITY_CONTEXT										256	00000100
MQZAO_SET_ALL_CONTEXT										512	00000200
MQZAO_ALTERNATE_USER_AUTHORITY										1024	00000400
MQZAO_ALL_MQI										2047	000007FF
MQZAO_CREATE										65536	00010000
MQZAO_DELETE										131072	00020000
MQZAO_DISPLAY										262144	00040000
MQZAO_CHANGE										524288	00080000
MQZAO_CLEAR										1048576	00100000

MQZAO_*

Table 565. MQZAO_* constants (continued)

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQZAO_AUTHORIZE										8388608	00800000
MQZAO_ALL_ADMIN										10354688	009E0000
MQZAO_ALL										10356735	009E07FF
MQZAO_NONE										0	00000000

MQZAS_* (Authority service version)

Table 566. MQZAS_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQZAS_VERSION_1										1	00000001
MQZAS_VERSION_2										2	00000002

MQZCI_* (Continuation indicator)

Table 567. MQZCI_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQZCI_DEFAULT										0	00000000
MQZCI_CONTINUE										0	00000000
MQZCI_STOP										1	00000001

MQZED_* (Entity descriptor structure identifier)

Table 568. MQZED_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Value
MQZED_STRUC_ID										ZEDb

MQZED_* (Entity descriptor version)

Table 569. MQZED_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQZED_VERSION_1										1	00000001
MQZED_CURRENT_VERSION										1	00000001

MQZID_* (Function identifier, all services)

Table 570. MQZID_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQZID_INIT										0	00000000
MQZED_TERM										1	00000001

MQZID_*

MQZID_* (Function identifier, authority service)

Table 571. MQZID_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQZID_INIT_AUTHORITY										0	00000000
MQZID_TERM_AUTHORITY										1	00000001
MQZID_CHECK_AUTHORITY										2	00000002
MQZID_COPY_ALL_AUTHORITY										3	00000003
MQZID_DELETE_AUTHORITY										4	00000004
MQZID_SET_AUTHORITY										5	00000005
MQZID_GET_AUTHORITY										6	00000006
MQZID_GET_EXPLICIT_AUTHORITY										7	00000007

MQZID_* (Function identifier, name service)

Table 572. MQZID_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQZID_INIT_NAME										0	00000000
MQZID_TERM_NAME										1	00000001
MQZID_LOOKUP_NAME										2	00000002
MQZID_INSERT_NAME										3	00000003
MQZID_DELETE_NAME										4	00000004

MQZID_* (Function identifier, userid service)

Table 573. MQZID_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQZID_INIT_USERID										0	00000000
MQZID_TERM_USERID										1	00000001
MQZID_FIND_USERID										2	00000002

MQZIO_* (Initialization options)

Table 574. MQZIO_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQZIO_PRIMARY										0	00000000
MQZIO_SECONDARY										1	00000001

MQZNS_* (Name service version)

Table 575. MQZNS_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQZNS_VERSION_1										1	00000001

MQZTO_*

MQZTO_* (Termination options)

Table 576. MQZTO_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQZTO_PRIMARY										0	00000000
MQZTO_SECONDARY										1	00000001

MQZUS_* (Userid service version)

Table 577. MQZUS_* constants

Constant	OS/390	OS/2	AS/400	UNIX systems	Digital OVMS	Tandem NSK	VSE/ESA	Windows	Windows NT	Decimal	Hex
MQZUS_VERSION_1										1	00000001

Appendix. Notices

This information was developed for products and services offered in the United States. IBM may not offer the products, services, or features discussed in this information in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this information. The furnishing of this information does not give you any license to these patents. You can send license inquiries, in writing, to:

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

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

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

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

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

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the information. IBM may make

Notices

improvements and/or changes in the product(s) and/or the program(s) described in this information at any time without notice.

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

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

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

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

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

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

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

Trademarks

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

AIX	AS/400	CICS
IBM	IBMLink	MQ
MQSeries	OS/2	OS/390
System/390	VSE/ESA	

Lotus, Freelance and WordPro are trademarks of Lotus Development Corporation in the United States, or other countries, or both.

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

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

UNIX is a registered trademark in the United States and/or other countries licensed exclusively through X/Open Company Limited.

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

Sending your comments to IBM

If you especially like or dislike anything about this book, please use one of the methods listed below to send your comments to IBM®.

Feel free to comment on what you regard as specific errors or omissions, and on the accuracy, organization, subject matter, or completeness of this book.

Please limit your comments to the information in this book and the way in which the information is presented.

To make comments about the functions of IBM products or systems, talk to your IBM representative or to your IBM authorized remarketer.

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

You can send your comments to IBM in any of the following ways:

- By mail, to this address:
User Technologies Department (MP095)
IBM United Kingdom Laboratories
Hursley Park
WINCHESTER,
Hampshire
SO21 2JN
United Kingdom
- By fax:
 - From outside the U.K., after your international access code use 44-1962-870229
 - From within the U.K., use 01962-870229
- Electronically, use the appropriate network ID:
 - IBM Mail Exchange: GBIBM2Q9 at IBMMAIL
 - IBMLink™: HURSLEY(IDRCF)
 - Internet: idrcf@hursley.ibm.com

Whichever method you use, ensure that you include:

- The publication title and order number
- The topic to which your comment applies
- Your name and address/telephone number/fax number/network ID.



Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

SX33-6095-07

